

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



LÊ THỊ THU HUYỀN

**NGHIÊN CỨU XÂY DỰNG HỆ THỐNG BÀI TẬP LẬP TRÌNH TRÊN
PYTHON CHO HỌC SINH PHỔ THÔNG**

LUẬN VĂN THẠC SĨ KỸ THUẬT

HÀ NỘI - 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



LÊ THỊ THU HUYỀN

**NGHIÊN CỨU XÂY DỰNG HỆ THỐNG BÀI TẬP LẬP TRÌNH TRÊN
PYTHON CHO HỌC SINH PHỔ THÔNG**

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH
MÃ SỐ: 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT
(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. VŨ VĂN THỎA

HÀ NỘI – 2022

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận văn có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí và các trang web theo danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà nội, ngày 12 tháng 07 năm 2022

Người cam đoan

Lê Thị Thu Huyền

LỜI CẢM ƠN

Trong quá trình thực hiện luận văn này, Học viên luôn nhận được sự hướng dẫn, chỉ bảo rất tận tình của Thầy giáo **TS. Vũ Văn Thỏ**, giảng viên Khoa Công nghệ thông tin 1, là cán bộ trực tiếp hướng dẫn khoa học. Thầy đã giành nhiều thời gian trong việc hướng dẫn học viên cách đọc tài liệu, thu thập và đánh giá thông tin cùng phương pháp nghiên cứu để hoàn thành luận văn.

Học viên xin chân thành cảm ơn các Thầy, Cô giáo trong Học viện Công nghệ Bưu chính Viễn thông đã luôn nhiệt tình giúp đỡ và tạo điều kiện tốt nhất cho em trong suốt quá trình học tập tại trường.

Xin chân thành cảm ơn các anh, các chị và các bạn học viên lớp Cao học – trong Học viện đã luôn động viên, giúp đỡ và nhiệt tình chia sẻ với em những kinh nghiệm học tập, công tác trong suốt khoá học.

Học viên cũng xin chân thành cảm ơn các vị lãnh đạo và các bạn đồng nghiệp tại cơ quan đã luôn tạo mọi điều kiện tốt nhất để em có thể hoàn thành tốt khoá học Cao học này.

Em xin chân thành cảm ơn !

Hà Nội, ngày 12 tháng 07 năm 2022

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
MỤC LỤC.....	iii
DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT	v
DANH SÁCH CÁC HÌNH VẼ	vi
MỞ ĐẦU	1
CHƯƠNG 1 - TỔNG QUAN VỀ DẠY LẬP TRÌNH CHO HỌC SINH PHỔ THÔNG.....	4
1.1 Thực trạng dạy lập trình cho học sinh phổ thông tại Việt Nam.	4
1.2 Tổng quan về chương trình môn Tin học theo khung chương trình mới ban hành năm 2018.	6
1.2.1 Đặc điểm môn học	6
1.2.2 Quan điểm xây dựng chương trình	8
1.2.3 Mục tiêu của môn học.....	10
1.2.4 Yêu cầu cần đạt.....	12
1.3 Các yêu cầu về dạy lập trình đối với học sinh phổ thông.	12
1.4 Phân tích lựa chọn ngôn ngữ lập trình phù hợp cho học sinh phổ thông.....	13
1.5 Kết luận chương 1	15
CHƯƠNG 2 - KHẢO SÁT ĐẶC TRƯNG CƠ BẢN NGÔN NGỮ LẬP TRÌNH PYTHON.....	16
2.1 Lịch sử hình thành và đặc điểm của ngôn ngữ lập trình Python.....	16
2.1.1 Lịch sử hình thành	16
2.1.2 Đặc điểm của ngôn ngữ lập trình Python	17
2.2 Khảo sát các yếu tố của ngôn ngữ lập trình Python	18
2.3 Lựa chọn và trình bày một số nội dung của Python phù hợp với học sinh phổ thông	21
2.3.1 Nhóm nội dung 1	21

2.3.2 Nhóm nội dung 2	29
2.3.3 Nhóm nội dung 3	33
2.3.4 Nhóm nội dung 4	35
2.3.5 Nhóm nội dung 5	38
2.3.6 Nhóm nội dung 6	40
2.4 Giải pháp công nghệ bảo đảm dạy lập trình Python cho học sinh phổ thông	41
2.4.1 Yêu cầu về phần mềm cài đặt Python.....	42
2.4.2 Yêu cầu về phòng máy tính	44
2.4.3 Giải pháp ứng dụng E-learning.....	44
2.5 Kết luận chương 2	45
CHƯƠNG 3- KẾT QUẢ THỰC HIỆN XÂY DỰNG HỆ THỐNG BÀI TẬP LẬP	
TRÌNH CHO HỌC SINH PHỔ THÔNG	46
3.1 Các yêu cầu đối với hệ thống bài tập lập trình cho học sinh phổ thông.....	46
3.1.1 Các yêu cầu về tính khoa học	46
3.1.2 Các yêu cầu về tính sư phạm	47
3.2 Xây dựng hệ thống bài tập lập trình trên Python	48
3.3 Hướng dẫn sử dụng hệ thống bài tập.....	50
3.3.1 Phương pháp hướng dẫn học sinh giải bài tập.....	50
3.3.2 Phương pháp đánh giá bài tập lập trình của học sinh	66
3.4 Kết luận chương 3	69
KẾT LUẬN	70
DANH MỤC TÀI LIỆU THAM KHẢO	72
PHỤ LỤC	73

DANH MỤC CÁC THUẬT NGỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng việt
CNTT	Information Technology	Công nghệ thông tin
CT2018		Chương trình cải cách giáo dục phổ thông năm 2018
CTC		Chương trình con
DL	Digital Literacy	Học văn số hoá phổ thông
HS		Học sinh
ICT	Information & Communication Technology	Công nghệ thông tin và truyền thông
IDE	Integrated Development Environment	Môi trường tích hợp phát triển
IDLE	Integrated Development and Learning Environment	Môi trường tích hợp phát triển và học
MT	Computer	Máy tính
NNLT	Programming language	Ngôn ngữ lập trình
NLT		Người lập trình
PPDH		Phương pháp dạy học
THPT		Trung học phổ thông
SCORM	Sharable Content Object Reference Model	Mô hình tham chiếu đối tượng nội dung
E-Learning	Electronic Learning	Bài giảng điện tử
LMS	Learning Management System	Hệ thống quản lý học tập

DANH SÁCH CÁC HÌNH VẼ

Hình 2.1: Thông tin chi tiết về lệnh <code>print()</code>	19
Hình 2.2: Tính tổng các số chẵn và tích các số lẻ trong các số tự nhiên < 5 . 23	
Hình 2.3: Cửa sổ thực hiện các câu lệnh trực tiếp của IDLE Python.	43
Hình 2.4: Cửa sổ soạn thảo và thực hiện chương trình của IDLE Python.....	43

MỞ ĐẦU

Mục tiêu của giáo dục phổ thông là giúp học sinh phát triển toàn diện về đạo đức, trí tuệ, thể chất, thẩm mỹ và các kỹ năng cơ bản, phát triển năng lực cá nhân, tính năng động và sáng tạo, hình thành nhân cách con người Việt Nam xã hội chủ nghĩa, xây dựng tư cách và trách nhiệm công dân; chuẩn bị cho học sinh tiếp tục học lên hoặc đi vào cuộc sống lao động, tham gia xây dựng và bảo vệ Tổ quốc. Chính vì vậy, giáo dục có vai trò vô cùng quan trọng trong việc xây dựng một thể hệ con người Việt Nam mới phát triển toàn diện, có lý tưởng, đạo đức, có tính tổ chức và kỷ luật, làm chủ tri thức hiện đại, có tư duy sáng tạo, đáp ứng yêu cầu xây dựng và bảo vệ tổ quốc.

Trong thực tế, Tin học là môn học vừa mang tính khoa học vừa mang tính kỹ thuật, ngành Tin học đang phát triển như vũ bão, ngày càng xâm nhập vào các ngành khoa học công nghệ và đời sống, môn Tin học được đưa vào nhà trường những tri thức và phương pháp phổ thông cơ bản nhất trong những thành tựu cơ bản của nhân loại, sắp xếp chúng thành một hệ thống logic đảm bảo tính khoa học, tính tư tưởng, tính thực tiễn, và tính sư phạm, phù hợp với điều kiện hoàn cảnh của đất nước, đáp ứng được yêu cầu của cách mạng khoa học công nghệ ngày nay, phục vụ công cuộc công nghiệp hóa, hiện đại hóa đất nước, chuẩn bị cho nền kinh tế tri thức. Tin học giúp học sinh hình thành nên những kiến thức cơ bản và quan trọng nhất, chuẩn bị cho học sinh một hành trang để bước vào đời hoặc học lên những bậc học cao hơn. Đó cũng chính là chiếc chìa khoá mở cửa cho tương lai.

Hiện nay môn Tin học đã trở thành môn học bắt buộc trong nhà trường phổ thông, đây là một thuận lợi lớn cho việc tiến hành giảng dạy. Tuy nhiên, thực tế ở trường phổ thông nhà trường và học sinh chưa chú ý nhiều đến môn Tin học. Vì vậy đa số học sinh THPT còn non kém về kỹ năng lập trình.

Trong chương trình môn Tin học lớp 11 trường THPT, kỹ năng được đề cập đến và quan trọng nhất đó là những kỹ năng lập trình giải quyết các bài toán. Nhờ có kỹ năng này mà học sinh có được những trải nghiệm thú vị, thấy được máy tính có thể giúp con người giải quyết các bài toán, các công việc một cách tự động bằng

chương trình, chính những nhận thức và kỹ năng lập trình này sẽ kích thích học sinh hứng thú hơn trong việc tìm tòi về thuật toán về ngôn ngữ lập trình (NNLT).

Chương trình Tin học mới 2018 [1] đòi hỏi môn tin phải giúp học sinh định hướng nghề nghiệp. Do đó yêu cầu dạy lập trình cho học sinh phải có đổi mới, phù hợp với yêu cầu của sách giáo khoa cũng như thực tế. NNLT Pascal hiện đang dạy cho học sinh bộc lộ nhiều nhược điểm cần phải thay thế. Hệ thống bài tập lập trình chưa đa dạng, chủ yếu dựa trên các bài tập toán.

Tại Hà Nội, trong năm học 2021 – 2022. Sở giáo dục đào tạo đã khuyến nghị các trường dạy học lập trình cho học sinh lớp 11 trên ngôn ngữ lập trình Python.

Hiện tại, chưa có nhiều tài liệu xây dựng hệ thống bài tập trên ngôn ngữ lập trình Python, đặc biệt là dành cho học sinh phổ thông. Một số tài liệu hiện có [2] đã đưa ra vấn đề dạy ngôn ngữ lập trình Python cho học sinh, nhưng hệ thống bài tập chưa phù hợp với học sinh phổ thông và các yêu cầu mới.

Các tài liệu nước ngoài về ngôn ngữ lập trình Python chủ yếu dành cho các lập trình viên chuyên nghiệp và chưa phù hợp với khung chương trình môn Tin học dành cho học sinh phổ thông [6].

Như vậy, có thể thấy quá trình dạy và học lập trình cho học sinh phổ thông có tầm quan trọng đặc biệt. Quá trình này cần được phát huy cao hơn nữa hiệu quả trong giảng dạy theo tinh thần đổi mới của chương trình cải cách 2018. Với mong muốn đóng góp một phần rất nhỏ về PPDH một môn học còn rất mới mẻ trong nhà trường THPT, xuất phát từ thực tế và mục tiêu như trên, học viên chọn thực hiện đề tài luận văn tốt nghiệp chương trình đào tạo thạc sĩ có tên ***“Nghiên cứu xây dựng hệ thống bài tập lập trình trên Python cho học sinh phổ thông”***.

Mục tiêu của luận văn là nghiên cứu cơ sở lý thuyết và thực tế để xây dựng hệ thống bài tập lập trình trên Python cho học sinh phổ thông.

Để hoàn thành mục tiêu trên, nội dung của luận văn được bố cục gồm ba chương như sau:

Chương 1: Tổng quan về dạy lập trình cho học sinh phổ thông

Nội dung chính của chương 1 là khảo sát tổng quan về thực trạng và yêu cầu dạy lập trình cho học sinh phổ thông theo chương trình cải cách giáo dục phổ thông năm 2018 (CT2018) và các vấn đề liên quan.

Chương 2: Khảo sát các đặc trưng cơ bản ngôn ngữ lập trình Python

Nội dung chính của chương 2 là khảo sát các yếu tố của ngôn ngữ lập trình (NNLT) Python và phân tích lựa chọn, thực hiện trình bày một số nội dung của NNLT Python phù hợp với học sinh phổ thông.

Chương 3: Kết quả thực hiện xây dựng hệ thống bài tập lập trình cho học sinh phổ thông

Nội dung chính của chương 3 là xác định yêu cầu, xây dựng hệ thống bài tập lập trình và hướng dẫn sử dụng phù hợp với giảng dạy lập trình trên Python cho học sinh phổ thông.

CHƯƠNG 1 - TỔNG QUAN VỀ DẠY LẬP TRÌNH CHO HỌC SINH PHỔ THÔNG

Trong chương 1 luận văn sẽ tiến hành khảo sát thực trạng dạy lập trình cho học sinh phổ thông tại Việt Nam và tổng quan về chương trình môn tin học theo khung chương trình mới ban hành cải cách năm 2018. Luận văn sẽ khảo sát các yêu cầu về dạy lập trình đối với học sinh phổ thông và phân tích lựa chọn ngôn ngữ lập trình phù hợp cho học sinh phổ thông.

1.1 Thực trạng dạy lập trình cho học sinh phổ thông tại Việt Nam.

Theo chương trình hiện hành, học sinh phổ thông chỉ học lập trình Pascal ở lớp 11. Ở cấp trung học cơ sở, học sinh chỉ học lập trình Pascal khi tham gia các câu lạc bộ.

Tuy nhiên, môn Tin học nói chung và dạy lập trình nói riêng ở cấp THPT là môn “siêu phụ”. Giáo viên Tin học chỉ được coi là giáo viên loại 2. Gọi là “siêu phụ” vì theo kiểu “so sánh” hay có ở Việt Nam, “Tin học” còn “phụ” hơn cả các môn Sinh – Sử – Địa. Những môn trên còn có thi tốt nghiệp, thi Đại học. Trong khi đó, Tin học không được thi tốt nghiệp, không được thi Đại học (kể cả vào các khoa CNTT).

Chính vì thế, ở cấp THPT, việc dạy và học Tin học đa phần bị buông lỏng, giáo viên muốn dạy như thế nào cũng được (đừng cho điểm quá thấp là được). Về mặt căn bản, học sinh không học đủ, không làm được những bài tập lập trình rất cơ bản. Giáo viên Tin học có thể phải kiêm nhiệm thêm hàng núi các công việc “vô danh” khác và ít có điều kiện trau dồi chuyên môn. Chính vì bị coi là môn phụ, tuyệt đại bộ phận phụ huynh đều ngăn cản con học lập trình sớm. Ngay cả tại một số trường chuyên, một trong những vấn đề khó khăn nhất chính là việc thuyết phục phụ huynh cho phép con học lập trình, theo đuổi niềm đam mê.

Trong bối cảnh đó, cũng có một vài điểm sáng về dạy và học lập trình cho học sinh phổ thông. Rất nhiều phần mềm được sáng tạo bởi học sinh được đánh giá cao trong các kỳ thi khoa học kỹ thuật, tin học trẻ và được triển khai sử dụng thực tế. Học sinh phổ thông Việt Nam cũng thường đoạt giải cao trong các kỳ thi Tin học

khu vực và quốc tế. Điều đó chứng tỏ hoàn toàn có thể dạy và học lập trình cho học sinh đạt kết quả tốt.

Nguyên nhân dẫn đến việc dạy và học lập trình cho học sinh phổ thông thời gian qua chưa đạt kết quả mong muốn có thể kể ra như sau.

Nguyên nhân về phía giáo viên

Giáo viên tin học cấp THPT hiện nay trên địa bàn cả nước xét về mặt bằng chung so với các môn học khác còn thiếu về số lượng và yếu về chuyên môn. Nhiều giáo viên Tin học là các thầy cô không được đào tạo đúng chuyên ngành sư phạm Tin nên còn lúng túng trong giảng dạy môn Tin học, đặc biệt là việc truyền đạt cho học sinh.

Với tâm lý là môn học “phụ”, nhiều giáo viên không yêu thích và đam mê giảng dạy môn Tin. Mặt khác, việc phải kiêm nhiệm nhiều công tác khác ngoài chuyên môn nên giáo viên cũng không có nhiều thời gian để đầu tư cho soạn giáo án, thiết kế chương trình, đổi mới phương pháp giảng dạy hay tự bồi dưỡng chuyên môn.

Nguyên nhân về phía học sinh

Ngoài một số ít học sinh quan tâm và cảm thấy hứng thú với lập trình và công việc lập trình viên thì đa số học sinh chỉ tập trung vào các môn học để thi đại học, thờ ơ hoặc học đối phó với học lập trình. Do đó kết quả học lập trình chưa đạt yêu cầu. Bên cạnh đó, số lượng học sinh/lớp ở các trường THPT khá cao (trung bình 45 - 50 học sinh) cũng gây khó khăn khi dạy và học lập trình.

Nguyên nhân về cơ sở vật chất phục vụ cho dạy và học lập trình

Ở một số trường THPT, cơ sở vật chất phục vụ cho dạy và học lập trình còn thiếu và chưa đồng bộ. Số lượng máy tính trong 1 phòng máy chưa đảm bảo đủ mỗi học sinh 1 máy nên khó khăn cho quá trình thực hành lập trình. Nhiều MT còn hỏng hoặc cấu hình thấp chưa được bảo dưỡng và thay thế kịp thời. Mặt khác, NNLT Pascal hiện đang dạy cho học sinh không có tính thực tiễn cao nên không gây hứng thú cho học sinh. Các tài liệu hỗ trợ cho dạy và học lập trình như hệ thống bài tập, hướng dẫn thực hành còn thiếu hoặc không đồng bộ.

Để đảm bảo tính hiệu quả của quá trình dạy và học lập trình cho học sinh cần từng bước khắc phục các nguyên nhân trên.

Trong luận văn này, học viên sẽ nghiên cứu xây dựng hệ thống bài tập ngôn ngữ lập trình trên Python cho học sinh phổ thông nhằm góp phần nhỏ để khắc phục những tồn tại trên.

1.2 Tổng quan về chương trình môn Tin học theo khung chương trình mới ban hành năm 2018.

Năm 2018, Bộ Giáo dục và Đào tạo đã ban hành “**Chương trình giáo dục phổ thông – Môn tin học**” kèm theo Thông tư số 32/2018/TT-BGDĐT, ngày 26 tháng 12 năm 2018 của Bộ trưởng Bộ Giáo dục và Đào tạo (CT2018) [1]. Trong mục này luận văn sẽ khảo sát tổng quan về CT2018.

1.2.1 Đặc điểm môn học

Giáo dục tin học đóng vai trò chủ đạo trong việc chuẩn bị cho học sinh khả năng tìm kiếm, tiếp nhận, mở rộng tri thức và sáng tạo trong thời đại cách mạng công nghiệp lần thứ tư và toàn cầu hoá. Tin học có ảnh hưởng lớn đến cách sống, cách suy nghĩ và hành động của con người, là công cụ hiệu quả hỗ trợ biến việc học thành tự học suốt đời [1].

Môn Tin học giúp học sinh thích ứng và hoà nhập được với xã hội hiện đại, hình thành và phát triển cho học sinh năng lực tin học để học tập, làm việc và nâng cao chất lượng cuộc sống, đóng góp vào sự nghiệp xây dựng và bảo vệ Tổ quốc.

Nội dung môn Tin học phát triển ba mạch kiến thức hoà quyện: **Học văn số hoá phổ thông (DL), Công nghệ thông tin và truyền thông (ICT), Khoa học máy tính (CS)** và được phân chia theo hai giai đoạn:

- Giai đoạn giáo dục cơ bản:

Môn Tin học giúp học sinh hình thành và phát triển khả năng sử dụng công cụ kỹ thuật số, làm quen và sử dụng Internet; bước đầu hình thành và phát triển tư duy giải quyết vấn đề với sự hỗ trợ của máy tính và hệ thống máy tính; hiểu và tuân theo các nguyên tắc cơ bản trong trao đổi và chia sẻ thông tin.

Ở cấp tiểu học, chủ yếu học sinh học sử dụng các phần mềm đơn giản hỗ trợ học tập và sử dụng thiết bị tin học tuân theo các nguyên tắc giữ gìn sức khoẻ, đồng thời bước đầu được hình thành tư duy giải quyết vấn đề có sự hỗ trợ của máy tính.

Ở cấp trung học cơ sở, học sinh học cách sử dụng, khai thác các phần mềm thông dụng để làm ra sản phẩm số phục vụ học tập và đời sống; thực hành phát hiện và giải quyết vấn đề một cách sáng tạo với sự hỗ trợ của công cụ và các hệ thống tự động hoá của công nghệ kỹ thuật số; học cách tổ chức lưu trữ, quản lý, tra cứu và tìm kiếm dữ liệu số, đánh giá và lựa chọn thông tin.

- Giai đoạn giáo dục định hướng nghề nghiệp:

Môn Tin học có sự phân hoá sâu. Tuỳ theo sở thích và dự định về nghề nghiệp trong tương lai, học sinh lựa chọn một trong hai định hướng: Tin học ứng dụng và Khoa học máy tính.

Hai định hướng có chung một số chủ đề con và mỗi định hướng này còn có những chủ đề con riêng.

Định hướng Tin học ứng dụng đáp ứng nhu cầu sử dụng máy tính như một công cụ của công nghệ kỹ thuật số trong cuộc sống, học tập và làm việc, đem lại sự thích ứng và khả năng phát triển dịch vụ trong xã hội số.

Định hướng Khoa học máy tính đáp ứng mục đích bước đầu tìm hiểu nguyên lý hoạt động của hệ thống máy tính, phát triển tư duy máy tính, khả năng tìm tòi, khám phá các hệ thống tin học, phát triển ứng dụng trên hệ thống máy tính.

Bên cạnh nội dung giáo dục cốt lõi, học sinh có thể chọn một số chuyên đề học tập tuỳ theo sở thích, nhu cầu và định hướng nghề nghiệp. Những chuyên đề thuộc định hướng Tin học ứng dụng nhằm tăng cường thực hành ứng dụng, giúp học sinh thành thạo hơn trong sử dụng các phần mềm thiết yếu, làm ra sản phẩm số thiết thực cho học tập và cuộc sống. Những chuyên đề thuộc định hướng Khoa học máy tính nhằm giới thiệu lập trình điều khiển robot giáo dục, kỹ thuật thiết kế thuật toán, một số cấu trúc dữ liệu và một số nguyên tắc thiết kế mạng máy tính.

1.2.2 Quan điểm xây dựng chương trình

Chương trình môn Tin học cụ thể hoá các quan điểm của Chương trình tổng thể, chú trọng các yêu cầu sau đây:

Tính kế thừa và phát triển

a) Kế thừa chương trình môn Tin học hiện hành

Chương trình môn Tin học kế thừa và phát triển những ưu điểm cơ bản của chương trình hiện hành là tính hệ thống và tính khoa học, đồng thời tránh thiên về lí thuyết trong một số nội dung và sự trùng lặp giữa các cấp học, gây quá tải.

b) Khai thác chương trình môn Tin học phổ thông của các nước tiên tiến

Trong bối cảnh nhiều nước coi trọng phát triển chương trình giáo dục tin học nhằm đào tạo nguồn nhân lực cho cuộc cách mạng công nghiệp lần thứ tư, Chương trình môn Tin học khai thác, chọn lọc vận dụng chương trình môn Tin học của các nước tiên tiến nhằm hội nhập, hướng tới trình độ quốc tế.

Tính khoa học, hiện đại và sư phạm

Chương trình môn Tin học chọn lọc các nội dung cơ bản, phổ thông và hiện đại của ba mạch kiến thức DL, ICT, CS, đồng thời quan tâm đúng mức đến nội dung về đạo đức, pháp luật, văn hoá và ảnh hưởng của tin học đến xã hội, bảo đảm nguyên lí “vừa dạy chữ vừa dạy người” và coi trọng tính nhân văn trong thời đại có sự kết nối cao của thế giới thực và thế giới số.

Chương trình được thiết kế với các nguyên tắc sư phạm: bảo đảm tính vừa sức, phát triển mạch kiến thức vừa theo đường thẳng vừa đồng tâm, xây dựng hệ thống khái niệm cốt lõi. Chương trình chọn lọc nội dung và yêu cầu phù hợp lứa tuổi, xen kẽ những nội dung lí thuyết với thực hành, trừu tượng với trực quan. Các chủ đề lớn xuyên suốt các cấp học với yêu cầu cần đạt nâng cao dần. Các khái niệm cốt lõi được bắt đầu hình thành ở cấp tiểu học và được phát triển hoàn chỉnh dần ở các cấp học cao hơn.

Tính thiết thực

a) Phục vụ định hướng nghề nghiệp

Trong bối cảnh của cuộc cách mạng công nghiệp lần thứ tư có nhiều ngành nghề và việc làm mới xuất hiện đòi hỏi kiến thức, kỹ năng tin học chuyên sâu, Chương trình môn Tin học thể hiện khả năng kết nối và lan toả sâu rộng của tin học đến tất cả các lĩnh vực khác nhau của đời sống, xác lập cho các đối tượng học sinh khác nhau một phổ rộng các ngành nghề chuyên sâu và các ngành nghề ứng dụng tin học.

b) Thực hiện giáo dục STEM

Định hướng giáo dục STEM đang được triển khai như một hướng đi quan trọng trong giáo dục và đào tạo tại nhiều nước trên thế giới. Với tư cách là công nghệ nền tảng, hội tụ đủ tất cả bốn yếu tố giáo dục STEM (Khoa học (S), Công nghệ (T), Kỹ thuật (E) và Toán học (M)), môn Tin học có vai trò trung tâm kết nối các môn học khác, đẩy mạnh giáo dục STEM, phát huy sáng tạo của học sinh nhằm tạo ra sản phẩm số có hàm lượng ICT cao. Chương trình môn Tin học khai thác ưu thế về tích hợp liên môn bằng cách yêu cầu học sinh làm ra sản phẩm số của cá nhân và của nhóm học tập để thu hẹp khoảng cách giữa giáo dục hàn lâm và thực tiễn.

Tính mở

a) Nội dung chương trình mở

Chương trình môn Tin học có các chủ đề bắt buộc, đồng thời có các chủ đề lựa chọn. Các chủ đề nội dung không phụ thuộc vào thiết bị phần cứng và phần mềm cụ thể, không phân biệt phần mềm và học liệu mở hay đóng, tạo thuận lợi cho việc vận dụng phù hợp với khả năng các địa phương và các đối tượng học sinh khác nhau. Do đặc thù riêng, Chương trình môn Tin học cần được cập nhật và điều chỉnh theo định kỳ ngắn hạn theo hướng dẫn của Bộ Giáo dục và Đào tạo nhằm bảo đảm tính hiện đại và thời sự, đáp ứng sự phát triển rất nhanh của công nghệ kỹ thuật số, phù hợp với điều kiện kinh tế - xã hội của đất nước.

b) Hình thức giáo dục đa dạng

Chương trình môn Tin học chọn lọc các chủ đề thiết thực và hấp dẫn, tạo điều kiện cho học sinh học tập và ứng dụng tin học không chỉ trong phạm vi môn Tin học mà cả trong các môn học khác, không chỉ trong khuôn viên nhà trường mà

ở cả các môi trường ngoài khuôn viên trường học (ở nhà, qua mạng máy tính, trong câu lạc bộ và trong thực tế).

1.2.3 Mục tiêu của môn học

Mục tiêu chung

Chương trình môn Tin học góp phần hình thành, phát triển những phẩm chất chủ yếu và năng lực chung đã được xác định trong Chương trình tổng thể, đồng thời góp phần chủ yếu hình thành, phát triển năng lực tin học cho học sinh. Môn Tin học trang bị cho học sinh hệ thống kiến thức tin học phổ thông gồm ba mạch kiến thức hoà quyện:

- Học văn số hoá phổ thông nhằm giúp học sinh hoà nhập với xã hội hiện đại, sử dụng được các thiết bị số và phần mềm cơ bản thông dụng một cách có đạo đức, văn hoá và tuân thủ pháp luật.
- Công nghệ thông tin và truyền thông nhằm giúp học sinh sử dụng và áp dụng hệ thống máy tính giải quyết vấn đề thực tế một cách hiệu quả và sáng tạo.
- Khoa học máy tính nhằm giúp học sinh hiểu biết các nguyên tắc cơ bản và thực tiễn của tư duy máy tính, tạo cơ sở cho việc thiết kế và phát triển các hệ thống máy tính.

Mục tiêu cấp tiểu học

Chương trình môn Tin học ở cấp tiểu học giúp học sinh bước đầu làm quen với công nghệ kỹ thuật số, bắt đầu hình thành năng lực tin học và chuẩn bị cho học sinh tiếp tục học môn Tin học ở cấp trung học cơ sở, cụ thể là:

- Bước đầu hình thành cho học sinh tư duy giải quyết vấn đề với sự trợ giúp của máy tính: Hình thành nhu cầu thu thập, sử dụng thông tin, ý tưởng điều khiển máy tính thông qua việc tạo chương trình đơn giản bằng ngôn ngữ lập trình trực quan.
- Giúp học sinh sử dụng phần mềm tạo ra được những sản phẩm số đơn giản như một văn bản ngắn, thiệp chúc mừng, đoạn hoạt hình vui,...
- Giúp học sinh bước đầu quen với công nghệ kỹ thuật số thông qua việc sử dụng máy tính để vui chơi, học tập, xem và tìm kiếm thông tin trên Internet; rèn luyện cho học sinh một số kỹ năng cơ bản trong sử dụng máy tính; biết bảo vệ sức

khỏe khi sử dụng máy tính, bước đầu có ý thức phòng tránh những tác hại khi sử dụng Internet và ý thức tôn trọng bản quyền.

Mục tiêu cấp trung học cơ sở

Chương trình môn Tin học ở cấp trung học cơ sở giúp học sinh tiếp tục phát triển năng lực tin học đã hình thành ở cấp tiểu học và hoàn thiện năng lực đó ở mức cơ bản, cụ thể là:

- Giúp học sinh phát triển tư duy và khả năng giải quyết vấn đề; biết chọn dữ liệu và thông tin phù hợp, hữu ích; biết chia một vấn đề lớn thành những nhiệm vụ nhỏ hơn; bước đầu có tư duy mô hình hoá một bài toán qua việc hiểu và sử dụng khái niệm thuật toán và lập trình trực quan; biết sử dụng mẫu trong quá trình thiết kế và tạo ra các sản phẩm số; biết đánh giá kết quả sản phẩm số cũng như biết điều chỉnh, sửa lỗi các sản phẩm đó.

- Giúp học sinh có khả năng sử dụng các phương tiện, thiết bị và phần mềm; biết tổ chức lưu trữ, khai thác nguồn tài nguyên đa phương tiện; tạo ra và chia sẻ sản phẩm số đơn giản phục vụ học tập, cuộc sống; có ý thức và khả năng ứng dụng ICT phục vụ cá nhân và cộng đồng.

- Giúp học sinh quen thuộc với dịch vụ số và phần mềm thông dụng để phục vụ cuộc sống, học và tự học, giao tiếp và hợp tác trong cộng đồng; có hiểu biết cơ bản về pháp luật, đạo đức và văn hoá liên quan đến sử dụng tài nguyên thông tin và giao tiếp trên mạng; bước đầu nhận biết được một số ngành nghề chính thuộc lĩnh vực tin học.

Mục tiêu cấp trung học phổ thông

Chương trình môn Tin học ở cấp trung học phổ thông giúp học sinh củng cố và nâng cao năng lực tin học đã được hình thành, phát triển ở giai đoạn giáo dục cơ bản, đồng thời cung cấp cho học sinh tri thức mang tính định hướng nghề nghiệp thuộc lĩnh vực tin học hoặc ứng dụng tin học, cụ thể là:

- Giúp học sinh có những hiểu biết cơ bản về hệ thống máy tính, một số kỹ thuật thiết kế thuật toán, tổ chức dữ liệu và lập trình; củng cố và phát triển hơn nữa

cho học sinh tư duy giải quyết vấn đề, khả năng đưa ra ý tưởng và chuyển giao nhiệm vụ cho máy tính thực hiện.

- Giúp học sinh có khả năng ứng dụng tin học, tạo ra sản phẩm số phục vụ cộng đồng và nâng cao hiệu quả công việc; có khả năng lựa chọn, sử dụng, kết nối các thiết bị số, dịch vụ mạng và truyền thông, phần mềm và các tài nguyên số khác.

- Giúp học sinh có khả năng hoà nhập và thích ứng được với sự phát triển của xã hội số, ứng dụng công nghệ thông tin và truyền thông trong học và tự học; tìm kiếm và trao đổi thông tin theo cách phù hợp, tuân thủ pháp luật, có đạo đức, ứng xử văn hoá và có trách nhiệm; có hiểu biết thêm một số ngành nghề thuộc lĩnh vực tin học, chủ động và tự tin trong việc định hướng nghề nghiệp tương lai của bản thân.

1.2.4 Yêu cầu cần đạt

Yêu cầu cần đạt về phẩm chất chủ yếu và năng lực chung

Môn Tin học góp phần thực hiện các yêu cầu cần đạt về phẩm chất chủ yếu và năng lực chung theo các mức độ phù hợp với môn học, cấp học đã được quy định trong Chương trình tổng thể.

Yêu cầu cần đạt về năng lực đặc thù

Học sinh hình thành, phát triển được năng lực tin học với năm thành phần năng lực sau đây:

- NLa: Sử dụng và quản lí các phương tiện công nghệ thông tin và truyền thông;
- NLb: Ứng xử phù hợp trong môi trường số;
- NLc: Giải quyết vấn đề với sự hỗ trợ của công nghệ thông tin và truyền thông;
- NLd: Ứng dụng công nghệ thông tin và truyền thông trong học và tự học;
- NLe: Hợp tác trong môi trường số.

1.3 Các yêu cầu về dạy lập trình đối với học sinh phổ thông.

Chương trình môn Tin học ở cấp trung học phổ thông thể hiện sự phân hoá sâu hơn về định hướng nghề nghiệp. Do vậy, chương trình có các yêu cầu cần đạt chung về năng lực tin học bắt buộc đối với mọi học sinh và có các yêu cầu bổ sung riêng tương ứng với học sinh chọn định hướng Tin học ứng dụng hoặc Khoa học máy tính. Trong bối cảnh đó cần khảo sát các yêu cầu về dạy lập trình cho học sinh.

Theo CT2018, môn Tin học là môn học tự chọn đối với học sinh phổ thông. Do đó đối tượng học sinh học tin học nói chung và học lập trình nói riêng là những học sinh có hứng thú với môn học và có định hướng nghề nghiệp theo CNTT. Điều này là một thuận lợi lớn khi triển khai dạy và học lập trình.

Yêu cầu về nội dung

Theo quy định của CT2018, khi học về lập trình học sinh cần biết được các cấu trúc dữ liệu cơ bản, các thuật toán sắp xếp và tìm kiếm cơ bản, viết được chương trình giải quyết các bài toán.

Như vậy, khi dạy lập trình cho học sinh phổ thông cần dạy những vấn đề cốt lõi của NNLT.

Yêu cầu về phát triển năng lực cho học sinh

Thông qua dạy lập trình cần rèn luyện cho học sinh khả năng tư duy thuật toán, khả năng đặt bài toán và tìm tòi phương pháp giải bài toán. Đồng thời, bước đầu rèn luyện cho học sinh năng lực phản biện thông qua quá trình đánh giá chương trình.

Như vậy, thông qua quá trình dạy các kỹ năng lập trình cơ bản sẽ giúp học sinh tăng tốc quá trình phát triển, thúc đẩy tính sáng tạo, tăng tính tự tin, nâng cao kỹ năng giải quyết vấn đề và tư duy phản biện. Đồng thời, giúp học sinh thấy được ứng dụng cụ thể của các môn khoa học khác, đặc biệt là Toán học.

Yêu cầu về định hướng nghề nghiệp

Một trong những yêu cầu của CT2018 là tính định hướng nghề nghiệp tương lai cho học sinh phổ thông. Quá trình dạy và học lập trình cần chọn NNLT và phương pháp giảng dạy phù hợp để tạo hứng thú cho học sinh với CNTT, trong đó có nghề lập trình viên.

1.4 Phân tích lựa chọn ngôn ngữ lập trình phù hợp cho học sinh phổ thông.

Do đặc thù của học sinh phổ thông và yêu cầu dạy lập trình, trong mục này luận văn tiến hành phân tích lựa chọn ngôn ngữ lập trình (NNLT) phù hợp cho học sinh phổ thông.

Có thể nêu ra một số tiêu chí để lựa chọn NNLT cho học sinh phổ thông bao gồm tính đơn giản, tính phổ biến và tính chuyên nghiệp.

Tính đơn giản

Tính đơn giản đảm bảo NNLT dễ dàng triển khai dạy và học lập trình cho học sinh phổ thông. Từ khóa, từ dành riêng trong NNLT gần với tiếng Anh thông dụng, số lượng từ khóa và từ dành riêng là vừa đủ. Cấu trúc của một chương trình được tạo với NNLT cần rõ ràng và đơn giản. Cú pháp của NNLT là trong sáng và đơn giản. NNLT phải có trình soạn thảo chương trình đơn giản, dễ học. Nghĩa là, về mặt ngữ nghĩa NNLT không khác biệt nhiều so với một ngôn ngữ thuật toán. Người học không cần thực hiện các biến đổi đáng kể với ngôn ngữ thuật toán để viết mã hoàn thiện chương trình.

Tính đơn giản của NNLT giúp học sinh có thể dễ dàng học các khái niệm cơ bản của lập trình như: đầu vào-đầu ra; câu lệnh rẽ nhánh; cấu trúc lặp; mảng một chiều và hai chiều; chuỗi (xâu ký tự); cấu trúc/bản ghi; tệp (file stream). Từ đó, học sinh có thể dễ dàng chuyển sang học một NNLT khác phù hợp với bản thân.

Tính phổ biến

NNLT cho học sinh phổ thông cần có cộng đồng người học và sử dụng đông đảo để tạo niềm yêu thích cho học sinh. Đồng thời NNLT cung cấp một nền tảng cho phép người học có thể tiến xa hơn trong lập trình. NNLT có thể dễ dàng thực thi trên nhiều nền tảng phần cứng và IDE khác nhau

Tính chuyên nghiệp

NNLT cho học sinh phải đảm bảo tính chuyên nghiệp nghĩa là học sinh có thể tiếp tục học và ứng dụng NNLT trong bậc học cao hơn và cả trong nghề nghiệp sau này. Như vậy, NNLT phải hỗ trợ lập trình có cấu trúc và hướng đối tượng.

Một số kết quả nghiên cứu đã chỉ ra rằng dạy học lập trình với lập trình cấu trúc và tiếp đó là lập trình hướng đối tượng, người học sẽ có kết quả học tập cũng như kỹ năng lập trình tốt hơn. Do đó, NNLT cho học sinh phổ thông cần hỗ trợ lập trình mô-đun. Điều này là quan trọng trong dạy học lập trình, vì đây là nền tảng của lập trình có cấu trúc. Hơn nữa, lập trình mô-đun hỗ trợ cho dạy học lập trình dựa

trên mô đun mẫu, trong đó giáo viên có thể đưa ra một số mô đun có sẵn và học sinh chỉ tập trung vào vấn đề chính, để phát triển sản phẩm. Phương pháp dạy học lập trình này hỗ trợ cho quá trình dạy phương pháp tiếp cận mô hình hóa và mô phỏng cho người học. Bên cạnh đó lập trình mô đun còn hỗ trợ cho phương pháp dạy học tích cực là phương pháp dạy học dựa trên dự án. Lập trình mô đun hóa còn giúp cho người học biết cách mô đun hóa công việc trong đời sống thực tiễn của mình. Đồng thời NNLT nên có khả năng của một ngôn ngữ macro. Đây là công cụ giúp người học hiểu sâu thêm mô hình thực tiễn của các phần mềm ứng dụng.

Từ những tiêu chí trên, luận văn đề xuất chọn NNLT Python để dạy lập trình cho học sinh phổ thông.

Trước hết NNLT Python đáp ứng các tiêu chí về tính đơn giản, tính phổ biến và tính chuyên nghiệp. So với các NNLT khác như Java, C/C++ việc dạy lập trình cho học sinh phổ thông trên Python có nhiều ưu điểm hơn. Mặt khác, hiện tại đã có một số trường THPT và một số khóa dạy lập trình dạy lập trình cho học sinh và bước đầu thu được kết quả khả quan.

Các nội dung khác về NNLT Python sẽ được nghiên cứu sâu hơn ở chương 2.

1.5 Kết luận chương 1

Trong chương 1 luận văn đã tiến hành khảo sát thực trạng dạy lập trình cho học sinh phổ thông tại Việt Nam và tổng quan về chương trình môn tin học theo khung chương trình mới ban hành cải cách năm 2018. Trên cơ sở đó, luận văn đã khảo sát các yêu cầu về dạy lập trình đối với học sinh phổ thông và phân tích lựa chọn ngôn ngữ lập trình phù hợp cho học sinh phổ thông. Từ những khảo sát đó, luận văn đề xuất lựa chọn NNLT Python để dạy và học lập trình cho học sinh phổ thông khi triển khai chương trình môn Tin học theo CT2018.

Các nội dung NNLT Python cần dạy cho học sinh phổ thông sẽ được nghiên cứu trong chương 2.

CHƯƠNG 2 - KHẢO SÁT ĐẶC TRƯNG CƠ BẢN NGÔN NGỮ LẬP TRÌNH PYTHON

Chương 2 của luận văn sẽ khảo sát tổng quan về ngôn ngữ lập trình Python và các vấn đề liên quan. Từ đó luận văn sẽ nghiên cứu lựa chọn và trình bày các nội dung của Python phù hợp với học sinh phổ thông. Luận văn sẽ khảo sát một số yêu cầu kỹ thuật đảm bảo hiệu quả dạy và học lập trình trên Python cho học sinh phổ thông.

2.1 Lịch sử hình thành và đặc điểm của ngôn ngữ lập trình Python

2.1.1 Lịch sử hình thành

Python được tác giả Guido van Rossum thai nghén vào những năm 1980 tại Trung tâm Toán học – Tin học (Centrum Wiskunde & Informatica, CWI) ở Hà Lan. Python lần đầu tiên được giới thiệu vào năm 1989 và chính thức ra đời với phiên bản 0.9.0 vào tháng 2/1991. Ban đầu, Guido van Rossum tạo ra Python với mục đích truy cập vào những cuộc gọi hệ thống Amoeba. Tên Python xuất phát từ tên của chương trình của BBC "Monty Python's Flying Circus" (*Rạp xiếc bay của Monty Python*) [8].

Phiên bản Python 3.0 ra mắt vào ngày 03/12 năm 2008 và được phát triển dưới dạng các phiên bản 3.x cho đến hiện nay.

Python đã trải qua một chặng đường dài để trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới. Kể từ năm 2003, Python luôn được xếp vào một trong số mười ngôn ngữ lập trình phổ biến nhất trong bảng Chỉ số Cộng đồng Lập trình TIOBE. Tính đến tháng 10/2021, Python là ngôn ngữ lập trình phổ biến nhất, từng được chọn là Ngôn ngữ Lập trình của năm vào các năm 2007, 2010, 2018 và 2020. Python được cả cộng đồng lập trình chuyên nghiệp và nghiệp dư đánh giá cao.

Python rất nổi tiếng trong lĩnh vực học máy. Nổi bật nhất là bộ công cụ TensorFlow trong thư viện học máy của Google.

2.1.2 Đặc điểm của ngôn ngữ lập trình Python

Có thể chỉ ra một số đặc điểm tiêu biểu về ngôn ngữ lập trình Python như sau.

- Python là một ngôn ngữ lập trình mã nguồn mở, dễ hiểu, giúp NLT dễ dàng trong lập trình và phù hợp với cho việc phát triển ứng dụng trong thời gian ngắn. Các chương trình trên Python thường dễ đọc do cú pháp đơn giản và được thực thi phù hợp. Python có một cộng đồng hỗ trợ rộng lớn, đã có nhiều diễn đàn hoạt động trực tuyến giúp đỡ khi người lập trình gặp khó khăn với vấn đề nào đó trong công việc.

- Python là một ngôn ngữ lập trình đa mẫu hình. Trong đó, hỗ trợ lập trình hướng đối tượng, lập trình cấu trúc, lập trình hàm và lập trình hướng khía cạnh.

- Python sử dụng kiểu động và một dạng kết hợp giữa đếm tham chiếu và bộ dọn rác kiểm tra theo chu kỳ để quản lý bộ nhớ. Python cũng có tính năng phân giải tên động (liên kết muện), cho phép liên kết các tên biến và phương thức trong quá trình thực thi chương trình.

- Python được thiết kế để dễ dàng mở rộng. Thay vì tích hợp hết tất cả các tính năng vào phần lõi, Python sử dụng các module. Python cho phép có thể dịch chéo sang các ngôn ngữ lập trình khác.

- Python có thư viện lớn để hỗ trợ NLT, trong đó có thể dễ dàng thực hiện các chức năng lập trình.

- Python sử dụng hệ thống kiểu duck typing, còn gọi là latent typing (tự động xác định kiểu). Có nghĩa là, Python không kiểm tra các ràng buộc về kiểu dữ liệu tại thời điểm dịch, mà là tại thời điểm thực thi. Khi thực thi, nếu một thao tác trên một đối tượng bị thất bại, thì có nghĩa là đối tượng đó không sử dụng một kiểu thích hợp.

Các đặc điểm nêu trên đã giúp cho quá trình dạy và học lập trình trên Python trở nên dễ dàng và hiệu quả.

Ngoài ra Python cũng được ứng dụng trong lập trình giải quyết các bài toán trong nhiều lĩnh vực khác nhau.

2.2 Khảo sát các yếu tố của ngôn ngữ lập trình Python

Trong mục này luận văn khảo sát tổng quan các yếu tố của ngôn ngữ lập trình Python. Trên cơ sở đó sẽ phân tích lựa chọn và trình bày các nội dung Python sẽ dạy cho học sinh phổ thông ở mục tiếp theo.

Bảng chữ cái và từ khóa

Trong Python sử dụng bảng chữ cái tiếng Anh, các chữ số thập phân và một số kí tự khác. Điều đặc biệt là Python sử dụng bảng mã Unicode nên có thể xử lý tiếng Việt có dấu rất thuận tiện.

Python sử dụng số lượng từ khóa không nhiều (khoảng 32 từ khóa). Do đó tạo thuận lợi cho NLT không phải nhớ quá nhiều.

Các kiểu dữ liệu đơn giản chuẩn

Trong Python xử lý dữ liệu kiểu số và kiểu logic (bool). Kiểu kí tự không xử lý riêng mà coi như xâu gồm 1 kí tự.

Kiểu số gồm hai loại là số nguyên (*int*) và số thực (*float*). Điểm khác biệt là phạm vi xử lý số nguyên không bị hạn chế như một số NNLT khác. Đồng thời, các số thực có thể xử lý dưới dạng phân số. Ưu điểm này rất phù hợp khi dạy học sinh giải quyết các bài tập tính toán.

Trong Python không yêu cầu khai báo kiểu của biến trước khi sử dụng. Tuy nhiên, kỹ thuật ép kiểu được sử dụng.

Trong Python chứa hầu hết các hàm số học thường gặp được sử dụng trực tiếp thông qua tên hàm hoặc trong module *math* sau khi khai báo `import math`.

Câu lệnh và khối lệnh

Trong Python có các lệnh đơn và các lệnh có cấu trúc. Các câu lệnh đơn bao gồm phép gán (=), nhập dữ liệu từ bàn phím *input()* và xuất dữ liệu ra màn hình *print()*.

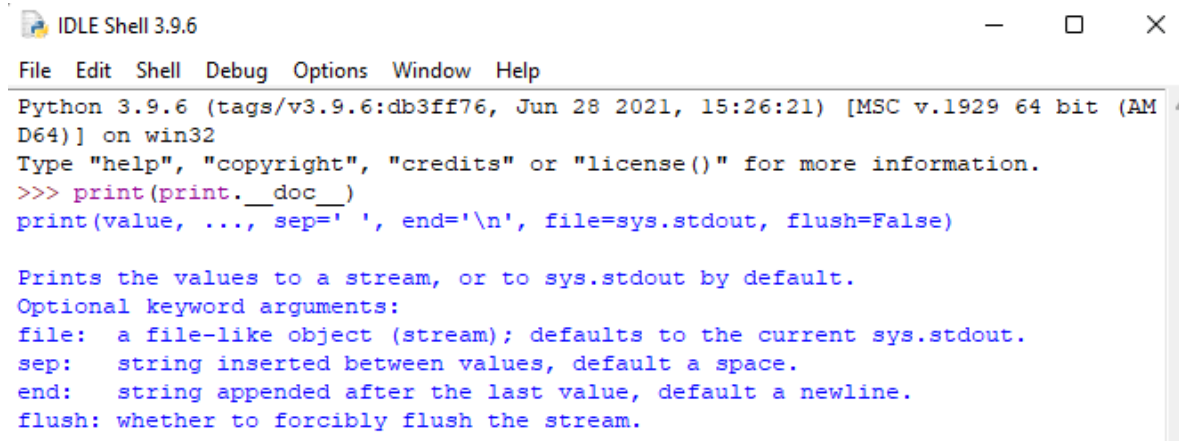
Phép gán trong Python khá linh hoạt. Ví dụ, có thể đổi giá trị của hai biến *x* và *y* cho nhau bằng câu lệnh `x, y = y, x`.

Thực hiện ép kiểu dữ liệu bằng câu lệnh gán:

$$\langle \text{Tên biến} \rangle = \langle \text{Tên kiểu} \rangle (\langle \text{biểu thức} \rangle)$$

Học sinh có thể tìm hiểu thêm thông tin về các lệnh `input()` hoặc `print()` thông qua câu lệnh: `print(<Tên hàm>.__doc__)`.

Hình 2.1 dưới đây mô tả thông tin về lệnh `print()`.



```

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print(print.__doc__)
print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)

Prints the values to a stream, or to sys.stdout by default.
Optional keyword arguments:
file: a file-like object (stream); defaults to the current sys.stdout.
sep: string inserted between values, default a space.
end: string appended after the last value, default a newline.
flush: whether to forcibly flush the stream.
  
```

Hình 2.1: Thông tin chi tiết về lệnh `print()`

Khi thực hiện gỡ lỗi (*debug*) chương trình, NLT có thể kiểm tra giá trị bằng cách gõ tên biến rồi nhấn enter.

Khối lệnh bao gồm một số lệnh được thực hiện như nhau. Trong Python sử dụng thực hiện bằng khoảng trắng hoặc ký tự tab thay vì dùng ngoặc nhọn hay các từ khoá để giới hạn khối lệnh. Lệnh thường được thực hiện vào sau một câu lệnh và thực hiện để đánh dấu kết thúc khối lệnh hiện tại.

Lệnh cấu trúc rẽ nhánh bao gồm lệnh rẽ 1 nhánh *if*, rẽ 2 nhánh *if else* và rẽ nhiều nhánh *if elif else*. Về nguyên tắc, các câu lệnh rẽ nhánh có thể lồng nhau nhiều lần. Tuy nhiên, đối với học sinh nên hạn chế sử dụng quá nhiều câu lệnh rẽ nhánh lồng nhau.

Lệnh lặp có hai câu lệnh *while* và *for*. Lệnh lặp với vòng lặp không xác định *while* có hai dạng *while* và *while else* rất thuận tiện khi sử dụng trong các trường hợp khác nhau. Lệnh lặp với số vòng lặp xác định *for* cũng có hai dạng *for* và *for else* rất thuận tiện khi sử dụng trong các trường hợp khác nhau. Khác với lệnh *while*, lệnh *for* thường gắn với việc sử dụng hàm *range(<Phạm vi>)*. Khi lựa chọn *<Phạm vi>* có thể tạo ra các vòng lặp *for* tiến hoặc lùi.

Các lệnh *break*, *continue* và *pass* thường gắn với các cấu trúc lặp cũng nên được giới thiệu cho học sinh biết và sử dụng.

Các kiểu dữ liệu có cấu trúc

Trong Python có các kiểu dữ liệu có cấu trúc được tích hợp sẵn bao gồm List (danh sách), Tuple, Set (tập hợp) và Dictionary (từ điển). Ngoài ra còn có các kiểu dữ liệu có cấu trúc được xây dựng thông qua các module (class) bao gồm String (xâu kí tự), File (tệp), Stack (ngăn xếp), Queue (hàng đợi), Trees (cây), Graphs (đồ thị), HashMaps.

Tuy nhiên, đối với học sinh phổ thông, luận văn khuyến nghị dạy cho học sinh phổ thông các kiểu dữ liệu có cấu trúc list, string, file. Các kiểu dữ liệu có cấu trúc khác học sinh có thể tự tìm hiểu hoặc thông qua các hình thức khác như câu lạc bộ lập trình, nhóm Tin học trẻ,

Trong Python, kiểu dữ liệu có cấu trúc list vừa có thể sử dụng như cấu trúc mảng (array) hoặc cấu trúc bản ghi (record). Điều đó có thể do list bao gồm các phần tử có kiểu khác nhau. Đối với học sinh phổ thông, luận văn đề xuất khi dạy cho học sinh về list nên trình bày cụ thể dưới dạng danh sách kiểu dãy (mảng 1 chiều) và kiểu bảng (mảng 2 chiều).

Dữ liệu có cấu trúc kiểu string cần được dạy cho học sinh do có nhiều ứng dụng trong các bài toán quản lí và xử lí ngôn ngữ tự nhiên. Học sinh cần làm quen với các thao tác và phép tính trên xâu.

Dữ liệu có cấu trúc kiểu tệp có vai trò quan trọng trong việc nhập, xử lí và lưu trữ dữ liệu. Đối với học sinh phổ thông, chỉ cần hiểu và thực hiện được các thao tác đối với tệp văn bản. Học sinh cần nắm vững các bước đọc dữ liệu từ tệp và ghi dữ liệu ra tệp theo khuôn dạng định trước.

Các hàm

Trong Python không mô tả nội dung về chương trình con (CTC) riêng mà chỉ đề cập đến các hàm mà NLT tự xây dựng. Vì vậy, nội dung này khi dạy cho học sinh cần trình bày dưới dạng CTC. Các thao tác về CTC học sinh cần nắm vững là xây dựng CTC và sử dụng CTC.

Để phát triển tư duy của học sinh và cho học sinh thấy ưu thế của MT khi giải các bài toán nên trình bày cho học sinh về CTC đệ quy. Những nội dung này nên hướng dẫn học sinh thông qua bài tập và các ví dụ cụ thể.

2.3 Lựa chọn và trình bày một số nội dung của Python phù hợp với học sinh phổ thông

Trong mục này luận văn tiến hành lựa chọn và trình bày một số nội dung của ngôn ngữ lập trình Python phù hợp với học sinh phổ thông.

Từ những nội dung đã nghiên cứu ở trên, luận văn đề xuất 6 nhóm nội dung nên được hướng dẫn học sinh phổ thông khi dạy lập trình Python:

- (1) Nhóm nội dung về các kiểu dữ liệu đơn giản chuẩn và các câu lệnh.
- (2) Nhóm nội dung về dữ liệu có cấu trúc dạng danh sách (*list*) với hai dạng kiểu dãy (mảng 1 chiều) và kiểu bảng (mảng hai chiều).
- (3) Nhóm nội dung về dữ liệu có cấu trúc dạng chuỗi ký tự (*string*).
- (4) Nhóm nội dung về dữ liệu có cấu trúc dạng tệp (*file*).
- (5) Nhóm nội dung về chương trình con.
- (6) Nhóm nội dung về sắp xếp và tìm kiếm dữ liệu.

Tiếp theo, luận văn đề xuất trình bày các nhóm nội dung trên phù hợp cho học sinh phổ thông.

2.3.1 Nhóm nội dung 1

Các nội dung nhóm 1 nhằm trang bị cho học sinh những khái niệm chung về NNLT Python, các kiểu dữ liệu đơn giản chuẩn và các câu lệnh.

Bảng chữ cái

Bảng chữ cái là tập các ký tự được dùng để viết chương trình. Không được phép dùng bất kỳ ký tự nào ngoài các ký tự quy định trong bảng chữ cái. Trong Python bảng chữ cái thường bao gồm:

- + 26 chữ cái thường tiếng Anh: a, b, c, ..., z
- + 26 chữ cái in hoa tiếng Anh: A, B, C, ..., Z
- + 10 chữ số thập phân: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- + Các ký tự đặc biệt thường dùng gồm:

+ - * / = < > . : , ; ‘ “
 () [] { } _ % # ^ & dấu cách

Từ khóa

Từ khóa là từ được qui định trước trong ngôn ngữ lập trình (NNLT) với một ý nghĩa cố định, thường dùng để chỉ các loại dữ liệu hoặc kết hợp thành câu lệnh. NLT có thể tạo ra những từ mới để chỉ các đối tượng của mình nhưng không được phép trùng với từ khóa.

Các từ khóa chính trong Python:

and	class	elif	finally	if	lambda	or	try
as	continue	else	for	import	none	pass	while
assert	def	except	from	in	nonlocal	raise	with
break	del	exec	global	is	not	return	yield

Câu lệnh

Những hướng dẫn mà trình thông dịch Python có thể thực hiện được gọi là các câu lệnh. Lệnh thường bao gồm từ khóa/tên hàm và các tham chiếu cần thiết.

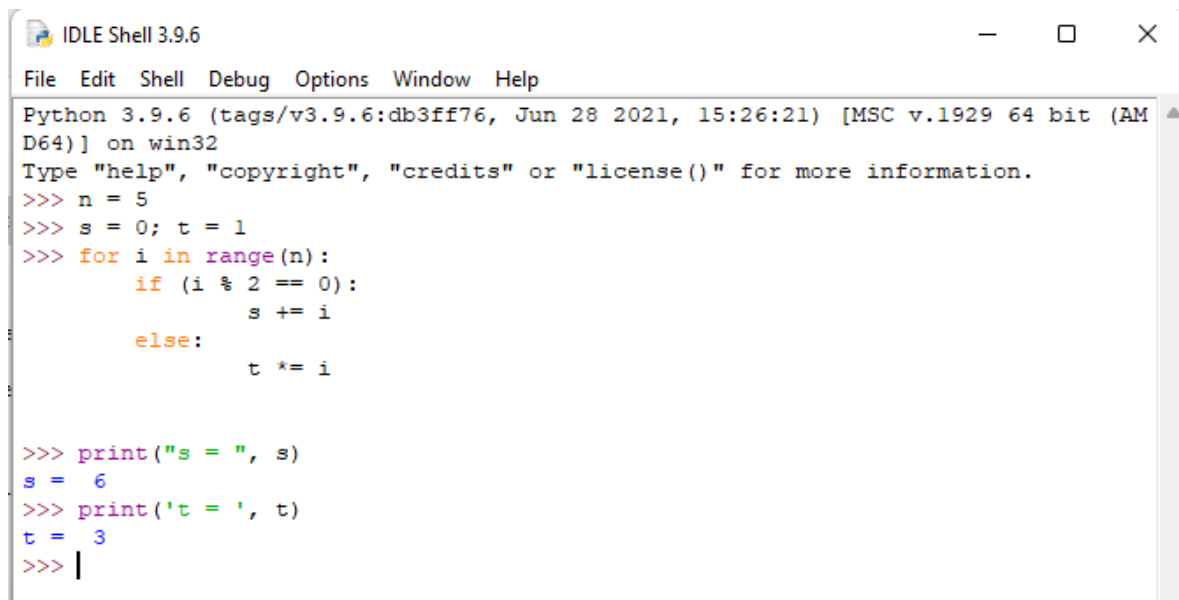
Khối lệnh

Một khối lệnh bao gồm các lệnh được nhận biết thông qua căn lề (thụt lề) như nhau hoặc các lệnh trên cùng 1 dòng được phân cách bởi các dấu ;.

Một khối lệnh bắt đầu với thụt lề và kết thúc khi gặp dòng đầu tiên không thụt lề. Thụt lề bao nhiêu là tùy thuộc NLT nhưng phải nhất quán trong suốt khối lệnh đó, tức là các lệnh trong cùng một khối thì phải có độ thụt lề bằng nhau. Thông thường, sẽ dùng 4 lần phím cách để thụt lề hoặc 1 phím tab.

Khi thực hiện từng câu lệnh trực tiếp, khối lệnh sẽ được máy tính (MT) thực hiện khi nhấn đúp Enter.

Hình 2.2 dưới đây minh họa lệnh và khối lệnh trong Python.



```

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> n = 5
>>> s = 0; t = 1
>>> for i in range(n):
>>>     if (i % 2 == 0):
>>>         s += i
>>>     else:
>>>         t *= i
>>> print("s = ", s)
s = 6
>>> print('t = ', t)
t = 3
>>> |

```

Hình 2.2: Tính tổng các số chẵn và tích các số lẻ trong các số tự nhiên < 5 .

Chương trình

Chương trình là một tập hợp các câu lệnh nhằm giải quyết một bài toán nào đó. Chương trình thường được lưu trong tệp có phần mở rộng **.py**. Các cấu trúc cơ bản của chương trình gồm cấu trúc tuần tự, cấu trúc rẽ nhánh và cấu trúc lặp.

Đặt tên trong Python

Mọi đối tượng trong chương trình đều được đặt tên theo quy tắc sau đây:

- Tên biến chỉ có thể chứa các chữ cái, số và dấu gạch dưới.
- Tên có thể bắt đầu bằng một chữ cái hoặc một dấu gạch dưới, nhưng không phải bằng một số.

- Không được phép sử dụng dấu cách trong tên biến, nhưng có thể sử dụng dấu gạch dưới để tách các từ trong tên biến.

Cần phân biệt tên chuẩn và tên do NLT tự đặt. Tên chuẩn là tên được Python dùng với ý nghĩa nhất định nào đó. Tuy nhiên NLT có thể khai báo và dùng chúng với ý nghĩa và mục đích khác. Tên do NLT tự đặt là tên do NLT đặt ra được dùng với ý nghĩa riêng, xác định bằng cách khai báo trước khi sử dụng. Tên này không được trùng với từ khóa.

Hằng và biến

Hằng là đại lượng có giá trị không thay đổi trong quá trình thực hiện chương trình. Trong Python, tên hằng nên viết bằng các kí tự in hoa để dễ phân biệt.

Biến là đại lượng được đặt tên, dùng để lưu trữ giá trị và giá trị có thể thay đổi trong quá trình thực hiện chương trình.

Chú thích

Có thể đặt các đoạn text để chú thích trong chương trình nguồn của NLT giúp cho người đọc chương trình nhận biết ý nghĩa của chương trình đó dễ dàng hơn. Chú thích không ảnh hưởng đến nội dung chương trình nguồn và được chương trình dịch bỏ qua. Đoạn chú thích được đặt sau dấu #.

Các kiểu dữ liệu đơn giản chuẩn

Trong Python có các kiểu dữ liệu đơn giản chuẩn là kiểu số, kiểu kí tự và kiểu logic.

Kiểu số nguyên:

Tên kiểu: *int*.

Phạm vi: Không giới hạn, phụ thuộc vào bộ nhớ của MT.

Kiểu số thực:

Tên kiểu: *float*.

Độ chính xác: 15 chữ số thập phân.

Kiểu logic

Tên kiểu: *bool*.

Phạm vi: Chỉ nhận 1 trong hai giá trị True (đúng hay 1) hoặc False (sai hay 0).

Câu lệnh gán

Dạng lệnh: <Danh sách n tên biến> = <Danh sách n biểu thức>

Công dụng: Gán giá trị trong <danh sách n biểu thức> tương ứng cho các biến trong <Danh sách n tên biến>. Các tên biến và biểu thức trong danh sách cách nhau bởi dấu phẩy.

Nhập dữ liệu vào từ bàn phím

Dạng lệnh: <tên biến> = input() (1)

<tên biến> = <tên kiểu>(input()) (2)

Công dụng:

(1): Biến <tên biến> sẽ nhận giá trị là xâu gõ từ bàn phím.

(2): Gán giá trị gõ từ bàn phím cho biến <tên biến> với kiểu <tên kiểu> của biến.

Xuất dữ liệu ra màn hình

Dạng lệnh: `print(<Danh sách kết quả>)`

Trong đó, <Danh sách kết quả> là hằng, biến, biểu thức hoặc các kí tự điều khiển phân cách bởi dấu “,”.

Công dụng: In giá trị của <Danh sách kết quả> ra màn hình.

Ghi chú: Các giá trị trong danh sách kết quả đều được chuyển thành xâu trước khi in ra màn hình. Điều khiển xuống dòng: “\n”; thụt lề 1 tab: “\t”. Các kết quả khác nhau phải in cách nhau ít nhất 1 dấu cách ‘ ‘.

Các câu lệnh với cấu trúc rẽ nhánh

Trong chương trình, lệnh/khối lệnh nào đó được thực hiện hoặc bỏ qua không thực hiện tùy thuộc vào một điều kiện xác định gọi là cấu trúc rẽ nhánh. Cấu trúc rẽ nhánh được thực thi bởi câu lệnh rẽ nhánh. Câu lệnh rẽ nhánh có 3 dạng: Lệnh rẽ một nhánh (*if*); Lệnh rẽ hai nhánh (*if else*) và Lệnh rẽ nhiều nhánh (*if elif else*)

Câu lệnh if

Dạng lệnh: `if <Điều kiện>:`

`<Khối lệnh của if>`

Giải thích: *if* là từ khóa, dấu “:” là dấu ngăn cách; <Điều kiện> là biểu thức logic; <Khối lệnh> gồm 1 hoặc nhiều lệnh được thụt lề như nhau.

Hoạt động của lệnh: Nếu <Điều kiện> đúng thì <Khối lệnh của if> được thực hiện. Nếu <Điều kiện> sai thì <Khối lệnh của if> bị bỏ qua không được thực hiện.

Câu lệnh if else

Dạng lệnh: `if <Điều kiện>:`

`<Khối lệnh của if>`

`else:`

`<Khối lệnh của else>`

Giải thích: if, else là từ khóa, dấu “:” là dấu ngăn cách; <Điều kiện> là biểu thức logic; <Khối lệnh của if>, <Khối lệnh của else> gồm 1 hoặc nhiều lệnh được thực hiện như nhau.

Hoạt động của lệnh: Nếu <Điều kiện> đúng thì <Khối lệnh của if> được thực hiện. Nếu <Điều kiện> sai thì <Khối lệnh của else> được thực hiện.

Câu lệnh if elif else

Dạng lệnh: *if <Điều kiện của if>:*

<Khối lệnh của if>

elif <Điều kiện của elif>:

<Khối lệnh của elif>

else:

<Khối lệnh của else>

Giải thích: if, elif, else là từ khóa, dấu “:” là dấu ngăn cách; <Điều kiện ...> là biểu thức logic; <Khối lệnh của if>, <Khối lệnh của elif>, <Khối lệnh của else> gồm 1 hoặc nhiều lệnh được thực hiện như nhau.

Hoạt động của lệnh: Nếu <Điều kiện của if> đúng thì <Khối lệnh của if> được thực hiện. Nếu <Điều kiện của if> sai và <Điều kiện của elif> đúng thì <Khối lệnh của elif> được thực hiện. Nếu <Điều kiện của if> sai và <Điều kiện của elif> sai thì <Khối lệnh của else> được thực hiện;

Ghi chú: Có thể có nhiều elif. Lệnh if elif else có thể xem như lệnh rẽ nhiều nhánh, nhưng chỉ có 1 nhánh được thực hiện. Trong lệnh có thể không có else.

Có thể viết lệnh if...elif...else trong một khối lệnh if...elif...else khác, và tạo thành lệnh if lồng nhau. Không giới hạn số lệnh được lồng vào lệnh khác. Thụt đầu dòng là cách duy nhất để nhận diện mức độ lồng nhau, do đó có thể gây rối hoặc nhầm lẫn. Hạn chế sử dụng các lệnh rẽ nhánh lồng nhau nếu có thể.

Các câu lệnh với cấu trúc lặp

Trong chương trình, lệnh/khối lệnh nào đó được thực hiện một hoặc nhiều lần tùy thuộc vào một điều kiện xác định gọi là cấu trúc lặp. Cấu trúc lặp được thực thi bởi câu lệnh lặp. Câu lệnh lặp thường gặp là lệnh while và lệnh for

Câu lệnh while

Dạng lệnh: *while* <Điều kiện>:

<Khối lệnh của while>

Giải thích: while là từ khóa, dấu “:” là dấu ngăn cách; <Điều kiện> là biểu thức logic; <Khối lệnh của while> gồm 1 hoặc nhiều lệnh được thực hiện lần lượt.

Hoạt động của lệnh: Nếu <Điều kiện> đúng thì <Khối lệnh của while> được thực hiện nhiều lần cho đến khi <Điều kiện> sai hoặc gặp lệnh break. Nếu <Điều kiện> sai thì <Khối lệnh của while> bị bỏ qua không được thực hiện và thoát khỏi vòng lặp.

Câu lệnh while với else

Dạng lệnh: *while* <Điều kiện>:

<Khối lệnh của while>

else:

<Khối lệnh của else>

Giải thích: while, else là từ khóa, dấu “:” là dấu ngăn cách; <Điều kiện> là biểu thức logic; <Khối lệnh ...> gồm 1 hoặc nhiều lệnh được thực hiện lần lượt.

Hoạt động của lệnh:

Nếu <Điều kiện> đúng thì <Khối lệnh của while> được thực hiện nhiều lần cho đến khi <Điều kiện> sai hoặc gặp lệnh break. Nếu <Điều kiện> sai thì <Khối lệnh của else> được thực hiện rồi thoát khỏi vòng lặp.

Trong <Khối lệnh của while> thường chứa lệnh thay đổi <Điều kiện> hoặc lệnh break để tránh vòng lặp vô hạn. Câu lệnh while thường gọi là lệnh lặp với số vòng lặp không biết trước.

Câu lệnh for

Dạng lệnh: *for* <Biến lặp> *in* <Đối tượng>:

<Khối lệnh của for>

Giải thích: for, in là từ khóa, dấu “:” là dấu ngăn cách; <Biến lặp> là tên biến NLT tự đặt; <Đối tượng> là tập hợp có chứa các phần tử như xâu kí tự, dãy số, ...; <Khối lệnh của for> gồm 1 hoặc nhiều lệnh được thực hiện lần lượt.

Hoạt động của lệnh: Kiểm tra <Đối tượng> còn có chứa phần tử nữa không? Nếu đúng thì lấy 1 phần tử theo thứ tự trong <Đối tượng> gán cho <Biến lặp>. Thực hiện <Khối lệnh của for>. Nếu phần tử vừa lấy ra là phần tử cuối cùng của <Đối tượng> thì kết thúc vòng lặp for, ngược lại, tiếp tục thực hiện lượt lặp mới.

Hàm range()

Dạng lệnh: *range(stop) # 1*

range(start, stop) # 2

range(start, stop, step) # 3

Giải thích: range: Tên hàm tạo dãy các số nguyên; start: Số nguyên bắt đầu dãy, mặc định là 0; stop: Số nguyên kết thúc dãy; step: Số nguyên là khoảng cách giữa các số trong dãy, mặc định là 1;

Công dụng: Lệnh 1 tạo dãy gồm stop số từ 0 đến stop-1. Lệnh 2 tạo dãy gồm các số từ start đến stop-1 cách nhau 1. Lệnh 3 tạo dãy các số trong phạm vi từ start đến stop-1 cách nhau step.

Nếu các tham số start, stop, step trong câu lệnh range() không phù hợp MT sẽ báo lỗi hoặc tạo ra dãy rỗng. Hàm range() thường được sử dụng gắn với lệnh for.

Câu lệnh for sử dụng lệnh range()

Dạng lệnh: *for <Biến lặp> in range():*

<Khối lệnh của for>

Giải thích: for, in là từ khóa, dấu “:” là dấu ngăn cách; range() là các dạng lệnh phù hợp tạo dãy các số nguyên; <Biến lặp> là tên biến do NLT tự đặt; <Khối lệnh của for> gồm 1 hoặc nhiều lệnh thụt lề như nhau.

Công dụng: Thực hiện <Khối lệnh của for> với số lần bằng số phần tử của dãy được tạo bởi range().

Câu lệnh for với else

Dạng lệnh: *for <Biến lặp> in <Đối tượng>:*

<Khối lệnh của for>

else:

<Khối lệnh của else>

Giải thích: for, in, else là từ khóa, dấu “:” là dấu ngăn cách; <Biến lặp> là tên biến NLT tự đặt; <Đối tượng> là tập hợp có chứa các phần tử như xâu kí tự, dãy số,...; <Khối lệnh ...> gồm 1 hoặc nhiều lệnh được thụt lề như nhau.

Công dụng: Thực hiện <Khối lệnh của for> với số lần bằng số phần tử của dãy được tạo bởi range() và sau đó thực hiện <Khối lệnh của else>.

Các lệnh break, continue và pass

Lệnh break:

Dạng lệnh: *break* **Công dụng:** Dừng vòng lặp chứa break.

Lệnh continue:

Dạng lệnh: *continue*

Công dụng: Bỏ qua các lệnh trong khối lệnh lặp sau continue

Lệnh pass:

Dạng lệnh: *pass* **Công dụng:** Dùng để làm dấu nhắc (null), không làm gì.

Lệnh break thường sử dụng trong các câu lệnh lặp kèm theo điều kiện nào đó để dừng vòng lặp, tránh vòng lặp vô hạn hoặc đã đạt được kết quả mong muốn. Lệnh continue thường sử dụng trong các câu lệnh lặp kèm theo điều kiện nào đó để bỏ qua 1 lần lặp.

2.3.2 Nhóm nội dung 2

Các nội dung nhóm 2 nhằm trang bị cho học sinh những thao tác làm việc với kiểu dữ liệu có cấu trúc danh sách (*list*). Để phù hợp với học sinh phổ thông, luận văn đề xuất cần dạy kỹ cho học sinh danh sách dạng dãy (mảng 1 chiều) và danh sách dạng bảng (mảng 2 chiều).

Khái niệm

Danh sách là một dãy hữu hạn các phần tử được khai báo trong hai dấu ngoặc vuông [] và phân cách nhau bởi dấu phẩy.

Danh sách có thể chứa các phần tử với kiểu khác nhau, hoặc cùng kiểu. Phần tử của danh sách có thể có kiểu là danh sách. Trong danh sách gồm n phần tử, các phần tử được đánh số thứ tự (index – chỉ mục) từ 0 đến n-1. Số lượng phần tử của

danh sách không bị hạn chế, phụ thuộc vào bộ nhớ trong của MT. Mỗi phần tử của danh sách được truy cập thông qua tên của danh sách và số thứ tự của nó trong [].

Khai báo danh sách

Dạng lệnh khai báo danh sách:

Dạng 1: <Tên danh sách> = []

Dạng 2: <Tên danh sách> = [<Liệt kê các phần tử phân cách bởi dấu phẩy>]

Giải thích: Dạng 1 khai báo một danh sách rỗng với tên là <Tên danh sách> ; Dạng 2 khai báo một danh sách với tên là <Tên danh sách> gồm các phần tử trong <Liệt kê các phần tử phân cách bởi dấu phẩy>.

Số thứ tự (index) của phần tử trong danh sách

Thông thường, trong danh sách gồm n phần tử, các phần tử được đánh số thứ tự (index – chỉ mục) từ đầu đến cuối bắt đầu là 0, kết thúc là n-1.

Số thứ tự có thể âm, trong đó phần tử cuối cùng có số thứ tự là -1, phần tử đầu tiên là -n.

Các thao tác trên danh sách

Phép gán các danh sách

Dạng lệnh: <ds1> = <ds2>

Giải thích: <ds1> sẽ gồm các phần tử như <ds2>.

Phép cộng các danh sách

Dạng lệnh: <ds> = <ds1> + <ds2>

Giải thích: Các phần tử <ds2> sẽ được ghép vào sau <ds1> để tạo nên <ds>.

Tính số phần tử hiện có của danh sách

Dạng lệnh: len(<ds>)

Giải thích: len() là tên hàm; lệnh cho giá trị là số phần tử hiện có của danh sách <ds>.

Truy cập các phần tử của danh sách

Truy cập 1 phần tử

Dạng lệnh: <ds>[<vị trí>]

Giải thích: <ds> là tên danh sách, <vị trí> là số thứ tự (index) của phần tử trong danh sách.

Công dụng: Truy cập phần tử đứng ở <vị trí> của danh sách <ds>.

Truy cập nhiều phần tử (lát cắt – slice)

Dạng lệnh: <ds>[<vị trí 1>:<vị trí 2>]

Giải thích: <ds> là tên danh sách, <vị trí 1>, <vị trí 2> là phạm vi cần truy cập, dấu : là toán tử lát cắt.

Công dụng: Truy cập các phần tử từ <vị trí 1> đến <vị trí 2>-1.

Kiểm tra phần tử có thuộc danh sách không

Dạng lệnh: <phần tử> in <ds>

Giải thích: <phần tử> là giá trị, in là toán tử, <ds> là tên danh sách. Lệnh sẽ trả về giá trị True nếu phần tử đã tồn tại trong <ds>, ngược lại sẽ trả về False

Xóa các phần tử của danh sách

Dạng lệnh: del <phần tử>

Giải thích: del là toán tử, <phần tử> là một hoặc nhiều phần tử được truy cập của danh sách hoặc bản thân danh sách.

Công dụng: Xóa <phần tử> của danh sách.

Các phương thức thực hiện với danh sách

append(): Thêm phần tử vào cuối list.

sort(): Sắp xếp các phần tử trong list theo thứ tự tăng dần.

sorted(): Cho phép hiển thị danh sách theo một thứ tự cụ thể nhưng không ảnh hưởng đến thứ tự thực tế của danh sách

Sử dụng các phương thức với danh sách

Dạng lệnh: <danh sách>.<tên phương thức>

Giải thích: <danh sách> là tên danh sách đã khai báo, <tên phương thức> là tên các phương thức cần sử dụng, dấu . là toán tử.

Làm việc với danh sách kiểu dãy (mảng 1 chiều)

Dãy là tập hợp các phần tử cùng kiểu được xác định thông qua chỉ mục (index). Dãy là kiểu danh sách 1 chiều. Mọi thao tác với danh sách đều có thể áp dụng cho dãy.

Dãy gồm n phần tử được đánh số từ 0 đến $n-1$. Truy cập phần tử thứ i của dãy: $\langle \text{Tên dãy} \rangle[i]$. Các yếu tố cần xác định khi làm việc với dãy gồm Tên dãy, Số lượng phần tử của dãy; Cách khai báo biến dãy và Nhập/xuất dãy.

Khai báo dãy tương tự khai báo danh sách.

Nhập dãy từ bàn phím: Nhập dãy từ bàn phím bằng cách thực hiện nhập lần lượt từng phần tử của dãy.

Xuất dãy ra màn hình: Xuất dãy ra màn hình thực hiện xuất từng phần tử của dãy.

Làm việc với danh sách kiểu mảng (mảng 2 chiều)

Bảng là một danh sách kiểu dãy mà mỗi phần tử là một kiểu dãy có số lượng phần tử như nhau. Mọi thao tác với danh sách đều có thể áp dụng cho bảng.

Một bảng gồm m hàng được đánh số từ 0 đến $m-1$. Một bảng gồm n cột được đánh số từ 0 đến $n-1$. Truy cập phần tử ở hàng i , cột j của bảng: $\langle \text{Tên bảng} \rangle[i][j]$

Các yếu tố cần xác định khi làm việc với bảng gồm Tên bảng; Số lượng hàng và cột của bảng; Cách khai báo biến bảng và Nhập/xuất bảng.

Khai báo bảng tương tự khai báo danh sách:

Khai báo bảng rỗng: $\langle \text{Tên bảng} \rangle = []$

Khai báo bảng có gán giá trị: $\langle \text{Tên bảng} \rangle = [\langle \text{Dãy 1} \rangle, \langle \text{Dãy 2} \rangle, \dots]$
trong đó, $\langle \text{Dãy } \dots \rangle$ là kiểu dãy có cùng số lượng phần tử.

Các bước nhập bảng từ bàn phím:

(1) Nhập số m nguyên dương là số lượng hàng của bảng.

(2) Nhập số n nguyên dương là số lượng cột của bảng

(3) Khai báo bảng là danh sách rỗng: $\langle \text{Tên bảng} \rangle = []$

(4) Khi nhập mỗi hàng i từ 0 đến $m-1$ cần thực hiện:

(4,1) Nhập lần lượt các phần tử từ 0 đến $n-1$ để tạo thành dãy hang_i

(4,2) Bỏ sung hang_i vào bảng sử dụng phương thức:

$\langle \text{Tên bảng} \rangle.append(\text{hang}_i)$

Yêu cầu xuất bảng ra màn hình:

Xuất từng phần tử của bảng theo hàng i từ 0 đến $m-1$ và cột j từ 0 đến $n-1$.
Mỗi hàng của bảng in trên 1 dòng.

Các bước tạo dãy gồm các số trong phạm vi cho trước

- (1) Thông báo sử dụng module random: `import random`.
- (2) Xác định giá trị n là số lượng phần tử của dãy.
- (3) Xác định phạm vi giá trị của các phần tử của dãy là từ $gt1$ đến $gt2$.
- (4) Sử dụng hàm `randint(gt1, gt2)` để tạo từng phần tử của dãy gồm n số nguyên: $x = random.randint(gt1, gt2)$.
- (5) Sử dụng hàm `uniform(gt1, gt2)` để tạo từng phần tử của dãy gồm n số thực: $x = random.uniform(gt1, gt2)$.

2.3.3 Nhóm nội dung 3

Các nội dung nhóm 3 nhằm trang bị cho học sinh những thao tác làm việc với kiểu dữ liệu có cấu trúc chuỗi ký tự (*string*).

Khái niệm

Chuỗi ký tự (*string* - chuỗi) là dãy các ký tự trong bảng mã Unicode, mỗi ký tự gọi là một phần tử của chuỗi. Số lượng ký tự trong chuỗi được gọi là độ dài của chuỗi. Chuỗi rỗng có độ dài bằng 0.

Hằng chuỗi là một chuỗi ký tự chứa trong hai dấu ngoặc kép “ ” hoặc hai dấu nháy trên ‘ ’. Khi trong hằng chuỗi có chứa các dấu nháy trên thì nên sử dụng hai dấu ngoặc kép để tránh nhầm lẫn. Ví dụ: $x = \text{"I'm a student"}$.

Trong Python không có kiểu dữ liệu ký tự, mỗi ký tự có thể xem như chuỗi gồm chính ký tự đó. Có thể xác định số thứ tự của ký tự x trong bảng mã Unicode qua hàm `ord('x')`. Ví dụ: `ord('A')= 65`, `ord('Â')= 194`, `ord('Ê')= 202`, `ord('0')= 48`.

Có thể xem chuỗi ký tự như List một chiều mà mỗi phần tử là một ký tự. Các ký tự của chuỗi đánh số thứ tự bắt đầu từ 0. Truy cập tới phần tử của chuỗi được xác định bởi tên biến chuỗi và chỉ số đặt trong cặp ngoặc `[]`.

Khai báo chuỗi ký tự

Khai báo chuỗi ký tự rỗng: `<Tên biến> = ""`

Khai báo chuỗi ký tự có giá trị hằng chuỗi: $\langle \text{Tên biến} \rangle = \langle \text{Dãy ký tự} \rangle$

Các thao tác trên chuỗi ký tự

Nhập chuỗi ký tự từ bàn phím

Dạng lệnh: $\langle \text{Tên biến} \rangle = \text{input}()$

Công dụng: Gán cho $\langle \text{Tên biến} \rangle$ các ký tự gõ từ bàn phím

Xuất chuỗi ký tự ra màn hình

Dạng lệnh: $\text{print}(\langle \text{Tên biến} \rangle)$

Công dụng: Xuất chuỗi ký tự $\langle \text{Tên biến} \rangle$ ra màn hình

Lệnh gán chuỗi trên một dòng: $\langle \text{Tên biến} \rangle = \langle \text{Các ký tự} \rangle$

Lệnh gán chuỗi trên nhiều dòng: $\langle \text{Tên biến} \rangle = \langle \text{Các dòng ký tự} \rangle$

Các phép toán trên chuỗi ký tự

Phép ghép chuỗi

Dạng lệnh: $\text{str} = \text{str1} + \text{str2}$

Công dụng: Ghép chuỗi str2 vào sau chuỗi str1 để nhận được chuỗi mới str. Các chuỗi str1, str2 có thể là các hằng chuỗi và biến chuỗi.

Hàm xác định chiều dài chuỗi

Dạng hàm: $\text{len}(\langle \text{Chuỗi} \rangle)$

Công dụng: Cho giá trị là số lượng ký tự của $\langle \text{Chuỗi} \rangle$.

Phép so sánh chuỗi ký tự

Sử dụng thứ tự từ điển. Các phép so sánh hai chuỗi ký tự s1 và s2 gồm bằng (==), khác (!=), nhỏ hơn (<), lớn hơn (>), nhỏ hơn hoặc bằng (<=), lớn hơn hoặc bằng (>=) thực hiện theo thứ tự từ điển:

Chuỗi s1 lớn hơn chuỗi s2 nếu như ký tự đầu tiên khác nhau giữa chúng kể từ trái sang trong s1 có mã Unicode lớn hơn.

Nếu s1 và s2 có độ dài khác nhau và s1 là đoạn đầu của s2 thì $s1 < s2$.

Các phép so sánh hai chuỗi có thứ tự ưu tiên thực hiện thấp hơn phép ghép chuỗi,

Một số hàm xử lý chuỗi ký tự

Hàm in: sc in sm

Nếu chuỗi sc chứa trong chuỗi sm hàm cho giá trị True

Hàm lấy xâu con: $sc = sm[<giá trị đầu>:<giá trị cuối>:<bước nhảy>]$

Xâu con sc gồm các kí tự của xâu mẹ sm bắt đầu bởi $<giá trị đầu>$, cách $<bước nhảy>$ cho đến $<giá trị cuối>$.

Hàm chuyển từ số sang xâu: $<tên biến> = str(<giá trị>)$

Số $<giá trị>$ sẽ chuyển sang xâu số $<tên biến>$.

Hàm chuyển xâu số sx thành giá trị số x :

$x = int(sx)$ # Chuyển sang số nguyên

$x = float(sx)$ # chuyển sang số thực

Một số phương thức xử lí xâu

Hàm strip() xóa dấu cách bên trái, bên phải xâu st : $<tên biến> = st.strip()$

Hàm chuyển kí tự in hoa upper(): $<tên biến> = <tên xâu>.upper()$

Hàm chuyển kí tự in thường lower(): $<tên biến> = <tên xâu>.lower()$

Hàm căn lề trái ljust(), phải rjust() và giữa center():

$<tên biến> = <tên xâu>.<hàm căn lề>(<số lượng>, ' ')$

Trong đó, xâu $<tên biến>$ sẽ được bổ sung thêm số $<số lượng>$ dấu cách rồi căn lề tương ứng.

2.3.4 Nhóm nội dung 4

Các nội dung nhóm 4 nhằm trang bị cho học sinh những thao tác làm việc với kiểu dữ liệu có cấu trúc tệp (*file*).

Khái niệm

Tất cả các dữ liệu có các kiểu dữ liệu đã xét đều được lưu trữ ở bộ nhớ trong (RAM) và do đó dữ liệu sẽ bị mất khi tắt máy. Với một số bài toán, dữ liệu cần được lưu trữ để xử lí nhiều lần và với khối lượng lớn sẽ sử dụng dữ liệu kiểu tệp.

Dữ liệu kiểu tệp là tập hợp các phần tử dữ liệu được lưu trữ trên bộ nhớ ngoài dưới dạng tệp tin (*file*). Do đó, dữ liệu không bị mất khi tắt nguồn điện. Lượng thông tin lưu trữ trên tệp có thể rất lớn và chỉ phụ thuộc vào dung lượng đĩa.

Các yếu tố của dữ liệu kiểu tệp bao gồm: Tên tệp, Kiểu tệp, Cách ghi dữ liệu ra tệp và Cách đọc dữ liệu từ tệp.

Có hai kiểu tệp thường sử dụng trong Python là tệp dữ liệu kiểu văn bản và tệp dữ liệu kiểu nhị phân. Đối với học sinh phổ thông chỉ cần làm việc với tệp văn bản.

Các tệp dữ liệu được đặt tên tương tự như trong các hệ điều hành gồm phần tên (name) và phần mở rộng (extension). Thông thường các tệp dữ liệu vào có đuôi .INP; các tệp kết quả có đuôi là .OUT. Trong trường hợp cần thiết phải chỉ rõ đường dẫn (path) đến tệp.

Làm việc với tệp

Làm việc với tệp gồm các bước : Mở tệp; Đọc hoặc ghi dữ liệu; Đóng tệp.

Ghi dữ liệu ra tệp

Trong Python xử lý các dữ liệu dưới dạng chuỗi (xâu kí tự) nên chỉ có thể ghi chuỗi ra tệp. Muốn ghi số x ra tệp cần chuyển x ra chuỗi: str(x).

Mở tệp để ghi

Dạng lệnh: `<Tên biến tệp> = open(<Tên tệp>, 'w')`

Trong đó, <Tên biến tệp> và <Tên tệp> do NLT tự đặt.

Công dụng: Mở tệp văn bản <Tên biến tệp> có tên <Tên tệp> để ghi các chuỗi

Ghi chú: <Tên tệp> có thể bao gồm cả đường dẫn. Nếu <Tên tệp> trùng với tệp đã có thì nội dung của tệp sẽ bị xóa. Thông thường tệp mới với <Tên tệp> sẽ được tạo trong thư mục chứa chương trình có lệnh mở tệp để ghi.

Ghi dữ liệu ra tệp đã mở

Dạng lệnh: `<Tên biến tệp>.write(s)`

Trong đó, <Tên biến tệp> là tệp đã mở để ghi, s là chuỗi.

Công dụng: Ghi chuỗi s ra tệp <Tên biến tệp>.

Ghi chú: Nếu s là số thì cần chuyển thành str(s). Nếu muốn chuyển sang dòng tiếp theo sử dụng lệnh `<Tên biến tệp>.write('\n')`. Có thể ghi s theo định dạng bởi hàm format().

Đóng tệp đã ghi

Dạng lệnh: `<Tên biến tệp>.close()`

Trong đó, <Tên biến tệp> là tệp đã mở để ghi.

Công dụng: Đóng tệp <Tên biến tệp>.

Ghi chú: Nếu không đóng tệp thì dữ liệu đã ghi vào tệp <Tên biến tệp> có thể không được lưu. Trong CT bắt buộc phải có lệnh đóng tệp.

Đọc dữ liệu từ tệp

Trong Python xử lý các dữ liệu dưới dạng chuỗi (xâu kí tự) nên chỉ có thể đọc dữ liệu từ tệp dưới dạng chuỗi. Các dữ liệu có kiểu khác chuỗi cần chuyển đổi phù hợp.

Mở tệp để đọc

Dạng lệnh: `<Tên biến tệp> = open(<Tên tệp>, 'r')`

Trong đó, <Tên biến tệp> và <Tên tệp> do NLT tự đặt.

Công dụng: Mở tệp văn bản <Tên biến tệp> có tên <Tên tệp> để đọc các chuỗi

Ghi chú: <Tên tệp> có thể bao gồm cả đường dẫn. Phải có sẵn tệp <Tên tệp> để đọc, nếu không sẽ báo lỗi. Thông thường tệp với <Tên tệp> phải có trong thư mục chứa chương trình có lệnh mở tệp để đọc.

Đọc dữ liệu từ tệp đã mở

Dạng lệnh: `s = <Tên biến tệp>.readline()`

Trong đó, <Tên biến tệp> là tệp đã mở để đọc, s là biến chuỗi.

Công dụng: Đọc chuỗi từ tệp <Tên biến tệp> và gán cho s.

Ghi chú: Nếu s là số thì cần chuyển đổi tương ứng:

`s = int(<Tên biến tệp>.readline())`

`s = float(<Tên biến tệp>.readline())`

Nếu s là danh sách (list) các số nguyên sử dụng lệnh chuyển đổi

`ds = list(map(int, s.split()))`

Nếu s là danh sách (list) các số thực sử dụng lệnh chuyển đổi

`ds = list(map(float, s.split()))`

Đóng tệp đã đọc

Dạng lệnh: `<Tên biến tệp>.close()`

Trong đó, <Tên biến tệp> là tệp đã mở để đọc.

Công dụng: Đóng tệp <Tên biến tệp>..

Sử dụng lệnh with

Có thể sử dụng lệnh `with` để đồng thời mở tệp, ghi/đọc dữ liệu và đóng tệp. Khi sử dụng lệnh `with` sẽ tránh gây lỗi khi ghi/đọc dữ liệu.

Sử dụng lệnh `with` để ghi ra tệp

Dạng lệnh: `with open(<Tên tệp>, 'w') as <Tên biến tệp>:`

`<Tên biến tệp>.write(s)`

Trong đó, `<Tên tệp>`, `<Tên biến tệp>` do NLT tự chọn, `s` là chuỗi.

Công dụng: Mở, ghi chuỗi `s` ra tệp `<Tên biến tệp>` và đóng tệp.

Sử dụng lệnh `with` để đọc từ tệp

Dạng lệnh: `with open(<Tên tệp>, 'r') as <Tên biến tệp>:`

`s = <Tên biến tệp>.readline()`

Trong đó, `<Tên tệp>`, `<Tên biến tệp>` do NLT tự chọn, `s` là biến chuỗi.

Công dụng: Mở, đọc và gán giá trị từ tệp `<Tên biến tệp>` cho biến chuỗi `s` và đóng tệp.

2.3.5 Nhóm nội dung 5

Các nội dung nhóm 5 nhằm trang bị cho học sinh những kiến thức để xây dựng và sử dụng chương trình con.

Khái niệm

Chương trình con (CTC) là một dãy lệnh mô tả một số thao tác nhất định và có thể được thực hiện (được gọi) từ nhiều vị trí trong chương trình. CTC giúp có thể chia một chương trình lớn thành những mô đun, khiến chương trình dễ hiểu, dễ duy trì và gỡ lỗi hơn. Các lập trình viên cùng làm việc trong một dự án lớn, có thể phân chia công việc cho nhau bằng cách tạo các CTC khác nhau. CTC có thể tái sử dụng trong nhiều chương trình khác nhau.

Trong Python CTC được mô tả dưới dạng hàm (Function). CTC có thể trả về giá trị cụ thể và gọi là CTC dạng hàm. Khi đó trong CTC phải có lệnh `return(<Giá trị>)`. CTC có thể không trả về giá trị cụ thể và gọi là CTC dạng thủ tục.

Các thao tác trên CTC gồm (1) Xây dựng CTC; (2) Sử dụng CTC. Khi làm việc với CTC cần quan tâm đến các yếu tố sau đây.

Từ khóa: *def* được sử dụng để khai báo CTC.

Tên CTC: Do NLT tự đặt như tên hằng hoặc biến.

Danh sách tham số/đối số: là các tên biến được đặt trong dấu ‘()’ và cách nhau dấu ‘,’. Danh sách có thể rỗng.

Dấu ‘:’ dùng để chỉ bắt đầu các lệnh của CTC.

docstring: Một đoạn văn bản để mô tả chức năng của CTC được chứa trong các dấu ‘””””. Có thể không có.

Các lệnh: CTC chứa các lệnh, trong đó thường chứa lệnh *return()* để ra khỏi hàm và trả về giá trị hàm.

Xây dựng chương trình con

def <TenCTC>(*DoiSo1*, *DoiSo2*, ..., *DoiSon*):

“””” Các dòng văn bản “”””

<Khởi lệnh của CTC>

Sử dụng chương trình con

Để sử dụng CTC, nhập lời gọi CTC dạng:

<*Tên CTC*>(*GiaTri1*, *GiaTri2*, ..., *GiaTrin*)

Trong đó *GiaTri1*, *GiaTri2*, ..., *GiaTrin* là các hằng và biến phù hợp.

Biến toàn cục và biến địa phương

Biến toàn cục là biến có thể được dùng mọi nơi trong chương trình kể từ khi biến đó xuất hiện. Biến toàn cục là biến xuất hiện và sử dụng bên ngoài các CTC. Nếu muốn biến khai báo trong CTC trở thành biến toàn cục thì phải thêm từ khóa *global*. Biến toàn cục là biến được sử dụng mọi nơi kể cả trong CTC.

Biến địa phương là biến chỉ có thể được dùng trong các chương trình con. Khi ra khỏi CTC các biến địa phương sẽ không còn tồn tại. Do đó, nếu sử dụng sẽ gây lỗi.

Chương trình con đệ quy

CTC đệ quy là CTC có chứa lệnh gọi đến chính nó.

CTC đề quy thường bao gồm 2 phần: phần tĩnh (không thay đổi) và phần động (thay đổi).

Phần tĩnh: bao gồm các lệnh, trong đó có lệnh điều kiện ra khỏi CTC.

Phần động: Lời gọi đến CTC nhưng cỡ của đối số phải thay đổi.

2.3.6 Nhóm nội dung 6

Các nội dung nhóm 6 nhằm trang bị cho học sinh những kiến thức và thao tác giải quyết bài toán sắp xếp và tìm kiếm dữ liệu. Đây là các bài toán cơ bản cần giải quyết của CNTT. Nhóm nội dung 6 có thể xem như phần tổng hợp và nâng cao để vận dụng lập trình Python giải quyết các bài toán cụ thể.

Trong Python đã có sẵn các hàm và phương thức sắp xếp danh sách. Trước hết cần hướng dẫn học sinh hiểu và sử dụng thành thạo các thao tác sắp xếp danh sách tăng/ giảm hoặc theo tiêu chí tăng/ giảm của thành phần nào đó trong danh sách. Tuy vậy, do tính đa dạng của bài toán sắp xếp, cần hướng dẫn học sinh các thuật toán sắp xếp.

Các thuật toán sắp xếp

Luận văn đề xuất hướng dẫn cho học sinh phổ thông 4 thuật toán sắp xếp cơ bản sau đây:

Sắp xếp kiểu lựa chọn (Selection Sort),

Sắp xếp kiểu xen vào (Insertion Sort)

Sắp xếp kiểu nổi bọt (Bubble Sort),

Sắp xếp nhanh (Quick Sort)

Bài toán tìm kiếm dữ liệu

Input: Tập dữ liệu D và yêu cầu (P) ;

Output: Xác định phần tử $x \in D$ thỏa mãn (P) hoặc thông báo không có phần tử nào của D thỏa mãn (P) .

Phương pháp duyệt toàn thể giải quyết bài toán tìm kiếm

Bước 1: Xác định phạm vi tìm kiếm

1.1 Xác định tập $X \subseteq D$ là tập con của tập D sao cho các phần tử của tập $D \setminus X$ không thỏa mãn (P) .

1.2 Sắp xếp các phần tử $x \in X$ theo một thứ tự phù hợp.

Bước 2: Thực hiện duyệt tất cả các phần tử $x \in X$

2.1 Nếu x thỏa mãn (P) thì ghi nhận x là kết quả tìm kiếm.

2.2 Nếu x không thỏa mãn thì bỏ qua.

Bước 3: Cài đặt chương trình

3.1 Đọc dữ liệu vào theo khuôn dạng của bài toán

3.2 Thực hiện tìm kiếm theo bước 2

3.3 Xuất kết quả ra theo khuôn dạng của bài toán.

Phương pháp tìm kiếm nhị phân

Bước 1: Xác định phạm vi tìm kiếm

1.1 Xác định tập $X \subseteq D$ là tập con của tập D sao cho các phần tử của tập $D \setminus X$ không thỏa mãn (P).

1.2 Xác định khoảng tìm kiếm $X_{\min} \leq (x \in X) \leq X_{\max}$.

Bước 2: Thực hiện lặp khi ($X_{\min} < X_{\max}$)

2.1 Xác định mốc tìm kiếm $m = (X_{\min} + X_{\max})/2$

2.2 Nếu m thỏa mãn (P) thì ghi nhận m và dừng;

2.3 Thay đổi phạm vi tìm kiếm: $X_{\min} = m + 1$ hoặc $X_{\max} = m - 1$;

Bước 3: Cài đặt chương trình

3.1 Đọc dữ liệu vào theo khuôn dạng của bài toán

3.2 Thực hiện tìm kiếm theo bước 2

3.3 Xuất kết quả ra theo khuôn dạng của bài toán.

Các kiến thức của nhóm nội dung 6 sẽ được dạy cho học sinh kết hợp với hướng dẫn học sinh giải các bài tập ở phần 5 và phần 6 trong hệ thống bài tập được trình bày trong phần phụ lục.

2.4 Giải pháp công nghệ bảo đảm dạy lập trình Python cho học sinh phổ thông

Để quá trình dạy và học lập trình cho học sinh đạt hiệu quả cần đảm bảo một số yêu cầu kỹ thuật. Trong mục này, luận văn khảo sát ba yêu cầu cơ bản là yêu cầu về phần mềm tạo môi trường cài đặt Python và yêu cầu về phòng máy tính. Đồng

thời, luận văn cũng đề xuất giải pháp triển khai ứng dụng E-Learning cho dạy và học lập trình Python.

2.4.1 Yêu cầu về phần mềm cài đặt Python

Môi trường tích hợp dùng để phát triển (IDE - Integrated Development Environment) đóng vai trò quan trọng trong quá trình NLT viết chương trình Python nhanh chóng và hiệu quả. Hiện nay có khá nhiều IDE cho lập trình Python phù hợp với nhiều đối tượng khác nhau như:

- PyDev - Python IDE miễn phí từ Eclipse. Eclipse Foundation là một tổ chức phi lợi nhuận độc lập quản lý cộng đồng nguồn mở Eclipse. ...
- Pycharm - Python IDE tuyệt vời từ JetBrains. ...
- Visual Studio Code. ...
- Sublime Text. ...
- Atom / Atom IDE.

Nhưng trong luận văn, học viên đề xuất sử dụng IDLE Python (Python's Integrated Development and Learning Environment) để dạy lập trình cho học sinh phổ thông vì IDLE Python có các ưu điểm sau [10]:

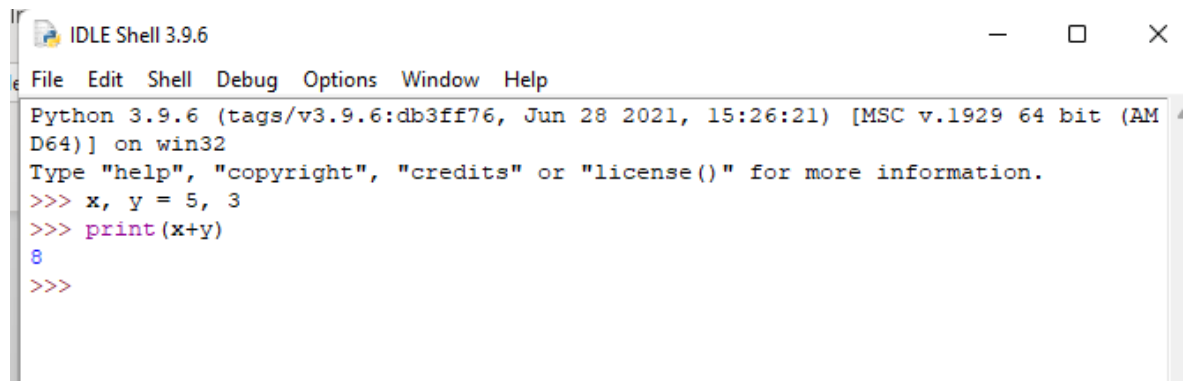
- IDLE Python được mã hóa bằng Python thuần túy 100%, sử dụng bộ công cụ tkinter GUI.
- IDLE Python là đa nền tảng, có thể hoạt động hầu như giống nhau trên Windows, Unix và macOS.
- Cửa sổ shell Python (trình thông dịch tương tác) phân biệt màu sắc của đầu vào, đầu ra và thông báo lỗi.
- Trình soạn thảo văn bản của IDLE Python có nhiều cửa sổ với nhiều lần hoàn tác, tô màu Python, thực lệ thông minh, mẹo gọi, tự động hoàn thành và các tính năng khác.
- Trong IDLE Python có thể tìm kiếm trong mọi cửa sổ, có thể thay thế trong cửa sổ trình soạn thảo và tìm kiếm qua nhiều tệp (grep).

- Trong IDLE Python trình gỡ lỗi với các điểm ngắt liên tục, theo từng bước và có thể xem không gian tên toàn cục và cục bộ.

- Có thể cấu hình trình duyệt và các hộp thoại khác của IDLE Python.

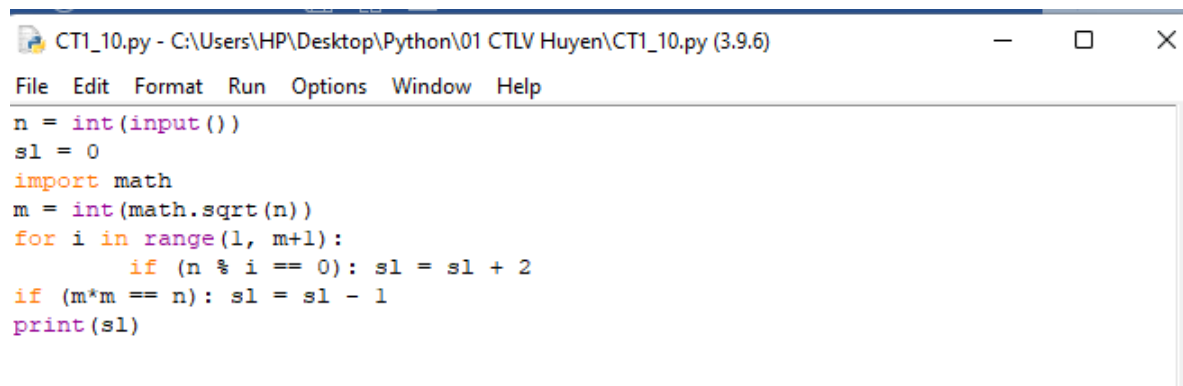
Do đó, IDLE Python thường được khuyến nghị sử dụng cho những người mới bắt đầu học lập Python.

Hình 2.3 mô tả cửa sổ thực hiện trực tiếp các câu lệnh của IDLE Python.



Hình 2.3: Cửa sổ thực hiện các câu lệnh trực tiếp của IDLE Python.

Hình 2.4 mô tả cửa sổ soạn thảo và chạy chương trình của IDLE Python.



Hình 2.4: Cửa sổ soạn thảo và thực hiện chương trình của IDLE Python.

Có thể tải miễn phí IDLE Python tại Link:

<https://www.python.org/downloads/>

Phiên bản IDLE Python nên sử dụng là 3.x và lựa chọn phù hợp với cấu hình máy tính.

2.4.2 Yêu cầu về phòng máy tính

Quá trình dạy và học lập trình cho học sinh phổ thông thường diễn ra tại các phòng máy tính của nhà trường. Do đó, việc đảm bảo các phòng máy tính phù hợp có vai trò quan trọng trong quá trình dạy lập trình Python cho học sinh phổ thông.

Theo các qui định hiện hành và thực tế tại các trường phổ thông, có thể đặt ra yêu cầu cho phòng máy tính dạy lập trình Python như sau:

(1) Về số lượng máy tính:

Lý tưởng nhất là phòng máy tính có đủ cho mỗi học sinh 1 máy tính trong quá trình dạy lập trình Python. Trong trường hợp khó khăn, cần đảm bảo 1 máy tính cho 2 học sinh.

(2) Cấu hình máy tính:

Các máy tính trong phòng máy dạy và học lập trình Python cần đảm bảo có thể cài đặt được Windows và IDLE Python. Các máy tính nên có ổ cứng cho phép học sinh làm việc với tệp dữ liệu khi học Python.

(3) Cài đặt phần mềm:

Các máy tính của phòng máy cần được cài đặt sẵn IDLE Python và thường xuyên cập nhật để tránh bị lỗi.

2.4.3 Giải pháp ứng dụng E-Learning

E-Learning (viết tắt của *Electronic Learning*) là một công nghệ để triển khai quá trình dạy và học dựa trên công nghệ thông tin và truyền thông. Vì vậy, giải pháp sử dụng công nghệ E-Learning để hỗ trợ quá trình dạy và học lập trình Python cho học sinh phổ thông là phù hợp, đặc biệt là trong thời kỳ dịch bệnh học sinh phải học trực tuyến.

Hệ thống E-Learning được triển khai để dạy và học lập trình Python cho học sinh phổ thông cần bao gồm các thành phần chính là hệ thống nội dung đào tạo, hệ thống hạ tầng phân phối, quản lý đào tạo và hệ thống tương tác, đánh giá.

(1) Hệ thống nội dung

Các nội dung đào tạo lập trình Python được xây dựng thành các bài giảng điện tử được thể hiện thông qua các phương tiện truyền thông điện tử, đa phương tiện. Luận văn đề xuất lựa chọn chuẩn SCORM (Sharable Content Object Reference Model) để xây dựng và đóng gói bài giảng điện tử về lập trình Python.

(2) Hệ thống hạ tầng

Hệ thống hạ tầng bao gồm hệ thống quản lý nội dung dạy và học Python, Hệ thống quản lý dạy và học Python và hạ tầng truyền thông và mạng.

Hệ thống quản lý nội dung học tập (LCMS - Learning Content Management System) là một môi trường đa người dùng cho phép giáo viên và cơ sở đào tạo kết hợp để tạo ra, lưu trữ, sử dụng lại, quản lý và phân phối nội dung bài giảng điện tử từ một kho dữ liệu trung tâm. Để cung cấp khả năng tương thích giữa các hệ thống, LCMS được thiết kế sao cho phù hợp với các tiêu chuẩn về siêu dữ liệu nội dung, đóng gói nội dung và truyền thông nội dung.

Hệ thống quản lý học tập (LMS – Learning Management System) là một hệ thống dịch vụ hỗ trợ và quản lý quá trình học tập của học sinh.

Hạ tầng truyền thông và mạng cung cấp môi trường triển khai E-Learning. Luận văn đề xuất sử dụng hệ thống web của trường phổ thông để triển khai E-Learning cho dạy và học Python.

(3) Hệ thống đánh giá

Hệ thống đánh giá hỗ trợ cho quá trình dạy và học Python nhằm nâng cao chất lượng dạy và học. Luận văn đề xuất xây dựng hệ thống đánh giá tự động kết quả lập trình của học sinh phổ thông. Trong hệ thống này, các bài tập lập trình Python với các bộ test mẫu được cài đặt sẵn có vai trò hết sức quan trọng

2.5 Kết luận chương 2

Trong chương 2 luận văn đã trình bày tổng quan về lịch sử hình thành và các đặc điểm của ngôn ngữ lập trình Python, tiến hành khảo sát các yếu tố hình thành của ngôn ngữ lập trình Python. Trên cơ sở đó, luận văn đã đề xuất lựa chọn và trình bày 6 nội dung dạy lập trình Python phù hợp cho học sinh phổ thông.

Để đảm bảo tính hiệu quả trong quá trình dạy và học lập trình Python, luận văn đã đề xuất sử dụng IDLE Python để dạy và học Python. Đồng thời luận văn đã khuyến nghị một số yêu cầu đối với phòng máy tính và ứng dụng công nghệ E-Learning trong dạy và học lập trình Python.

Trên cơ sở các nội dung nghiên cứu của chương 2, trong chương 3 luận văn sẽ thực hiện xây dựng hệ thống bài tập lập trình trên Python cho học sinh phổ thông.

CHƯƠNG 3-KẾT QUẢ THỰC HIỆN XÂY DỰNG HỆ THỐNG BÀI TẬP LẬP TRÌNH CHO HỌC SINH PHỔ THÔNG

Chương 3 của luận văn sẽ khảo sát các yêu cầu đối với hệ thống bài tập lập trình cho học sinh phổ thông nhằm hỗ trợ rèn luyện kỹ năng lập trình. Từ đó luận văn sẽ nghiên cứu, xây dựng hệ thống bài tập lập trình trên Python, đề xuất phương pháp hướng dẫn học sinh giải bài tập, phương pháp đánh giá chương trình Python của học sinh và khảo sát các vấn đề liên quan.

3.1 Các yêu cầu đối với hệ thống bài tập lập trình cho học sinh phổ thông

Hệ thống bài tập lập trình trên Python cho học sinh cần đảm bảo các yêu cầu về tính khoa học và tính sư phạm [3].

3.1.1 Các yêu cầu về tính khoa học

Trong quá trình dạy và học lập trình cho học sinh phổ thông, hệ thống bài tập sẽ giúp học sinh hiểu và vận dụng kiến thức lập trình đã học trong lý thuyết. Trong một số trường hợp, thông qua các bài tập, giáo viên có thể hướng dẫn thêm những phần kiến thức khó trong dạy lập trình như CTC đệ quy, những thuật toán cơ bản giải các bài toán sắp xếp và tìm kiếm dữ liệu.

Luận văn đề xuất một số yêu cầu về tính khoa học của hệ thống bài tập cần xây dựng như sau.

- Hệ thống phải có các bài tập tương ứng với cả 6 nhóm nội dung đề xuất dạy cho học sinh phổ thông đã trình bày trong mục 2.3. Hệ thống bài tập phải hỗ trợ học sinh nắm được các kịch bản khác nhau khi sử dụng các câu lệnh và viết chương trình Python.

- Theo công thức “**Chương trình = Cấu trúc dữ liệu + Thuật toán**”, hệ thống bài tập phải cung cấp cho học sinh các thuật toán cơ bản thông qua quá trình giải bài tập. Do đó, hệ thống bài tập phải có các bài tập mang tính chất điển hình để sau khi được hướng dẫn giải các bài tập này học sinh có thể vận dụng giải một lớp các bài toán khác tương tự và thường gặp trong CNTT.

- Hệ thống bài tập cần bao gồm các bài tập phù hợp cho đối tượng học sinh trung bình, khá và giỏi tương ứng với các mức độ vận dụng kiến thức lập trình từ dễ đến khó, từ chương trình đơn giản đến các chương trình có cấu trúc phức tạp.

- Các bài tập tổng hợp và nâng cao trong hệ thống bài tập cần được trình bày theo đúng khuôn dạng của bài toán CNTT, trong đó cần nêu rõ yêu cầu về thời gian chạy chương trình, cỡ dữ liệu đầu vào, khuôn dạng nhập dữ liệu và xuất kết quả. Đặc biệt cần lưu ý là trong các bài tập phải có ví dụ số cụ thể và tiêu biểu giúp học sinh hiểu yêu cầu của bài tập. Do hạn chế khuôn khổ của luận văn, một số bài tập đơn giản có thể không kèm theo ví dụ số. Khi sử dụng trong quá trình dạy lập trình Python, giáo viên nên bổ sung các ví dụ số phù hợp.

- Hệ thống bài tập cần xây dựng có thể sử dụng cho dạy lập trình trên các ngôn ngữ lập trình khác. Đồng thời, trong quá trình sử dụng, giáo viên và học sinh có thể sử dụng lời thuật toán để có thể biên tập thành các bài tập khác nhau.

3.1.2 Các yêu cầu về tính sư phạm

Do đối tượng sử dụng hệ thống bài tập là giáo viên và học sinh phổ thông nên khi xây dựng hệ thống bài tập cần đảm bảo các yêu cầu về tính sư phạm.

Luận văn đề xuất một số yêu cầu về tính sư phạm của hệ thống bài tập cần xây dựng như sau.

- Các bài tập trong hệ thống đảm bảo tính vừa sức với học sinh, phát triển các mạch kiến thức vừa theo đường thẳng vừa đồng tâm, kết hợp trừu tượng với trực quan. Do đó, các bài tập phải bao gồm các bài tập phải được sắp xếp theo mức độ yêu cầu tăng dần.

- Các bài tập trong hệ thống phải được trình bày rõ ràng, logic, tránh dùng quá nhiều các thuật ngữ chuyên môn quá phức tạp.

- Hệ thống bài tập phải gây hứng thú học lập trình cho học sinh phổ thông. Đồng thời, giúp giáo viên dễ dàng triển khai khi dạy cho học sinh phổ thông. Do đó, các bài tập được lựa chọn nên có nhiều cách giải hoặc có thể giải ở nhiều cấp độ, từ duyệt toàn thể cho đến sử dụng các thuật toán hữu hiệu.

3.2 Xây dựng hệ thống bài tập lập trình trên Python

Căn cứ các yêu cầu đối với hệ thống bài tập lập trình và các nội dung Python được đề xuất dạy cho học sinh phổ thông ở mục 2.3, trong mục này luận văn tiến hành xây dựng hệ thống bài tập lập trình trên Python.

Hệ thống bài tập lập trình trên Python được xây dựng gồm 100 bài tập, được phân chia thành 6 phần. Do giới hạn số trang của luận văn, hệ thống bài tập được trình bày chi tiết trong phần phụ lục.

Trong mục này, luận văn sẽ trình bày tổng quan về hệ thống bài tập đã xây dựng.

Phần 1: Các bài tập mở đầu

Phần 1 bao gồm 20 bài tập đánh số từ 1.1 đến 1.20. Các bài tập trong mục này hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 1 bao gồm các kiểu dữ liệu đơn giản chuẩn và các câu lệnh. Đặc điểm của các bài tập trong phần này là đơn giản, chỉ yêu cầu học sinh vận dụng trực tiếp các câu lệnh đã học. Các bài tập cũng liên quan đến các kiến thức học sinh đã học trong các môn học khác ở phổ thông. Yêu cầu cần đạt là học sinh viết đúng các câu lệnh và thực hiện từng câu lệnh cho kết quả phù hợp. Từ đó giúp học sinh bước đầu làm quen với việc tập hợp các câu lệnh để tạo thành chương trình hoàn chỉnh.

Phần 2: Các bài tập về danh sách, danh sách kiểu dãy và kiểu bảng

Phần 2 bao gồm 12 bài tập đánh số từ 2.1 đến 2.12. Các bài tập trong mục này hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 2 về dữ liệu có cấu trúc dạng danh sách (list) với hai dạng cụ thể kiểu dãy (mảng 1 chiều) và kiểu bảng (mảng hai chiều). Các bài tập trong phần này giúp học sinh rèn luyện các kỹ năng cơ bản khi làm việc với danh sách như nhập/xuất danh sách, truy cập các phần tử của danh sách và hai dạng kiểu dãy và kiểu bảng có nhiều ứng dụng thực tế. Yêu cầu cần đạt là học sinh viết đúng các câu lệnh để tạo thành chương trình hoàn chỉnh do lượng dữ liệu xử lý thường lớn. Có thể yêu cầu học sinh tạo ngẫu nhiên dãy và bảng để xử lý.

Phần 3: Các bài tập về chuỗi ký tự

Phần 3 bao gồm 8 bài tập đánh số từ 3.1 đến 3.8. Các bài tập trong mục này hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 3 về dữ liệu có cấu trúc dạng chuỗi ký tự (*string*). Các bài tập trong phần này giúp học sinh rèn luyện các kỹ năng cơ bản khi làm việc với chuỗi ký tự như đếm ký tự, đếm từ, tách số. Yêu cầu cần đạt là học sinh viết đúng các câu lệnh để tạo thành chương trình hoàn chỉnh và chạy ra kết quả phù hợp.

Phần 4: Các bài tập về dữ liệu có cấu trúc kiểu tệp

Phần 4 bao gồm 10 bài tập đánh số từ 4.1 đến 4.10. Các bài tập trong mục này hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 4 về dữ liệu có cấu trúc dạng tệp (*file*). Các bài tập trong phần này giúp học sinh rèn luyện các kỹ năng cơ bản khi làm việc với tệp như nhập/xuất dữ liệu theo khuôn dạng cho trước. Đồng thời các bài tập cũng yêu cầu học sinh vận dụng các kiến thức đã học ở các phần trước để lập trình. Yêu cầu cần đạt là học sinh viết đúng chương trình và xuất, nhập dữ liệu ra tệp theo đúng khuôn dạng qui định.

Phần 5: Các bài tập về chương trình con

Phần 5 bao gồm 10 bài tập đánh số từ 5.1 đến 5.10. Các bài tập trong mục này hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 5 về chương trình con. Đây là những nội dung khó đối với học sinh khi học lập trình. Vì thế, một số bài tập trong mục này chỉ yêu cầu học sinh sử dụng lại các chương trình đã có khi giải các bài tập ở 4 phần trên để viết thành CTC.

Các bài tập trong phần này giúp học sinh rèn luyện các kỹ năng cơ bản khi làm việc với CTC là xây dựng và sử dụng CTC. Một số bài tập trong mục này liên quan đến cài đặt các thuật toán sắp xếp dữ liệu trong nhóm nội dung 6.

Các bài tập cần xây dựng CTC đệ quy dành cho học sinh khá và giỏi.

Phần 6: Các bài tập tổng hợp và nâng cao

Phần 6 bao gồm 40 bài tập đánh số từ 6.1 đến 6.40. Các bài tập trong mục 6 hỗ trợ học sinh nắm vững các kiến thức trong nhóm nội dung 6 về sắp xếp và tìm kiếm dữ liệu. Đồng thời các bài tập ở mục này cũng giúp học sinh tổng hợp các kiến thức về Python và các thuật toán đơn giản để giải quyết các bài toán Tin học. Mục

này cũng cung cấp cho học sinh một tài liệu để rèn luyện và chuẩn bị cho các kỳ thi học sinh giỏi và Tin học trẻ các cấp. Học sinh được khuyến khích giải được càng nhiều các bài tập trong mục 6 càng tốt.

Các bài tập trong mục 6 cũng có thể được sử dụng trong quá trình dạy và học các ngôn ngữ lập trình khác.

3.3 Hướng dẫn sử dụng hệ thống bài tập

3.3.1 Phương pháp hướng dẫn học sinh giải bài tập

Trong mục này luận văn sẽ trình bày các khuyến nghị về phương pháp hướng dẫn học sinh giải bài tập trong hệ thống đã xây dựng.

Các bước giải bài tập trên máy tính [3]:

Bước 1: Xác định bài toán

Xác định bài toán là làm rõ Input/Output của bài toán đã cho. Trong đó:

Input: Xác định các đại lượng vào (đã cho) và các thuộc tính của chúng.

Output: Xác định các đại lượng ra (cần tìm) và các thuộc tính của chúng.

Bước 2: Lựa chọn hoặc thiết kế thuật toán

Trước hết cần tìm các ý tưởng biến đổi Input thành Output của bài toán. Từ đó, thiết kế thuật toán phù hợp về thời gian thực hiện, tài nguyên máy cần có, ... Nếu có nhiều thuật toán để giải bài toán thì lựa chọn thuật toán hiệu quả nhất hoặc phù hợp với ngữ cảnh của người sử dụng theo các tiêu chí: Thời gian thực hiện ngắn, Sử dụng ít ô nhớ, Thuật toán dễ hiểu, ít phức tạp, Tốn ít tài nguyên nhất có thể.

Bước 3: Viết chương trình

Sử dụng các câu lệnh phù hợp để viết chương trình Python phù hợp thuật toán đã thiết kế. Học sinh có thể tiến hành theo hai giai đoạn.

Giai đoạn 1: Học sinh viết và thực hiện từng câu lệnh để kiểm tra tính đúng đắn của chúng.

Giai đoạn 2: Học sinh tổng hợp các câu lệnh để tạo thành một chương trình hoàn chỉnh.

Bước 4: Hiệu chỉnh và đánh giá

Sau khi viết xong chương trình cần kiểm thử chương trình để phát hiện các lỗi và khắc phục. Quá trình này gọi là hiệu chỉnh và đánh giá chương trình.

Trong mục này thực hiện chi tiết hướng dẫn giải một số bài tập trong hệ thống đã xây dựng tập trung vào ba bước 1, 2 và 3. Bước sẽ được khảo sát ở mục tiếp theo.

Ví dụ 3.1 (Bài tập 1.10 - Phụ lục):

Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^9 . Xuất ra màn hình số lượng các ước nguyên dương của n .

Hướng dẫn giải

1) Xác định bài toán

Input: Số nguyên dương n không vượt quá 10^9 được nhập từ bàn phím;

Output: Xuất ra màn hình số lượng các ước nguyên dương của n ;

Nhân xét: Nếu số u là ước của n thì số n/u cũng là ước của n . Do đó, chỉ cần tìm các ước u của n trong phạm vi từ 1 đến \sqrt{n} . Cần lưu ý trường hợp n là số chính phương. Ví dụ, $n = 16$ có 3 cặp ước (1, 16), (2, 8), (4, 4), nhưng chỉ có 5 ước là 1, 2, 4, 8 và 16.

2) Thuật toán

(1) Nhập vào từ bàn phím số nguyên dương n ;

(2) Tính sl là số lượng các ước nguyên dương của n :

(2.1) Khởi tạo $sl = 0$

(2.2) Tính $m = \text{int}(n^{0.5})$ (hoặc sử dụng module math $m = \text{int}(\sqrt{n})$)

(2.3) Duyệt i từ 1 đến m : Nếu $n \% i = 0$ thì $sl = sl + 2$

(2.4) Xử lý trường hợp n chính phương: Nếu $m * m = n$ thì $sl = sl - 1$

(3) Xuất ra màn hình giá trị sl .

3) Cài đặt:

Cài đặt sử dụng câu lệnh lặp for:

Thực hiện từng câu lệnh trực tiếp:

```
>>> n = int(input())
```

Viết thành chương trình rồi thực hiện:

```
n = int(input())
```

```

16
>>> sl = 0
>>> import math
>>> m = int(math.sqrt(n))
>>> for i in range(1, m+1):
        if (n % i == 0): sl = sl + 2
>>> if (m*m == n): sl = sl - 1
>>> print(sl)
5
>>>

```

```

sl = 0
import math
m = int(math.sqrt(n))
for i in range(1, m+1):
    if (n % i == 0): sl = sl + 2
if (m*m == n): sl = sl - 1
print(sl)
16
5
>>>

```

Cài đặt sử dụng câu lệnh lặp while:

Thực hiện từng câu lệnh trực tiếp

```

>>> n = int(input())
16
>>> sl = 0
>>> import math
>>> m = int(math.sqrt(n))
>>> i = 1
>>> while (i <= m):
        if (n % i == 0): sl += 2
        i += 1
>>> if (m*m == n): sl -= 1
>>> print(sl)
5
>>>

```

Viết thành chương trình rồi thực hiện:

```

n = int(input())
sl = 0
import math
m = int(math.sqrt(n))
i = 1
while (i <= m):
    if (n % i == 0): sl += 2
    i += 1
if (m*m == n): sl -= 1
print(sl)
16
5
>>>

```

Ví dụ 3.2 (Bài tập 2.5 - Phụ lục):

Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^6 . Sử dụng module random, tạo dãy gồm n số nguyên có giá trị tuyệt đối không vượt quá 10^9 . Xuất ra màn hình các giá trị maxa và mind trên cùng 1 dòng tương ứng là giá trị lớn nhất của các số âm và giá trị nhỏ nhất trong các số dương của dãy. Nếu không tìm được max hoặc min thì ghi số 0 thay thế.

Hướng dẫn giải**1) Xác định bài toán****Input:**

- Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^6 ;
- Dãy (A) gồm n số nguyên bất kỳ có giá trị tuyệt đối không vượt quá 10^9 được tạo ngẫu nhiên;

Output: Xuất ra màn hình trên cùng 1 dòng giá trị lớn nhất maxa của các số âm và giá trị nhỏ nhất mind của các số dương có trong dãy (A). Nếu không tìm được max hoặc min thì ghi số 0 thay thế;

2) Thuật toán

- (1) Nhập vào từ bàn phím số nguyên dương n ;
- (2) Sử dụng module random tạo ngẫu nhiên dãy $a[]$ gồm n số nguyên bất kỳ có giá trị tuyệt đối không vượt quá 10^9 ;
- (3) Tính các giá trị maxa và mind:
 - (3.1) Khởi tạo $\text{maxa} = -10^{**9} - 1$; $\text{mind} = 10^{**9} + 1$;
 - (3.2) Duyệt các phần tử i của $a[]$ trong phạm vi n :
 - Nếu ($a[i] < 0$ và $a[i] > \text{maxa}$) thì $\text{maxa} = a[i]$;
 - Nếu ($a[i] > 0$ và $a[i] < \text{mind}$) thì $\text{mind} = a[i]$;
 - (3.3) Xử lý maxa, mind:
 - Nếu ($\text{maxa} < -10^9$) thì $\text{maxa} = 0$;
 - Nếu ($\text{mind} > 10^9$) thì $\text{mind} = 0$;
- (4) Xuất ra màn hình trên 1 dòng các giá trị maxa và mind.

3) Cài đặt:

Thực hiện từng câu lệnh trực tiếp:

```
>>> n = int(input())
10
>>> a = [0]*n
>>> import random
>>> for i in range(n):
    a[i] = random.randint(-10**9, 10**9)
>>> maxa = -10**9 - 1; mind = 10**9 + 1
>>> for i in range(n):
    if (a[i] < 0 and a[i] > maxa): maxa =
a[i]
    if (a[i] > 0 and a[i] < mind): mind =
a[i]
>>> if (maxa < -10**9): maxa = 0
>>> if (mind > 10**9): mind = 0
>>> print(maxa, ' ', mind)
-291592625 78584917
>>>
```

Viết thành chương trình rồi thực hiện:

```
n = int(input())
a = [0]*n
import random
for i in range(n):
    a[i] = random.randint(-10**9, 10**9)
maxa = -10**9 - 1; mind = 10**9 + 1
for i in range(n):
    if (a[i] < 0 and a[i] > maxa): maxa = a[i]
    if (a[i] > 0 and a[i] < mind): mind = a[i]
    if (maxa < -10**9): maxa = 0
    if (mind > 10**9): mind = 0
print(maxa, ' ', mind)
10
-47306171 248571910
>>>
```

Ví dụ 3.3 (Bài tập 2.12 - Phụ lục):

Nhập vào từ bàn phím số nguyên dương n không vượt quá 30. Xuất ra màn hình bảng số gồm n hàng và n cột sao cho khi đi từ trái sang phải, từ trên xuống dưới, từ phải sang trái và từ dưới lên trên sẽ nhận được dãy các số tự nhiên theo thứ tự tăng từ 1 đến n^2 .

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ bàn phím số nguyên dương n không vượt quá 30;

Output: Xuất ra màn hình bảng số gồm n hàng và n cột sao cho khi đi từ trái sang phải, từ trên xuống dưới, từ phải sang trái và từ dưới lên trên sẽ nhận được dãy các số tự nhiên theo thứ tự tăng từ 1 đến n^2 ;

Nhân xét: Bảng số cần tạo thường gọi là bảng xoáy tròn ốc. Mỗi lần thực hiện thứ p ($0 \leq p \leq n-1$) sẽ điền các số nguyên liên tiếp bắt đầu từ 1 vào hàng p , cột $n-p-1$, hàng $n-p-1$ và cột p .

2) Thuật toán

- (1) Nhập vào từ bàn phím số nguyên dương n ;
- (2) Tạo bảng số dạng danh sách gồm n hàng và n cột các số nguyên 0: $bs = [[]]$;
- (3) Tạo bảng xoáy tròn ốc bs :
 - (3.1) Khởi tạo: $p = 0$; $k = 0$
 - (3.2) Lặp khi $2*p \leq n$:
 - Điền hàng p của bs với các cột j từ p đến $n-p-1$;
 - Điền cột $n-p-1$ của bs với các hàng i từ $p+1$ đến $n-p-1$;
 - Điền hàng $n-p-1$ của bs với các cột j từ $n-p-2$ ngược đến p ;
 - Điền cột p của bs với các hàng i từ $n-p-2$ ngược đến $p+1$;
- (4) Xuất ra màn hình bảng bs gồm n hàng và n cột.

3) Cài đặt:

$n = \text{int}(\text{input}())$	$j = n-p-2$
$bs = []$;	$\text{while } (j \geq p)$:
$\text{for } i \text{ in range}(n)$:	$k = k+1$
$h = [0]*n$; $bs.append(h)$	$\text{if } (k \leq n*n): bs[n-p-1][j] = k$
$p = 0$; $k = 0$	$\text{else: } p = n$; break
$\text{while } (2*p < n)$:	$j = j-1$
$j = p$	$i = n-p-2$
$\text{while } (j < n-p)$:	$\text{while } (i \geq p+1)$:
$k = k+1$;	$k = k+1$
$\text{if } (k \leq n*n): bs[p][j] = k$	$\text{if } (k \leq n*n): bs[i][p] = k$

<i>else: p = n; break</i>	<i>else: p = n; break</i>
<i>j = j+1</i>	<i>i = i-1</i>
<i>i = p+1</i>	<i>p = p+1</i>
<i>while (i < n-p):</i>	
<i>k = k+1</i>	<i>for i in range(n):</i>
<i>if (k <= n*n): bs[i][n-p-1] = k</i>	<i>for j in range(n):</i>
<i>else: p = n; break</i>	<i>print(bs[i][j], end=' ')</i>
<i>i = i+1</i>	<i>print()</i>

Ví dụ 3.4 (Bài tập 3.3 - Phụ lục):

Nhập vào bàn phím chuỗi ký tự s. Xuất ra màn hình số lượng mỗi chữ số có trong s.

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ bàn phím chuỗi ký tự s;

Output: Xuất ra màn hình số lượng mỗi chữ số có trong s;

2) Thuật toán

(1) Nhập vào từ bàn phím chuỗi s;

(2) Tính số lượng mỗi chữ số có mặt trong s:

(2.1) Khởi tạo: $sl = [0]*10$

(2.2) Khai báo chuỗi các chữ số $cs = "0123456789"$

(2.3) Duyệt các chữ số i trong chuỗi cs:

Duyệt các ký tự j trong chuỗi s:

Nếu $(i = j)$ thì $sl[i] += 1$

(4) Xuất ra màn hình trên 1 dòng các giá trị của $sl[]$.

3) Cài đặt:

$s = input()$

$cs = "0123456789"$

$sl = [0]*10$

Kết quả chạy chương trình:

1234567891011121314151617181920


```

for i in cs:                2 12 3 2 2 2 2 2 2
    for j in s:              >>>
        if (i == j):
            k = int(i)
            sl[k] += 1
for i in range(10):
    print(sl[i], end=' ')

```

Ví dụ 3.5 (Bài tập 4.8 - Phụ lục):

Đọc số nguyên dương m và các phần tử của dãy $a[]$ gồm n số nguyên dương được cho trong tệp MOD.INP:

- Dòng đầu chứa hai số nguyên dương m và n không vượt quá 10^5 ;
- Dòng sau chứa n số nguyên dương, mỗi số không vượt quá 10^9 .

Ghi ra tệp MOD.OUT:

- Dòng đầu ghi số d là giá trị lớn nhất trong các số dư của phép chia các phần tử của dãy $a[]$ cho m .
- Dòng sau ghi ra các phần tử của dãy $a[]$ có số dư khi chia cho m là d .

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ tệp MOD.INP:

- Dòng đầu chứa hai số nguyên dương m và n không vượt quá 10^5 ;
- Dòng sau chứa n số nguyên dương, mỗi số không vượt quá 10^9 .

Output: Ghi ra tệp MOD.OUT:

- Dòng đầu ghi số d là giá trị lớn nhất trong các số dư của phép chia các phần tử của dãy $a[]$ cho m ;
- Dòng sau ghi ra các phần tử của dãy $a[]$ có số dư khi chia cho m là d .

2) Thuật toán

- (1) Mở tệp MOD.INP để đọc;
- (2) Đọc vào các giá trị m và n ;

(3) Đọc n số nguyên vào dãy $a[]$;

(4) Đóng tệp MOD.INP;

(5) Tìm giá trị d :

(5.1) Khởi tạo: $d = -1$

(5.2) Duyệt các phần tử của $a[]$:

Nếu $(a[i] \% m > d)$ thì $d = a[i] \% m$

(6) Mở tệp MOD.OUT để ghi;

(7) Ghi ra tệp giá trị d và xuống dòng;

(8) Duyệt các phần tử của $a[]$:

Nếu $(a[i] \% m == d)$ thì ghi ra tệp

(9) Đóng tệp MOD.OUT.

3) Cài đặt:

```

fi = open('MOD.INP', 'r')
s = fi.readline()
m, n = list(map(int, s.split()))
s = fi.readline()
a = list(map(int, s.split()))
fi.close()
d = -1
for i in range(n):
    if (a[i] % m > d): d = a[i] % m
fo = open('MOD.OUT', 'w')
fo.write(str(d) + '\n')
for i in range(n):
    if (a[i] % m == d):
        fo.write(str(a[i]) + ' ')
fo.close()

```

Kết quả chạy chương trình:

Nội dung tệp MOD.INP:

10 5

9 27 29 19 2

Nội dung tệp MOD.OUT:

9

9 29 19

Ví dụ 3.6 (Bài tập 5.3 - Phụ lục):

Viết chương trình con $ucln(a, b)$ tìm ước chung lớn nhất của hai số nguyên dương a , b sử dụng thuật toán Euclid.

Hướng dẫn giải

1) Xây dựng chương trình con

- Tên CTC: $ucln$;
- Các tham số hình thức của CTC: hai số nguyên dương a , b ;
- Giá trị trả về: ước chung lớn nhất của a và b ;
- Thuật toán sử dụng: Thuật toán Euclid;
- Lưu ý vai trò của các biến toàn cục và biến địa phương;

2) Sử dụng CTC

- Thay thế các tham số hình thức bằng các giá trị hằng hoặc biến là các số nguyên dương;
- Lưu ý hình thức cài đặt CTC để sử dụng phù hợp

3) Cài đặt:

Cài đặt CTC dưới dạng hàm

Xây dựng CTC tìm ước chung lớn nhất của hai số nguyên a và b dưới dạng hàm:

```
>>> def ucln(a, b):
```

```
    while (b > 0):
```

```
        r = a % b
```

```
        a = b
```

```
        b = r
```

```
    return(a)
```

```
>>>
```

Hoạt động của CTC:

```
>>> a = 15; b = 80
```

```
>>> print(ucln(a,b))
```

```
5
```

```
>>> print(a)
```

```
15
```

```
>>> print(b)
```

```
80
```

```
>>> print(ucln(10,25))
```

```
5
```

```
>>> print(ucln(ucln(a,b),ucln(10,25)))
```

```
5
```

```
>>>
```

Cài đặt CTC dưới dạng thủ tục

Xây dựng CTC tìm ước chung lớn nhất của hai số nguyên a và b dưới dạng thủ tục sử dụng biến toàn cục d:

```
>>> def ucln(a, b):
```

```
    while (b > 0):
```

```
        r = a % b
```

```
        a = b
```

```
        b = r
```

```
    global d
```

```
    d = a
```

```
>>>
```

Hoạt động của CTC:

```
>>> a = 15; b = 80
```

```
>>> ucln(a,b)
```

```
>>> print(d)
```

```
5
```

```
>>> ucln(12,18)
```

```
>>> print(d)
```

```
6
```

```
>>> print(r)
```

```
Traceback (most recent call last):
```

```
File "<pyshell#10>", line 1, in <module>
```

```
    print(r)
```

```
NameError: name 'r' is not defined
```

```
>>>
```

Ví dụ 3.7 (Bài tập 5.8 - Phụ lục):

Có n tầng tháp khác nhau đặt tại cọc 1. Cần chuyển n tầng tháp sang cọc 2 với điều kiện:

- (1) Mỗi lần chỉ được chuyển 1 tầng tháp;
- (2) Không được đặt tầng tháp lớn trên tầng tháp nhỏ;
- (3) Được sử dụng cọc 3 để đặt các tầng tháp.

Tìm cách chuyển tháp với số lần chuyển tháp ít nhất thỏa mãn bài toán.

Nhập vào từ bàn phím số nguyên dương n ($n \leq 20$). Ghi ra tệp THAP.OUT gồm một số dòng, mỗi dòng là 1 lần chuyển tháp từ cọc a sang cọc b dưới dạng: a 'Move' b.

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ bàn phím số nguyên dương n ($n \leq 20$);

Output: Ghi ra tệp THAP.OUT gồm một số dòng, mỗi dòng là 1 lần chuyển tháp từ cọc a sang cọc b dưới dạng: a 'Move' b.

Nhận xét: Cần xây dựng CTC mô tả quá trình chuyển n tầng tháp từ cọc a sang cọc b được sử dụng cọc trung gian $c = 6-a-b$ theo các bước sau:

- (1) Nếu $n == 0$ thì dừng.
- (2) Nếu $n > 0$ thì
 - (2.1) Chuyển $n - 1$ tầng tháp từ a sang $6-a-b$;
 - (2.2) Chuyển trực tiếp tầng tháp thứ n từ a sang b : a 'Move' b ;
 - (2.3) Chuyển $n - 1$ tầng tháp từ $6-a-b$ sang b ;

2) Thuật toán

- (1) Nhập vào từ bàn phím số nguyên dương $n \leq 30$;
- (2) Mở tệp THAP.OUT để ghi;
- (3) Xây dựng CTC $Hn(n, a, b)$:
 - (3.1) Nếu $n = 0$ thì dừng;
 - (3.2) Nếu $n > 0$ thì:

$Hn(n-1, a, 6-a-b)$
 Ghi ra tệp nội dung a 'Move' b rồi xuống dòng;
 $Hn(n-1, 6-a-b, b)$
- (4) Thực hiện $Hn(n, 1, 2)$;
- (5) Đóng tệp THAP.OUT.

3) Cài đặt:

<pre> n = int(input()) fo = open('THAP.OUT', 'w') def Hn(n, a, b): if (n == 0): return() Hn(n-1, a, 6-a-b) fo.write(str(a) + ' Move ' + str(b) + '\n') Hn(n-1, 6-a-b, b) Hn(n, 1, 2) fo.close() </pre>	<p>Thực hiện chương trình với $n = 3$:</p> <p>Nội dung tệp THAP>OUT:</p> <pre> 1 Move 2 1 Move 3 2 Move 3 1 Move 2 3 Move 1 3 Move 2 1 Move 2 </pre>
--	--

Ví dụ 3.8 (Bài tập 6.25 - Phụ lục):

6.25 Dãy và số (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm n số nguyên dương a_1, a_2, \dots, a_n .

Yêu cầu: Tìm số nguyên dương nhỏ nhất m không có mặt trong dãy (A).

Dữ liệu: Vào từ tệp NN.INP:

- Dòng đầu chứa số nguyên dương n, $n \leq 10^6$;
- Dòng tiếp theo chứa n số nguyên dương không vượt quá 10^9 .

Kết quả : Ghi ra tệp NN.OUT là số m tìm được.

Ví dụ :

NN.INP	NN.OUT	Giải thích
5 4 1 3 2 4	5	Số nguyên dương nhỏ nhất không có mặt trong dãy là 5.

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ tệp NN.INP:

- Dòng đầu chứa số nguyên dương n, $n \leq 10^6$;
- Dòng tiếp theo chứa n số nguyên dương không vượt quá 10^9 .

Output: Ghi ra tệp NN.OUT số nguyên dương m nhỏ nhất không có mặt trong n số.

Nhận xét: Đây là bài toán tìm kiếm dữ liệu. Phạm vi tìm kiếm m: $1 \leq m \leq n+1$.

2) Thuật toán

- (1) Mở tệp NN.INP để đọc;
- (2) Đọc số n và n số vào danh sách dạng dãy a[];
- (3) Đóng tệp NN.INP;
- (4) Tìm m:
 - (4.1) Khởi tạo m = n+1
 - (4.2) Đưa vào dãy vs đánh dấu các số i trong phạm vi từ 1 đến n có trong dãy a[]

(4.3) Duyệt các phần tử của dãy $a[]$:

Nếu $(a[i] \leq n)$ thì $vs[a[i]] = 1$

(4.4) Duyệt các phần tử i của dãy vs , với $1 \leq i \leq n$:

Nếu $(vs[i] = 0)$ thì $m = i$ và dừng

(5) Mở tệp NN.OUT để ghi;

(6) Ghi giá trị m ra tệp;

(7) Đóng tệp NN.OUT.

3) Cài đặt:

```

fi = open('NN.INP', 'r')
n = int(fi.readline())
s = fi.readline()
a = list(map(int, s.split()))
fi.close()

m = n+1
vs = [0]*(n+1)
for i in range(n):
    if (a[i] <= n): vs[a[i]] = 1

for i in range(1, n+1):
    if (vs[i] == 0):
        m = i
        break

fo = open('NN.OUT', 'w')
fo.write(str(m))
fo.close()

```

Ví dụ 3.9 (Bài tập 6.40 - Phụ lục):

6.40 Xâu còn lại (Thời gian chạy chương trình ≤ 1 giây)

Cho một bảng các ô vuông gồm n hàng và n cột. Tại mỗi ô của bảng người ta viết 1 chữ cái in hoa nào đó trong các chữ cái từ A đến Z. Từ bảng đã cho, người ta lần lượt chọn ra m xâu theo quy tắc: mỗi khi chọn ra một xâu cần phải xóa tất cả các chữ cái trong bảng thuộc xâu đó. Các chữ cái còn lại trong bảng được viết thành xâu s có thứ tự từ điển nhỏ nhất.

Yêu cầu: Tìm xâu s .

Dữ liệu: Vào từ tệp XAU.INP:

- Dòng đầu chứa hai số nguyên dương n, m , mỗi số không vượt quá 1000.
- Trong n dòng tiếp theo, mỗi dòng chứa n chữ cái in hoa trong phạm vi từ A đến Z.

- Trong m dòng tiếp theo, mỗi dòng chứa một xâu gồm các chữ cái trong phạm vi từ A đến Z và có độ dài không vượt quá 200. Biết rằng các xâu này có thể chọn ra từ bảng đã cho.

Kết quả : Ghi ra tệp văn bản XAU.OUT:

- Dòng đầu ghi số k là số lượng các chữ cái còn lại trong bảng.
- Trong trường hợp $k > 0$, dòng thứ hai ghi xâu s tìm được.

Ví dụ :

XAU.INP	XAU.OUT	Giải thích
5 3 POLTE RWYMS OAIPT BDARN LEMES OLYMPIAD PROBLEM TEST	6 AENRSW	Khi xóa đi ba xâu đã chọn là OLYMPIAD, PROBLEM, TEST thì còn lại 6 chữ cái tạo thành xâu có thứ tự từ điển nhỏ nhất $s = \text{AENRSW}$.

Hướng dẫn giải

1) Xác định bài toán

Input: Nhập vào từ tệp XAU.INP:

- Dòng đầu chứa hai số nguyên dương n, m, mỗi số không vượt quá 1000;
- Trong n dòng tiếp theo, mỗi dòng chứa n chữ cái in hoa của bảng trong phạm vi từ A đến Z;
- Trong m dòng tiếp theo, mỗi dòng chứa một xâu gồm các chữ cái in hoa xóa khỏi bảng.

Output: Ghi ra tệp NN.OUT:

- Dòng đầu ghi số k là số lượng các chữ cái còn lại trong bảng;

- Với $k > 0$, dòng thứ hai ghi xâu s có số thứ tự từ điển nhỏ nhất gồm k chữ cái còn lại.

Nhận xét: Đây là bài toán tìm kiếm dữ liệu. Cần tìm kiếm số lượng mỗi chữ cái in hoa còn lại trong phạm vi từ A đến Z.

2) Thuật toán

- (1) Mở tệp XAU.INP để đọc;
- (2) Đọc số n và m];
- (3) Đọc vào n xâu và cộng vào số lượng mỗi chữ cái in hoa từ A đến Z có trong xâu;
- (4) Đọc vào m xâu và trừ đi số lượng mỗi chữ cái in hoa từ A đến Z có trong xâu;
- (3) Đóng tệp XAU.INP;
- (4) Tìm k : k là tổng của tất cả số lượng mỗi chữ cái in hoa từ A đến Z;
- (5) Mở tệp XAU.OUT để ghi;
- (6) Ghi giá trị k ra tệp;
- (7) Nếu k thì ghi ra tệp lần lượt các chữ cái in hoa từ A đến Z với số lượng tương ứng;
- (8) Đóng tệp XAU.OUT.

3) Cài đặt:

```
cc = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
```

```
sl = [0]*26
```

```
f1 = open('XAU.INP', 'r')
```

```
s = f1.readline()
```

```
n, m = list(map(int, s.split()))
```

```
for i in range(n):
```

```
    s = f1.readline()
```

```
    for x in cc:
```

```
        for y in s:
```

```
            if (x == y):
```

```
                t = ord(x) - ord('A')
```

```

        sl[t] += 1
for i in range(m):
    s = fi.readline()
    for x in cc:
        for y in s:
            if (x == y):
                t = ord(x) - ord('A')
                sl[t] -= 1
fi.close()
k = 0
for i in range(26):
    k += sl[i]
fo = open('XAU.OUT', 'w')
fo.write(str(k) + '\n')
if (k > 0):
    for i in range(26):
        for j in range(sl[i]):
            fo.write(cc[i])
fo.close()

```

3.3.2 Phương pháp đánh giá bài tập lập trình của học sinh

Phương pháp đánh giá lời giải bài tập lập trình của học sinh đóng vai trò rất quan trọng trong quá trình hướng dẫn học sinh học lập trình trên Python. Mục đích đánh giá nhằm giúp giáo viên nắm được mức độ tiếp thu kiến thức lập trình, các lỗi còn gặp trong chương trình. Từ đó giáo viên có kế hoạch trợ giúp học sinh lập trình tốt hơn. Mặt khác, quá trình đánh giá bài tập lập trình của học sinh còn để cho điểm học sinh theo qui định của Bộ Giáo dục và Đào tạo về môn Tin học.

Hai phương pháp đánh giá thường được sử dụng là phương pháp đánh giá trực tiếp và phương pháp đánh giá tự động.

Phương pháp đánh giá trực tiếp là giáo viên kiểm tra trực tiếp từng câu lệnh trong chương trình của học sinh. Khi thực hiện đánh giá trực tiếp, giáo viên có thể yêu cầu học sinh thực hiện trực tiếp từng câu lệnh. Trên cơ sở đó, giáo viên phân tích cho học sinh thấy rõ tính đúng, tính hiệu quả của từng câu lệnh và toàn bộ chương trình.

Xét ví dụ 3.1. Ngoài lời giải đúng như đã trình bày, học sinh có thể phạm các lỗi sau:

Lỗi 1: Học sinh có thể thực hiện vòng lặp *for* hoặc *while* trong phạm vi từ 1 đến n . Điều này dẫn đến thời gian thực hiện chương trình lớn khi n đủ lớn. Do đó, lời giải của học sinh không đảm bảo tính hiệu quả của chương trình.

Lỗi 2: Học sinh khi cài đặt chương trình có thể quên trường hợp đặc biệt khi n là số chính phương $n = m^2$. Khi đó cặp ước nguyên dương (m, m) chỉ được tính 1 thay vì tính 2 như các cặp ước khác. Do đó, lời giải của học sinh không đảm bảo tính đúng của chương trình.

Phương pháp đánh giá trực tiếp được sử dụng khi đánh giá lời giải bài tập lập trình của học sinh trong các phần 1, 2, 3, 4 và 5 của Bài tập lập trình trên Python.

Phương pháp đánh giá tự động là giáo viên đánh giá chương trình của học sinh bằng phần mềm chấm tự động qua các bộ Test mẫu. Để thực hiện phương pháp đánh giá tự động giáo viên cần tiến hành như sau:

Bước 1: Giáo viên xây dựng các bộ Test mẫu bao quát hết các kịch bản có thể xảy ra tương ứng các trường hợp khác nhau của bài toán. Trong đó lưu ý đến các trường hợp đặc biệt của dữ liệu vào và kích cỡ của chúng. Điều đó đảm bảo đánh giá được tính đúng và tính hiệu quả của chương trình.

Bước 2: Giáo viên sử dụng phần mềm chấm tự động để chấm điểm chương trình của học sinh thông qua bộ Test mẫu đã xây dựng.. Giáo viên thực hiện phân tích kết quả chấm tự động để sửa lỗi nếu có của học sinh.

Phương pháp đánh giá tự động được sử dụng khi đánh giá lời giải bài tập lập trình của học sinh trong các phần 4, 5 và 6 của Bài tập lập trình trên Python. Trong đó các bài tập phần 6 nên đánh giá tự động.

Từ thực tế, luận văn khuyến nghị:

- Khi xây dựng bộ Test mẫu giáo viên nên sử dụng module *random* của Python để tạo ngẫu nhiên các bộ test. Trong bộ Test mẫu, số lượng Test khó nên chiếm khoảng 20%.

- Phần mềm chấm tự động nên lựa chọn phần mềm Themis của hai tác giả Lê Minh Hoàng và Đỗ Đức Đông. Đây là phần mềm miễn phí và thường được sử dụng chấm thi trong các kỳ thi học sinh giỏi Tin học.

Phần mềm Themis có thể tải tại link:

https://drive.google.com/file/d/1hz_QkTicnB7pHoR4M51fL5LMLucMU38l/view

Ví dụ 3.10 (Bài tập 6.23 - Phụ lục):

6.23 Chọn phân số lớn nhất (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm n số hạng là các số nguyên bất kỳ khác 0.

Yêu cầu: Tìm hai số hạng a_i và a_j của dãy (A) sao cho phân số a_i/a_j có giá trị nhỏ nhất.

Dữ liệu: Vào từ tệp PSLN.INP:

- Dòng đầu chứa số nguyên dương n , với $2 \leq n \leq 10^5$.

- Dòng tiếp theo chứa n số nguyên bất kỳ khác 0, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

Kết quả: Ghi ra tệp PSLN.OUT hai số a_i và a_j tìm được trên cùng 1 dòng.

Ví dụ:

PSLN.INP	PSLN.OUT	Giải thích
4 -1 5 -4 10	-4 -1	Hai số hạng $a_3 = -4$ và $a_1 = -1$ tạo thành phân số có giá trị lớn nhất là 4..

Khi giải bài tập này cần chú ý đến các trường hợp sau:

Trường hợp 1: Dãy (A) chỉ gồm các số dương.

Trường hợp 2: Dãy (A) chỉ gồm các số âm.

Trường hợp 3: Dãy (A) có chứa các số dương và âm được chia thành các trường hợp sau đây.

Trường hợp 3.1: Dãy (A) chỉ gồm $n = 2$ số hạng;

Trường hợp 3.2: Dãy (A) với $n > 2$ nhưng chỉ có 1 số âm;

Trường hợp 3.3: Dãy (A) với $n > 2$ nhưng chỉ có 1 số dương;

Trường hợp 3.4: Dãy (A) với $n > 2$ và số lượng số âm và số dương đều lớn hơn 1.

Khi xây dựng bộ Test mẫu để đánh giá tự động cần phải có các Test đại diện cho cả 6 trường hợp trên.

3.4 Kết luận chương 3

Trong chương 3 luận văn đã khảo sát các yêu cầu đối với hệ thống bài tập lập trình cho học sinh phổ thông nhằm hỗ trợ rèn luyện kỹ năng lập trình. Luận văn đã xây dựng hệ thống bài tập lập trình trên Python gồm 100 bài tập được chia thành 6 phần phù hợp với tiến trình dạy và học lập trình.

Luận văn cũng trình bày phương pháp hướng dẫn học sinh giải bài tập lập trình thông qua các ví dụ cụ thể. Luận văn cũng trình bày hai phương pháp đánh giá bài tập lập trình của học sinh là phương pháp đánh giá trực tiếp và phương pháp đánh giá tự động..

Trên cơ sở kinh nghiệm giảng dạy Tin học cho học sinh phổ thông của học viên, bước đầu có thể đánh giá hệ thống bài tập lập trình đã xây dựng là phù hợp và có thể góp phần nâng cao tính hiệu quả của dạy lập trình cho học sinh phổ thông.

KẾT LUẬN

Các kết quả đạt được của luận văn:

Với mục tiêu nghiên cứu xây dựng hệ thống bài tập lập trình trên Python cho học sinh phổ thông, luận văn đã đạt được một số kết quả sau đây:

- Luận văn đã khảo sát thực trạng dạy lập trình cho học sinh phổ thông tại Việt Nam và tổng quan về chương trình môn tin học theo khung chương trình mới ban hành cải cách năm 2018.

- Trên cơ sở khảo sát các yêu cầu về dạy lập trình đối với học sinh phổ thông, luận văn đã phân tích và lựa chọn ngôn ngữ Python để dạy lập trình cho học sinh phổ thông.

- Luận văn đã khảo sát lịch sử hình thành, các đặc điểm và các yếu tố chính của ngôn ngữ lập trình Python

- Luận văn đã phân tích lựa chọn và tiến hành trình bày một số nội dung của ngôn ngữ Python dưới dạng 6 nhóm nội dung phù hợp với học sinh phổ thông.

- Luận văn đã khảo sát các yêu cầu kỹ thuật và giải pháp ứng dụng công nghệ E-Learning nhằm đảm bảo tính hiệu quả trong quá trình dạy lập trình Python cho học sinh phổ thông.

- Luận văn đã nghiên cứu tổng quan về các yêu cầu đối với hệ thống bài tập lập trình trên Python cho học sinh phổ thông.

- Luận văn đã xây dựng hệ thống bài tập lập trình trên Python gồm 100 bài tập, được chia thành 6 phần phù hợp với học sinh phổ thông.

- Luận văn đã nghiên cứu và đề xuất phương pháp hướng dẫn học sinh giải bài tập lập trình Python và hai phương pháp đánh giá chương trình Python của học sinh là phương pháp đánh giá trực tiếp và phương pháp đánh giá tự động.

Tuy nhiên, do hạn chế về mặt thời gian, luận văn chưa triển khai thử nghiệm đầy đủ và toàn diện quá trình sử dụng hệ thống bài tập đã xây dựng trong dạy lập trình Python cho học sinh phổ thông.

Hướng phát triển tiếp theo:

Luận văn sẽ thực hiện thử nghiệm sử dụng hệ thống bài tập lập trình trên Python trong dạy lập trình cho học sinh phổ thông. Trên cơ sở đó, tiếp tục hoàn thiện hệ thống bài tập để có thể trở thành tài liệu tốt hỗ trợ dạy và học lập trình tại các trường phổ thông.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Bộ Giáo dục và đào tạo (2018) – ”*Chương trình giáo dục phổ thông – Môn tin học*” - Ban hành kèm theo Thông tư số 32/2018/TT-BGDĐT, ngày 26 tháng 12 năm 2018 của Bộ trưởng Bộ Giáo dục và Đào tạo.
- [2] Bùi Việt Hà (2017) – ”*Python cơ bản*”.
- [3] Nguyễn Bá Kim, Lê Khắc Thành (2006) – “*Phương pháp dạy học đại cương môn Tin học*”, Nhà xuất bản Đại học sư phạm, Hà Nội, 2006.

Tiếng Anh

- [4] Albert Sweigart (2015) - "*Invent Your Own Computer Games with Python*"
- [5] Allen B. Downey (2015) – “*Think Python*”
- [6] Martin Jones (2013) – “*A Programming Course for Beginners*”
- [7] Zed A. Shaw (2016) – “*Learn Python 3 the Hard Way*”, Addison-Wesley, 2016

Trang WEB

- [8] <https://vi.wikipedia.org/wiki>
- [9] <https://www.coursera.org/learn/python>
- [10] <https://www.python.org>

PHỤ LỤC

BÀI TẬP LẬP TRÌNH TRÊN PYTHON

1. Các bài tập mở đầu

1.1 Nhập vào từ bàn phím số thực dương r là bán kính của hình tròn (O). Xuất ra màn hình các giá trị là chu vi và diện tích của hình tròn (O) với giá trị số $\pi = 3.142$.

1.2 Nhập vào từ bàn phím số thực dương a là cạnh của hình vuông. Xuất ra màn hình giá trị diện tích của phần hình vuông nằm ngoài hình tròn nội tiếp nó với giá trị số $\pi = 3.142$.

1.3 Vận tốc v khi chạm đất của vật rơi từ độ cao h (m) được tính theo công thức $v = \sqrt{2gh}$, trong đó g là gia tốc rơi tự do và $g = 9.8\text{m/s}^2$. Nhập vào từ bàn phím số thực dương h . Xuất ra màn hình giá trị v tìm được.

1.4 Một quyển sách gồm một số trang, mỗi trang gồm k dòng. Cho trước số nguyên dương n là số thứ tự của một dòng trong quyển sách đó. Với các giá trị k và n nhập từ bàn phím. Xuất ra màn hình hai số x và y , trong đó x là số thứ tự của trang chứa dòng n và y số thứ tự của dòng n trong trang x .

1.5 Nhập vào từ bàn phím các số thực a, b, c . Xuất ra màn hình giá trị nghiệm hoặc thông báo về tính chất nghiệm của phương trình $ax^2 + bx + c = 0$.

1.6 Nhập vào từ bàn phím các số thực a, b, c . Xuất ra màn hình giá trị nghiệm hoặc thông báo về tính chất nghiệm của phương trình $ax^4 + bx^2 + c = 0$.

1.7 Dãy số Fibonacci được xác định như sau: $a_1 = a_2 = 1$, $a_n = a_{n-1} + a_{n-2}$ với $n \geq 2$. Nhập vào từ bàn phím số nguyên dương n . Xuất ra màn hình giá trị số hạng thứ n của dãy số.

1.8 Nhập vào từ bàn phím số nguyên dương n . Xuất ra màn hình giá trị của tổng

$$s = 1 + 1/2 + \dots + 1/n.$$

1.9 Xuất ra màn hình số lượng gà và số lượng chó thỏa mãn bài toán: “Vừa gà vừa chó. Bó lại cho tròn. Ba mươi sáu con. Một trăm chân chẵn.”.

1.10 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^9 . Xuất ra màn hình số lượng các ước nguyên dương của n .

1.11 Nhập vào từ bàn phím số nguyên dương p không vượt quá 10^9 . Xuất ra màn hình Yes nếu p là số nguyên tố và No nếu p không phải là số nguyên tố.

1.12 Cho $e(n) = 1 + 1/2! + \dots + 1/n!$. Xuất ra màn hình giá trị $e(n)$ khi $1/n! < 2 \cdot 10^{-6}$.

1.13 Nhập vào từ bàn phím hai số nguyên dương m và n không vượt quá 10^9 . Xuất ra màn hình số d là ước chung lớn nhất của m và n .

1.14 Nhập vào từ bàn phím ba số nguyên dương với a, b và m không vượt quá 10^9 , với $a \leq b$. Xuất ra màn hình số lượng các số nguyên dương trong phạm vi từ a đến b là bội của m .

1.15 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^9 . Xuất ra màn hình số lượng các cặp (x, y) với x, y nguyên dương thỏa mãn $x*y + x + y = n$.

1.16 Nhập vào từ bàn phím ba số nguyên dương a , b và c không vượt quá 10^6 . Xuất ra màn hình giá trị lớn nhất \max của biểu thức $x^2 + y^2$, trong đó x và y là các số nguyên dương thỏa mãn $a \cdot x + b \cdot y = c$. Nếu không có x, y thỏa mãn thì xuất ra màn hình số 0.

1.17 Nhập vào từ bàn phím số nguyên dương n . Xuất ra màn hình giá trị của tổng các chữ số của n .

1.18 Hai số nguyên dương a và b gọi là nguyên tố tương đương nếu chúng có cùng các ước nguyên tố. Nhập vào từ bàn phím hai số nguyên dương a và b không vượt quá 10^9 . Xuất ra màn hình Yes nếu a và b nguyên tố tương đương và No nếu a và b không nguyên tố tương đương.

1.19 Một lớp học có m bạn trai và n bạn gái. Cần xếp các bạn trong lớp thành một hàng sao cho mỗi bạn trai đứng cạnh ít nhất một bạn gái và mỗi bạn gái đứng cạnh ít nhất một bạn trai.

Các giá trị m và n được nhập từ bàn phím. Xuất ra màn hình một dãy kí tự chỉ gồm hai chữ cái T và G mô tả phương án xếp hàng, trong đó T là bạn trai, G là bạn gái.

Nếu không có cách xếp hàng thì ghi số 0.

1.20 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^9 . Gọi a, b, d là các số nguyên dương sao cho $a + b = n$ và d là ước chung lớn nhất của a và b . Xuất ra màn hình giá trị lớn nhất của số d .

2. Các bài tập về danh sách, danh sách kiểu dãy và kiểu bảng

2.1 Khai báo dãy số nguyên. Nhập vào từ bàn phím 10 phần tử của dãy. Xuất ra màn hình các phần tử của dãy theo thứ tự tăng.

2.2 Nhập vào từ bàn phím số nguyên dương n và dãy gồm n số bất kỳ. Xuất ra màn hình:

- Dòng đầu ghi số n ;
- Dòng sau ghi giá trị của n số đã nhập được phân cách bởi 1 dấu cách.

2.3 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^6 . Sử dụng module random, tạo dãy gồm n số nguyên có giá trị tuyệt đối không vượt quá 10. Xuất ra màn hình giá trị s là tổng các số hạng và giá trị t là tích các số hạng của dãy.

2.4 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^6 . Sử dụng module random, tạo dãy gồm n số thực có giá trị tuyệt đối không vượt quá 100. Xuất ra màn hình các giá trị \max và \min trên cùng 1 dòng tương ứng là giá trị lớn nhất và giá trị nhỏ nhất trong các số hạng của dãy.

2.5 Nhập vào từ bàn phím số nguyên dương n không vượt quá 10^6 . Sử dụng module random, tạo dãy gồm n số nguyên có giá trị tuyệt đối không vượt quá 10^9 . Xuất ra màn hình các giá trị \max và \min trên cùng 1 dòng tương ứng là giá trị lớn nhất của các số âm và giá trị nhỏ nhất trong các số dương của dãy. Nếu không tìm được \max hoặc \min thì ghi số 0 thay thế.

2.6 Nhập vào từ bàn phím số nguyên n , với $2 \leq n \leq 10^6$. Sử dụng module random, tạo dãy gồm n số nguyên có giá trị tuyệt đối không vượt quá 10^9 . Xuất ra màn hình trên cùng 1 dòng các giá trị của hai số hạng $a[i]$ và $a[j]$ của dãy có tích lớn nhất.

2.7 Nhập vào từ bàn phím số nguyên n , với $2 \leq n \leq 10^6$. Sử dụng module random, tạo dãy gồm n số nguyên có giá trị tuyệt đối không vượt quá 10^9 . Xuất ra màn hình trên cùng 1 dòng các giá trị của hai số hạng $a[i]$ và $a[j]$ của dãy có tích nhỏ nhất.

2.8 Khai báo bảng số nguyên. Nhập vào từ bàn phím 10 số nguyên để tạo thành bảng gồm 2 hàng và 5 cột. Xuất ra màn hình bảng số vừa tạo với các số trong 1 hàng được in trên 1 dòng và cách nhau 1 dấu cách.

2.9 Khai báo bảng số nguyên. Nhập vào từ bàn phím số hàng m và số cột n . Sử dụng module random tạo bảng các số nguyên gồm m hàng và n cột. Xuất ra màn hình:

- Dòng đầu ghi m số là giá trị lớn nhất của mỗi hàng;
- Dòng sau ghi n số là giá trị nhỏ nhất của mỗi cột.

2.10 Một danh sách gồm n học sinh, mỗi học sinh bao gồm các thông tin: Họ tên, ngày sinh, điểm trung bình các môn học. Nhập các thông tin của n học sinh từ bàn phím. Xuất ra màn hình thông tin của n học sinh, mỗi học sinh trên 1 dòng.

2.11 Dãy các số nguyên dương $a[1], \dots, a[n]$ gọi là dãy số hạnh phúc nếu là dãy số giảm và với mọi $i > 1$ thì $a[i]$ hoặc là số nguyên tố, hoặc là ước của 1 trong các số $a[1], \dots, a[i-1]$. Nhập vào từ bàn phím số nguyên dương x không vượt quá 10^5 . Xuất ra màn hình:

- Dòng đầu ghi số k là số lượng các phần tử của dãy hạnh phúc với $a[1] = x$;
- Dòng sau ghi k số hạng của dãy hạnh phúc tìm được.

2.12 Nhập vào từ bàn phím số nguyên dương n không vượt quá 30. Xuất ra màn hình bảng số gồm n hàng và n cột sao cho khi đi từ trái sang phải, từ trên xuống dưới, từ phải sang trái và từ dưới lên sẽ nhận được dãy các số tự nhiên theo thứ tự tăng từ 1 đến n^2 .

3. Các bài tập về xâu kí tự

3.1. Một xâu kí tự không chứa dấu cách gọi là một từ. Nhập vào từ bàn phím xâu kí tự s . Xuất ra màn hình số lượng các từ có chứa trong s .

3.2 Nhập vào bàn phím xâu kí tự s chỉ chứa các chữ cái tiếng Anh. Xuất ra màn hình số lượng mỗi chữ cái tiếng Anh có trong s (không biệt chữ hoa hoặc chữ thường).

3.3 Nhập vào bàn phím xâu kí tự s . Xuất ra màn hình số lượng mỗi chữ số có trong s .

3.4 Nhập vào từ bàn phím xâu kí tự s . Xuất ra màn hình xâu kí tự có thể nhận được từ s bằng cách thực hiện một số lần hoán vị hai kí tự bất kỳ của s và có thứ tự từ điển nhỏ nhất.

3.5 Nhập vào từ bàn phím chuỗi ký tự s . Xuất ra màn hình chuỗi ký tự có thể nhận được từ s bằng cách thực hiện một số lần hoán vị hai ký tự bất kỳ của s và có thứ tự từ điển lớn nhất.

3.6 Nhập vào từ bàn phím ba chuỗi ký tự s , s_1 và s_2 . Thực hiện thay thế tất cả các chuỗi s_1 có trong s bởi chuỗi s_2 . Xuất ra màn hình số lần thực hiện thay thế và chuỗi s sau khi đã thay thế.

3.7 Nhập vào từ bàn phím chuỗi ký tự s có chứa các ký tự số. Xuất ra màn hình giá trị max là số lớn nhất có mặt trong s .

3.8 Một chuỗi s chỉ gồm các dấu mở ngoặc '(' và đóng ngoặc ')' gọi là hợp lệ nếu có thể viết thêm các số và các phép toán để tạo thành biểu thức toán học đúng. Ví dụ chuỗi $()()$ là hợp lệ, còn chuỗi $()()$ là không hợp lệ. Nhập chuỗi vào từ bàn phím chuỗi s , Xuất ra màn hình Yes nếu chuỗi s hợp lệ hoặc No nếu chuỗi s không hợp lệ.

4. Các bài tập về dữ liệu có cấu trúc kiểu tệp

4.1. Nhập từ bàn phím số nguyên dương n không vượt quá 10^5 . Tạo ngẫu nhiên dãy ds gồm n số nguyên có giá trị tuyệt đối không vượt quá 10^9 . Ghi ra tệp DAY.INP:

- Dòng đầu ghi số n ;
- Dòng sau ghi n giá trị nguyên của dãy ds .

4.2. Nhập từ bàn phím hai số nguyên dương m , n không vượt quá 10^5 . Tạo ngẫu nhiên bảng bs gồm m hàng và n cột các số nguyên có giá trị tuyệt đối không vượt quá 100.

Ghi ra tệp BANG.INP:

- Dòng đầu ghi hai số m và n ;
- Ghi bảng bs gồm m hàng và n cột.

4.3. Đọc các phần tử của dãy ds gồm n số nguyên được cho trong tệp DAY.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 ;
- Dòng sau chứa n số nguyên, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

4.4. Đọc các phần tử của bảng bs gồm m hàng và n cột các số nguyên có giá trị tuyệt đối không vượt quá 100 từ tệp BANG.INP:

- Dòng đầu chứa hai số nguyên dương m và n ;
- Trong m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không vượt quá 100.

4.5 Đọc từ tệp TONG.INP số nguyên dương n . Ghi ra tệp TONG.OUT số x nhỏ nhất có tổng các chữ số bằng n .

4.6 Đọc từ tệp TICH.INP số nguyên dương n . Ghi ra tệp TICH.OUT số x nhỏ nhất có tích các chữ số bằng n . Nếu không tìm được x thì ghi số 0 thay thế.

4.7. Đọc các phần tử của dãy $a[]$ gồm n số nguyên được cho trong tệp ABS.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 ;
- Dòng sau chứa n số nguyên, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

Ghi ra tệp ABS.OUT các giá trị x và y trên cùng 1 dòng, trong đó x và y tương ứng là phân tử có giá trị tuyệt đối nhỏ nhất và phân tử có giá trị tuyệt đối lớn nhất của dãy $a[]$.

4.8. Đọc số nguyên dương m và các phần tử của dãy $a[]$ gồm n số nguyên dương được cho trong tệp MOD.INP:

- Dòng đầu chứa hai số nguyên dương m và n không vượt quá 10^5 ;
- Dòng sau chứa n số nguyên dương, mỗi số không vượt quá 10^9 .

Ghi ra tệp MOD.OUT:

- Dòng đầu ghi số d là giá trị lớn nhất trong các số dư của phép chia các phần tử của dãy $a[]$ cho m .
- Dòng sau ghi ra các phần tử của dãy $a[]$ có số dư khi chia cho m là d .

4.9 Trong bảng gồm m hàng, n cột chứa các số nguyên. Số đứng ở hàng i , cột j ký hiệu là $a[i][j]$. Bốn số $a[i][j]$, $a[i][j + 1]$, $a[i + 1][j + 1]$ và $a[i + 1][j]$ gọi là một chu trình nếu chúng theo thứ tự trên lập thành một dãy tăng hoặc dãy giảm.

Đọc các phần tử của bảng $a[][]$ từ tệp CT.INP:

- Dòng đầu chứa hai số nguyên m và n , với $2 \leq m$, $n \leq 100$;
- Trong m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 .

Ghi ra tệp CT.OUT số lượng các chu trình trong bảng $a[][]$.

4.10 Trong bảng gồm m hàng, n cột chứa các số nguyên. Số đứng ở hàng i và cột j của bảng gọi là phần tử yên ngựa nếu nó vừa là số lớn nhất trong hàng i , vừa là số nhỏ nhất trong cột j , $1 \leq i \leq m$, $1 \leq j \leq n$.

Đọc các phần tử của bảng từ tệp YN.INP:

- Dòng đầu chứa hai số nguyên dương m và n , với $m, n \leq 100$;
- Trong m dòng tiếp theo, mỗi dòng chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^9 .

Ghi ra tệp YN.OUT số lượng các phần tử yên ngựa có trong bảng.

5. Các bài tập về chương trình con

5.1 Viết chương trình con $nto(p)$ trả về giá trị 1 nếu số nguyên p là số nguyên tố và trả về 0 nếu ngược lại.

5.2. Nhập vào từ bàn phím số nguyên dương n ($2 \leq n \leq 10^9$). Xuất ra màn hình giá trị $p - q$, trong đó p là số nguyên tố nhỏ nhất lớn hơn n và q là số nguyên tố lớn nhất không vượt quá n .

5.3 Viết chương trình con $ucln(a, b)$ tìm ước chung lớn nhất của hai số nguyên dương a, b sử dụng thuật toán Euclid.

5.4 Nhập vào từ bàn phím hai số nguyên dương a và b ($a, b \leq 10^9$). Xuất ra màn hình bội chung nhỏ nhất m của hai số a và b .

5.5. Nhập từ bàn phím số nguyên n ($2 \leq n \leq 10^5$) và tạo ngẫu nhiên một dãy gồm n số nguyên bất kỳ, mỗi số có giá trị tuyệt đối không vượt quá 10^9 . Xuất ra màn hình giá trị d là ước chung lớn nhất của n số đã cho.

5.6 Sắp xếp dãy số

- Viết chương trình con $ss(k, n)$ biểu diễn thuật toán sắp xếp tăng dãy k gồm n số nguyên theo kiểu lựa chọn.
- Viết chương trình con $iss(k, n)$ biểu diễn thuật toán sắp xếp tăng dãy k gồm n số nguyên theo kiểu xen vào.
- Viết chương trình con $bs(k, n)$ biểu diễn thuật toán sắp xếp tăng dãy k gồm n số nguyên theo kiểu nổi bọt.

5.7 Nhập vào từ bàn phím số nguyên dương n ($n \leq 10^5$) và tạo ngẫu nhiên một dãy gồm n số nguyên bất kỳ, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

Ghi ra tệp BS.OUT:

- Dòng đầu ghi số n ;
- Dòng thứ hai ghi n số của dãy đã được sắp xếp giảm sử dụng thuật toán sắp xếp theo kiểu nổi bọt.

5.8 Tháp Hà Nội

Có n tầng tháp khác nhau đặt tại cọc 1. Cần chuyển n tầng tháp sang cọc 2 với điều kiện:

- Mỗi lần chỉ được chuyển 1 tầng tháp;
- Không được đặt tầng tháp lớn trên tầng tháp nhỏ;
- Được sử dụng cọc 3 để đặt các tầng tháp.

Tìm cách chuyển tháp với số lần chuyển tháp ít nhất thỏa mãn bài toán.

Nhập vào từ bàn phím số nguyên dương n ($n \leq 20$). Ghi ra tệp THAP.OUT gồm một số dòng, mỗi dòng là 1 lần chuyển tháp từ cọc a sang cọc b dưới dạng: a 'Move' b ..

5.9 Viết chương trình con $qs(k, Left, Right)$ biểu diễn thuật toán sắp xếp tăng dãy k gồm các số nguyên từ vị trí $Left$ đến vị trí $Right$ theo kiểu sắp xếp nhanh..

5.10 Cho đẳng thức $((((A_1 ? A_2) ? A_3) ? A_4) ? A_5) ? A_6 = B$, trong đó A_i ($1 \leq i \leq 6$) và B là các số nguyên dương đã cho được nhập vào từ bàn phím. Ghi ra tệp DT.OUT gồm một số dòng, mỗi dòng là 1 đẳng thức đúng, trong đó các dấu $?$ được thay thế bởi các dấu $+$, $-$, $*$ và chia nguyên $//$. Nếu không tìm được đẳng thức đúng thì ghi số 0.

6. Các bài tập tổng hợp và nâng cao

6.1 Mua hoa (Thời gian chạy chương trình ≤ 1 giây)

Cửa hàng của An có bán hai loại hoa với giá mỗi bông hoa là $a\$$ và $b\$$. Bình có $c\$$ và muốn mua được số lượng bông hoa nhiều nhất.

Yêu cầu: Tìm số tiền nhiều nhất mà An thu được khi bán hoa theo yêu cầu của Bình

Dữ liệu: Vào từ tệp MH.INP gồm 1 dòng chứa ba số nguyên dương a, b, c không vượt quá 10^9 .

Kết quả: Ghi ra tệp MH.OUT số tiền S lớn nhất mà An có thể thu được khi bán hoa cho Bình.

Ví dụ:

MH.INP	MH.OUT	Giải thích
2 3 11	11	An bán cho Bình 4 bông hoa giá 2\$ và 1 bông giá 3\$

6.2 Chi tiết máy (Thời gian chạy chương trình ≤ 1 giây)

Tại phân xưởng sản xuất chi tiết máy có tổng số nguyên liệu là s (đơn vị khối lượng). Quá trình sản xuất chi tiết máy được thực hiện như sau:

- (1) Người ta tạo ra các phôi, mỗi phôi có khối lượng là k (đơn vị khối lượng);
- (2) Từ mỗi phôi người ta tạo ra các chi tiết máy, mỗi chi tiết có khối lượng m (đơn vị khối lượng).

Sau đó các phần nguyên liệu dư khi sản xuất phôi và chi tiết sẽ được tiếp tục sản xuất theo quá trình trên.

Yêu cầu: Tìm số chi tiết máy nhiều nhất có thể sản xuất được.

Dữ liệu: Vào từ tệp CTM.INP chứa ba số nguyên dương s, k và $m \leq 10^9$.

Kết quả: Ghi ra tệp CTM.OUT số chi tiết máy nhiều nhất có thể sản xuất được.

Ví dụ :

CTM.INP	CTM.OUT	Giải thích
13 5 3	3	Sau khi sản xuất được 2 chi tiết, số lượng nguyên liệu còn dư là 7. Do đó có thể sản xuất thêm 1 chi tiết nữa. Tổng cộng có 3 chi tiết được sản xuất.

6.3 Biểu diễn số (Thời gian chạy chương trình ≤ 1 giây)

Cho số nguyên dương n.

Yêu cầu: Tìm số lượng S các cách biểu diễn số n thành tổng của các số nguyên dương liên tiếp.

Dữ liệu: Vào từ tệp BDS.INP chứa số nguyên dương n, trong đó $n \leq 10^9$.

Kết quả: Ghi ra tệp BDS giá trị S tìm được.

Ví dụ:

BDS.INP	BDS.OUT	Giải thích
15	4	Số 15 = 15 = 7 + 8 = 4 + 5 + 6 = 1 + 2 + 3 + 4 + 5. Có 4 cách biểu diễn 15 thành tổng của các số nguyên dương liên tiếp.

6.4 Số giả nguyên tố (Thời gian chạy chương trình ≤ 1 giây)

Số nguyên dương x gọi là số giả nguyên tố nếu x có không quá 3 ước số nguyên dương. Cho số nguyên dương n.

Yêu cầu: Tìm số giả nguyên tố x lớn nhất không vượt quá n.

Dữ liệu: Vào từ tệp GNT.INP chứa số nguyên dương n, trong đó $n \leq 10^9$.

Kết quả: Ghi ra tệp GNT.OUT số x tìm được.

Ví dụ :

GNT.INP	GNT.OUT	Giải thích
---------	---------	------------

26	25	Số 25 có đúng 3 ước số nguyên dương là 1, 5 và 25.
----	----	--

6.5 Chia hết (Thời gian chạy chương trình ≤ 1 giây)

Cho trước ba số nguyên dương a, b và n .

Yêu cầu: Tìm số lượng s các số trong khoảng từ 1 đến n chia hết cho a hoặc b .

Dữ liệu: Vào từ tệp CH.INP chứa ba số nguyên dương a, b và n , mỗi số $\leq 10^9$;

Kết quả: Ghi ra tệp CH.OUT số s tìm được.

Ví dụ:

CH.INP	CH.OUT	Giải thích
2 10	5 2	- Có 5 số không vượt quá 10 là bội của 2. - 2 là số nguyên tố lớn nhất là ước chung của 2 và 10.

6.6. Phân tích số thành tích các số nguyên tố (Thời gian chạy chương trình ≤ 1 giây)

Cho trước số nguyên dương x , $2 \leq x \leq 10^9$.

Yêu cầu: Tìm cách phân tích số x thành tích các số nguyên tố.

Dữ liệu: Vào từ tệp văn bản PT.INP gồm một dòng chứa số nguyên dương x .

Kết quả: Ghi ra tệp văn bản PT.OUT:

- Dòng đầu ghi số k là số lượng các ước nguyên tố khác nhau của x .
- Trong k dòng tiếp theo, mỗi dòng ghi hai số nguyên dương p và m , trong đó p là ước nguyên tố của x , m là số mũ của p trong phân tích của x thành các tích các số nguyên tố.

Ví dụ:

PT.INP	PT.OUT	Giải thích
200	2 2 3 5 2	Có $200 = 2^3 \cdot 5^2$

6.7 Ước nguyên tố (Thời gian chạy chương trình ≤ 2 giây)

Cho trước số nguyên dương n .

Yêu cầu: Tìm số nguyên tố p lớn nhất là ước của n .

Dữ liệu: Vào từ tệp UNT.INP chứa số nguyên dương n , $2 \leq n \leq 10^9$.

Kết quả: Ghi ra tệp UNT.OUT số p tìm được.

Ví dụ:

UNT.INP	UNT.OUT	Giải thích
2015	403	Số 403 là ước nguyên tố lớn nhất của 2015

6.8 Số mũ lớn nhất (Thời gian chạy chương trình ≤ 1 giây)

Cho hai số nguyên dương n và p , trong đó $n \leq 10^9$, $p \leq 10^9$.

Yêu cầu: Tìm số tự nhiên m lớn nhất sao cho p^m là ước của n !

Dữ liệu: Vào từ tệp SM.INP chứa hai số nguyên dương n và p .

Kết quả: Ghi ra tệp SM.OUT số m .

Ví dụ :

SM.INP	SM.OUT	Giải thích
6 10	1	Có $6! = 720$ nên $m = 1$

6.9 Đếm các phân số (Thời gian chạy chương trình ≤ 1 giây)

Cho trước số nguyên dương n không vượt quá 10^4 .

Yêu cầu: Đếm số lượng L các phân số có giá trị khác nhau có tử số và mẫu số là các số nguyên a, b , trong đó $0 \leq a \leq b$, $1 \leq b \leq n$.

Dữ liệu: Vào từ tệp PS.INP chứa số nguyên dương n không vượt quá 10^4 .

Kết quả: Ghi ra tệp PS.OUT số L tìm được.

Ví dụ :

PS.INP	PS.OUT	Giải thích
4	7	Các phân số khác nhau thỏa mãn: $0/1, 1/1, 1/2, 1/3, 2/3, 1/4, 3/4$.

6.10 Số song nguyên tố (Thời gian chạy chương trình ≤ 1 giây)

Với mỗi số tự nhiên x , ký hiệu $S(x)$ là tổng các chữ số của x trong hệ thập phân, Số x gọi là số song nguyên tố nếu x và $S(x)$ đều là số nguyên tố.

Cho trước số nguyên dương n .

Yêu cầu: Tìm số nguyên dương lớn nhất không vượt quá n là số song nguyên tố.

Dữ liệu: Vào từ tệp SNT.INP chứa số nguyên dương n , $2 \leq n \leq 10^9$.

Kết quả: Ghi ra SNT số song nguyên tố tìm được.

Ví dụ:

SNT.INP	SNT.OUT	Giải thích
25	23	Số 23 là số song nguyên tố, còn các số 24, 25 thì không. Do đó số 23 thỏa mãn bài ra.

6.11 Tổng bình phương các chữ số (Thời gian chạy chương trình ≤ 1 giây)

Cho trước số nguyên dương x không vượt quá 10^9 . Ký hiệu $S(x)$ là tổng bình phương các chữ số của x . Ta lập dãy số $\{u_n\}$ như sau: $u_1 = S(x)$; còn với $n > 1$ thì $u_n = S(u_{n-1})$.

Yêu cầu: Xác định giá trị lớn nhất y của các số hạng có trong dãy $\{u_n\}$.

Dữ liệu: Vào từ tệp BP.INP chứa số x .

Kết quả: Ghi ra tệp BP.OUT số y tìm được.

Ví dụ:

BP.INP	BP.OUT	Giải thích
9	145	Các số hạng khác nhau của dãy $\{u_n\}$ là: 81, 65, 61, 37, 58, 89, 145, 42, 20, 4, 16. Do đó $\text{Max} = 145$.

6.12 Bội số dưới dạng lũy thừa (Thời gian chạy chương trình ≤ 1 giây)

Cho trước số nguyên dương a .

Yêu cầu: Tìm số nguyên dương X nhỏ nhất sao cho X^X là một bội số của a .

Dữ liệu: Vào từ tệp văn bản LT.INP chứa số nguyên dương $a \leq 10^9$.

Kết quả: Ghi ra tệp văn bản LT.OUT số X tìm được.

Ví dụ:

LT.INP	LT.OUT	Giải thích
80	10	$X = 10$ là số thỏa mãn vì 10^{10} chia hết cho 80 và mọi số nguyên dương b có b^b chia hết cho 80 thì $b \geq 10$.

6.13 Hành khách trên xe buýt (Thời gian chạy chương trình ≤ 1 giây)

Một xe buýt đi qua n bến đỗ được đánh số thứ tự từ 1 đến n . Tại mỗi bến đỗ biết được số khách xuống xe, số khách còn lại trên xe hoặc số khách lên xe.

Yêu cầu: Tìm số lượt khách đã đi xe buýt và số lượng khách trên xe lớn nhất tại một thời điểm nào đó trong quá trình xe buýt đi từ bến đầu đến bến cuối.

Dữ liệu: Vào từ tệp HK.INP:

- Dòng thứ nhất chứa số nguyên dương n ($2 \leq n \leq 100$).
- Dòng thứ hai chứa số tự nhiên m ($m \leq 100$) là số lượng khách có trên xe khi xuất phát từ bến thứ nhất.
- Trong $n - 2$ dòng tiếp theo, dòng thứ i chứa hai số tự nhiên a_i ($a_i \leq 100$) là số lượng khách xuống xe và b_i ($b_i \leq 100$) là số lượng khách lên xe tại bến thứ i ($2 \leq i \leq n - 1$).
- Dòng cuối cùng chứa số tự nhiên s ($s \leq 100$) là số lượng khách còn lại trên xe khi về tới bến cuối cùng.

Kết quả : Ghi ra file văn bản HK.OUT :

- Dòng thứ nhất ghi số lượt khách đã đi xe buýt.
- Dòng thứ hai ghi số lượng khách lớn nhất trên xe tại một thời điểm nào đó trong quá trình xe buýt đi từ bến đầu đến bến cuối.

Ví dụ:

HK.INP	HK.OUT	Giải thích
5 10 3 1 5 10 0 2 15	23 15	<ul style="list-style-type: none"> - Số lượt khách đã đi xe buýt là tổng số khách có trên xe ở bến thứ nhất và số lượng khách đã lên xe tại các bến từ thứ hai đến bến thứ 4 là $10 + 1 + 10 + 2 = 23$. - Từ bến 1 đến bến 2 có 10 khách trên xe. Từ bến 2 đến bến 3 có 8 khách trên xe. Từ bến 3 đến bến 4 có 13 khách trên xe. Từ bến 4 đến bến 5 có 15 khách trên xe.

6.14 Tổng 0 (Thời gian tối đa chạy chương trình ≤ 1 giây)

Cho trước n số nguyên bất kỳ a_1, \dots, a_n . Dãy các số a_i, a_{i+1}, \dots, a_j ($1 \leq i \leq j \leq n$) được gọi là dãy con của n số đã cho.

Yêu cầu: Tìm số lượng dãy con có tổng các số hạng bằng 0 của n số nguyên đã cho.

Dữ liệu: Vào từ tệp SUM.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 .
- Dòng sau chứa n số nguyên a_1, \dots, a_n , mỗi số có giá trị tuyệt đối không vượt quá 10^4 .

Kết quả: Ghi ra tệp SUM.OUT: Số lượng dãy con có tổng các số hạng bằng 0 của n số nguyên đã cho.

Ví dụ :

SUM.INP	SUM.OUT	Giải thích
4 3 -15 12 0	3	Có ba dãy con có tổng các số hạng bằng 0 là: - Dãy 1 gồm 3 số: 3, -15, 12; - Dãy 2 gồm: 0; - Dãy 3 gồm cả 4 số đã cho.

6.15 Mua xăng (Thời gian chạy chương trình ≤ 1 giây)

Một ông vua có a dola và b euro để mua xăng. Có n công ty nước ngoài bán xăng được đánh số từ 1 đến n . Công ty thứ i , $1 \leq i \leq n$, có giá bán d_i dola / 1 lit xăng và e_i euro/1 lit xăng. Tuy nhiên, theo quy định, tại mỗi công ty chỉ được phép bán xăng cho một khách hàng hoặc theo dola, hoặc theo euro.

Yêu cầu: Tìm số xăng nhiều nhất mà vua có thể mua được.

Dữ liệu vào cho trong tệp văn bản MX.INP:

- Dòng đầu chứa 3 số nguyên n, a, b , $1 \leq n \leq 10^6$, $0 \leq a, b \leq 10^9$.
- Trong n dòng tiếp theo, mỗi dòng thứ i , $1 \leq i \leq n$, chứa hai số nguyên dương d_i và e_i không vượt quá 10^9 .

Kết quả ghi ra tệp văn bản MX.OUT là số xăng lớn nhất mà vua có thể mua được theo quy định trên với hai chữ số phần lẻ.

Ví dụ:

MX.INP	MX.OUT	Giải thích
3 2 5 6 4 3 5 8 7	1.92	Vua mua xăng theo dola tại công ty số 2 được 0.67 lit, mua xăng theo euro tại công ty thứ nhất được 1.25 lit. Tổng cộng, số xăng lớn nhất mà vua mua được là 1.92 lit.

6.16 Hệ nhị phân (Thời gian chạy chương trình ≤ 1 giây)

Mọi số nguyên dương X đều có thể biểu diễn trong hệ nhị phân tương tự như biểu diễn trong hệ thập phân.

Yêu cầu: Cho trước số X . Tìm biểu diễn trong hệ nhị phân của số X và số Y lớn nhất trong hệ thập phân sao cho biểu diễn nhị phân của Y nhận được từ biểu diễn nhị phân của X bằng cách hoán vị vòng quanh các chữ số.

Dữ liệu vào từ tệp văn bản NP.INP chứa số nguyên dương X ($1 \leq X \leq 10^9$).

Kết quả ghi ra tệp văn bản NP.OUT:

- Dòng thứ nhất biểu diễn nhị phân của số X .
- Dòng thứ hai ghi số Y tìm được.

Ví dụ :

NP.INP	NP.OUT	Giải thích
17	10001 24	$17 = 1 \cdot 2^4 + 1$ $24 = 1 \cdot 2^4 + 1 \cdot 2^3$

6.17 Hệ Hexa (Thời gian chạy chương trình ≤ 1 giây)

Trong hệ cơ số 16 (hệ Hecxa) ta sử dụng 16 chữ số: 0, 1, ..., 9 và A = 10, B = 11, C = 12, D = 13, E = 14, F = 15. Mọi số nguyên dương X đều có thể biểu diễn trong hệ Hecxa tương tự như biểu diễn trong hệ nhị phân.

Yêu cầu: Cho trước số X. Tìm số H là biểu diễn trong hệ Hecxa của số X và số Y lớn nhất trong hệ thập phân sao cho biểu diễn Hecxa của Y nhận được từ biểu diễn Hecxa của X bằng cách hoán vị vòng quanh các chữ số.

Dữ liệu vào từ tệp văn bản HEXA.INP chứa số nguyên dương X ($1 \leq X \leq 10^9$).

Kết quả ghi ra tệp văn bản HEXA.OUT:

- Dòng thứ nhất ghi số H tìm được.
- Dòng thứ hai ghi số Y tìm được.

Ví dụ:

HEXA.INP	HEXA.OUT	Giải thích
1610	64A 2660	$1610 = \$64A$ $2660 = \$A64$

6.18 Mua máy (Thời gian chạy chương trình ≤ 1 giây)

Một nhà máy cần gia công n sản phẩm như nhau nên phải mua một số máy như nhau để sản xuất. Mỗi sản phẩm thứ i được đưa tới nhà máy vào thời điểm d_i và gia công hết t_i đơn vị thời gian. Tại mỗi thời điểm, mỗi máy chỉ gia công một sản phẩm cho đến khi hoàn thành mới chuyển sang gia công sản phẩm khác. Mỗi sản phẩm khi đến nhà máy phải được gia công ngay lập tức.

Yêu cầu: Xác định xem nhà máy cần mua số lượng máy ít nhất là bao nhiêu để đảm bảo sản xuất.

Dữ liệu vào từ tệp văn bản MM.INP:

- Dòng đầu chứa số nguyên dương $n \leq 10^3$;
- Trong n dòng tiếp theo, dòng thứ i ($1 \leq i \leq n$) chứa hai số nguyên dương d_i, t_i không vượt quá 10^9 .

Kết quả ghi ra tệp văn bản MM.OUT là số lượng máy cần mua tìm được.

Ví dụ:

MM.INP	MM.OUT	Giải thích
5 13 4 15 1 11 5 12 3 10 3	3	Máy thứ nhất gia công sản phẩm 5 và 1; Máy thứ hai gia công sản phẩm 3; Máy thứ ba gia công sản phẩm 4 và 2.

6.19 Chênh lệch nhỏ nhất (Thời gian chạy chương trình ≤ 1 giây)

Cho trước số nguyên dương n và dãy gồm n số thực bất kỳ a_1, a_2, \dots, a_n .

Yêu cầu: Tìm cách sắp xếp n số đã cho để tổng S các giá trị tuyệt đối của hiệu hai số kề nhau trong dãy $S = \sum |a_{i+1} - a_i|$, $i = 1, 2, \dots, n-1$ là nhỏ nhất.

Dữ liệu: Vào từ tệp văn bản CL.INP :

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 .
- Dòng sau chứa n số thực bất kỳ a_1, a_2, \dots, a_n .

Kết quả: Ghi ra tệp văn bản CL.OUT:

- Dòng đầu là giá trị nhỏ nhất của S với hai chữ số sau dấu phẩy.
- Dòng sau ghi cách sắp xếp tương ứng của n số thực đã cho.

Ví dụ:

CL.INP	CL.OUT	Giải thích
4 5.1 -7 8 4.2	15.00 -7.0 4.2 5.1 8.0	Sắp xếp -7, 4.2, 5.1, 8 có $S_{\min} = 15.00$

6.20 Tham quan (Thời gian chạy chương trình ≤ 1 giây)

Có n lớp học sinh được đánh số từ 1 đến n tổ chức tham quan. Lớp thứ i ($1 \leq i \leq n$) đi tham quan tại địa điểm thứ i cách trường $D[i]$ km. Một công ty vận tải có m xe ca đánh số từ 1 đến m ($m \geq n$) để phục vụ đưa các lớp học sinh đi tham quan. Xe thứ j ($1 \leq j \leq m$) có mức tiêu thụ xăng là $V[j]$ lit/1 km.

Yêu cầu: Chọn n xe để đưa các lớp học sinh đi tham quan sao cho tổng lượng xăng cần dùng là ít nhất.

Dữ liệu: Vào từ tệp TQ.INP:

- Dòng đầu chứa hai số nguyên dương n, m với $n \leq m < 100$;
- Dòng hai chứa n số nguyên dương $D[1], \dots, D[n]$ không vượt quá 200;
- Dòng ba chứa m số nguyên dương $V[1], \dots, V[m]$ không vượt quá 50.

Kết quả: Ghi ra tệp TQ.OUT: Số lượng xăng cần dùng ít nhất.

Ví dụ:

TQ.INP	TQ.OUT	Giải thích
3 4 7 5 9 17 13 15 10	256	Lớp thứ nhất đi xe thứ 2, lớp thứ hai đi xe thứ 3, lớp thứ ba đi xe thứ 4. Tổng lượng xăng là 256.

6.21 Thuê phòng máy (Thời gian chạy chương trình ≤ 1 giây)

Công ty phần mềm XYZ có n nhóm làm việc, nhóm thứ i ($1 \leq i \leq n$) có v_i nhân viên và cần phòng với ít nhất v_i máy tính để làm việc. Trong thành phố có m phòng máy tính cho thuê, phòng thứ j ($1 \leq j \leq m$) có t_j máy tính. Giá thuê phòng máy tính được tính bằng số lượng máy tính trong phòng nhân với giá thuê 1 máy tính là a .

Yêu cầu: Tìm phương án thuê phòng máy tính cho công ty XYZ sao cho tổng tiền phải trả là ít nhất.

Dữ liệu: Vào từ tệp PM.INP:

- Dòng đầu chứa 3 số nguyên dương n, m và a , với $n, m \leq 10^6$, $a \leq 100$.
- Dòng thứ hai chứa n số nguyên dương v_i , mỗi số không vượt quá 10^9 .
- Dòng thứ ba chứa m số nguyên dương t_j , mỗi số không vượt quá 10^9 .

Kết quả: Ghi ra tệp PM.OUT giá trị x tìm được hoặc số -1 nếu không có phương án thuê.

Ví dụ:

TM.INP	TM.OUT	Giải thích
4 6 10 8 16 15 7 20 7 8 8 15 17	470	Nhóm 1 thuê phòng máy 3, nhóm 2 thuê phòng máy 6, nhóm 3 thuê phòng máy 5 và nhóm 4 thuê phòng máy 2.

6.22 Chọn phân số nhỏ nhất (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm n số hạng là các số nguyên bất kỳ khác 0.

Yêu cầu: Tìm hai số hạng a_i và a_j của dãy (A) sao cho phân số a_i/a_j có giá trị nhỏ nhất.

Dữ liệu: Vào từ tệp văn bản PSNN.INP:

- Dòng đầu chứa số nguyên dương n, với $2 \leq n \leq 10^5$.
- Dòng tiếp theo chứa n số nguyên bất kỳ khác 0, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

Kết quả: Ghi ra tệp văn bản PSNN.OUT hai số a_i và a_j tìm được trên cùng 1 dòng.

Ví dụ:

PSNN.INP	PSNN.OUT	Giải thích
4 -1 5 -4 10	10 -1	Hai số hạng $a_4 = 10$ và $a_1 = -1$ tạo thành phân số có giá trị nhỏ nhất là -10.

6.23 Chọn phân số lớn nhất (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm n số hạng là các số nguyên bất kỳ khác 0.

Yêu cầu: Tìm hai số hạng a_i và a_j của dãy (A) sao cho phân số a_i/a_j có giá trị nhỏ nhất.

Dữ liệu: Vào từ tệp PSLN.INP:

- Dòng đầu chứa số nguyên dương n, với $2 \leq n \leq 10^5$.
- Dòng tiếp theo chứa n số nguyên bất kỳ khác 0, mỗi số có giá trị tuyệt đối không vượt quá 10^9 .

Kết quả: Ghi ra tệp PSLN.OUT hai số a_i và a_j tìm được trên cùng 1 dòng.

Ví dụ:

PSLN.INP	PSLN.OUT	Giải thích
4 -1 5 -4 10	-4 -1	Hai số hạng $a_3 = -4$ và $a_1 = -1$ tạo thành phân số có giá trị lớn nhất là 4..

6.24 Dãy con (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy gồm n số nguyên a_1, a_2, \dots, a_n .

Yêu cầu: Tìm cách chia dãy số thành nhiều đoạn nhất có tổng các số trong mỗi đoạn bằng nhau.

Dữ liệu: Vào từ tệp DC.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 ;
- Dòng tiếp theo chứa n số nguyên có giá trị tuyệt đối không vượt quá 10^4 .

Kết quả: Ghi ra tệp DC.OUT số lượng các đoạn con và tổng của mỗi đoạn con.

Ví dụ:

DC.INP	DC.OUT	Giải thích
--------	--------	------------

5 4 1 3 0 4	3 4	Có thể chia thành 3 đoạn con, mỗi đoạn con có tổng là 4 như sau: Đoạn 1: 4; Đoạn 2: 1, 3; Đoạn 3: 0, 4.
----------------	-----	--

6.25 Dãy và số (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm n số nguyên dương a_1, a_2, \dots, a_n .

Yêu cầu: Tìm số nguyên dương nhỏ nhất m không có mặt trong dãy (A).

Dữ liệu: Vào từ tệp NN.INP:

- Dòng đầu chứa số nguyên dương n, $n \leq 10^6$;
- Dòng tiếp theo chứa n số nguyên dương không vượt quá 10^9 .

Kết quả: Ghi ra tệp NN.OUT là số m tìm được.

Ví dụ :

NN.INP	NN.OUT	Giải thích
5 4 1 3 2 4	5	Số nguyên dương nhỏ nhất không có mặt trong dãy là 5.

6.26 Chọn số (Thời gian tối đa chạy chương trình ≤ 1 giây)

Cho n số nguyên dương a_1, a_2, \dots, a_n . Người ta chia n số nguyên này thành k nhóm, mỗi nhóm có m số. Trong mỗi nhóm thứ i, ($1 \leq i \leq k$) ta tính h_i là hiệu của số lớn nhất và số nhỏ nhất trong nhóm đó và gọi h là số lớn nhất trong k số h_i .

Yêu cầu: Xác định giá trị nhỏ nhất của h.

Dữ liệu: Vào từ tệp CS.INP:

- Dòng đầu chứa ba số nguyên dương n, k và m, $1 \leq k.m \leq n \leq 10^5$.
- Trong n dòng tiếp theo, mỗi dòng thứ i ($1 \leq i \leq n$) chứa số nguyên dương a_i có giá trị không vượt quá 10^9 .

Kết quả: Ghi ra tệp CS.OUT số h tìm được.

Ví dụ :

CS.INP	CS.OUT	Giải thích
8 2 3 170 205 225 190 260 130 225 160	30	Có thể chia 8 số đã cho thành hai nhóm: - Nhóm 1 gồm ba số thứ 1, thứ 4 và thứ 8 có $h_1 = 30$; - Nhóm 2 gồm ba số thứ 2, thứ 3 và thứ 7 có $h_2 = 20$; Do đó $h = 30$ là nhỏ nhất có thể.

6.27 Trò chơi xếp sỏi (Thời gian chạy chương trình ≤ 1 giây)

Có n hộp đánh số từ 1 đến n xếp thành một hình tròn theo chiều quay kim đồng hồ. Trong một số hộp có đựng sỏi sao cho tổng số sỏi không vượt quá 10^6 . Mỗi lượt chơi, người chơi chọn một hộp i có sỏi và lấy tất cả sỏi trong hộp rồi bỏ vào mỗi hộp 1 viên sỏi bắt đầu từ hộp đứng ngay sau hộp i theo chiều quay kim đồng hồ cho đến hết. Sau P lượt chơi như vậy, ta biết được số sỏi trong mỗi hộp.

Yêu cầu: Hãy tìm số sỏi ban đầu trong mỗi hộp.

Dữ liệu vào từ tệp TC.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 200.
- Dòng hai chứa n số nguyên dương a_1, a_2, \dots, a_n là số lượng sỏi sau lượt chơi cuối cùng của mỗi hộp.
- Dòng ba chứa số nguyên dương P ($1 \leq P \leq 200$).
- Dòng bốn chứa P số nguyên dương c_1, c_2, \dots, c_P , trong đó số nguyên dương c_i chỉ rõ số hiệu hộp được bỏ viên đá cuối cùng ở lượt chơi thứ i , $1 \leq i \leq P$.

Kết quả ghi ra tệp TC.OUT gồm n số tự nhiên, trong đó số thứ i là số lượng sỏi ban đầu có trong hộp thứ i , $1 \leq i \leq n$.

Ví dụ:

TC.INP	TC.OUT	Giải thích
4 1 2 2 2 1 4	7 0 0 0	Bốc 7 viên sỏi từ hộp 1 và rải sỏi theo qui tắc sẽ có số lượng sỏi tương ứng từng hộp là 1, 2, 2, 2 và viên sỏi cuối cùng được bỏ vào hộp 4. Thực hiện 1 lượt chơi.

6.28 Công số (Thời gian chạy chương trình ≤ 1 giây)

Một máy tính thực hiện phép cộng các số nguyên dương theo nguyên tắc sau đây. Mỗi lần tính tổng của hai số nào đó, cần trả một khoản tiền có giá trị bằng 5% tổng của chúng. Như vậy, để tính tổng của n số nguyên dương cho trước, cần thực hiện $n-1$ phép tính cộng hai số theo một thứ tự nào đó.

Yêu cầu: Xác định số tiền ít nhất phải trả khi tính được tổng của n số nguyên dương đã cho.

Dữ liệu: Vào từ tệp CS.INP:

- Dòng đầu chứa số nguyên dương $n \leq 10^5$.
- Dòng tiếp theo chứa n số nguyên dương, mỗi số $\leq 10^4$.

Kết quả: Ghi ra tệp CS..OUT là số tiền cần trả ít nhất để tính được tổng của n số đã cho với 2 chữ số sau dấu phẩy.

Ví dụ:

CS.INP	CS.OUT	Giải thích
4 10 11 12 13	4.60	Ta cộng theo thứ tự sau: $10 + 11 = 21$; $12 + 13 = 25$ và $21 + 25 = 46$. Tổng tiền phải trả ít nhất là $0.05(21 + 25 + 46) = 4.60$.

6.29 Dự đoán phim (Thời gian chạy chương trình ≤ 1 giây)

Trong liên hoan phim tại thành phố X có n bộ phim tham gia tranh giải phim hay nhất và đạo diễn xuất sắc nhất. Ban tổ chức nhận được kết quả dự đoán của khán giả qua mạng như sau: Phim thứ i ($1 \leq i \leq n$) có tổng điểm dự báo không đạt giải thưởng nào là a_i , đạt giải thưởng phim hay nhất là b_i và đạt giải thưởng đạo diễn xuất sắc nhất là c_i .

Yêu cầu: Chọn ra hai bộ phim khác nhau u và v , trong đó u đạt giải phim hay nhất và v đạt giải đạo diễn xuất sắc nhất, còn các phim khác không đạt giải nào sao cho tổng điểm dự đoán đúng là lớn nhất.

Dữ liệu: Vào từ tệp PHIM.INP:

- Dòng đầu chứa số nguyên dương n , với $2 \leq n \leq 10^5$;
- Trong n dòng tiếp, dòng thứ i ($1 \leq i \leq n$) chứa 3 số nguyên dương a_i , b_i và c_i , mỗi số không vượt quá 10^9 .

Kết quả: Ghi ra tệp PHIM.OUT tổng điểm dự đoán đúng lớn nhất tìm được;

Ví dụ :

PHIM.INP	PHIM.OUT	Giải thích
3 3 6 9 1 5 7 1 3 9	17	Chọn phim 2 đạt giải phim hay nhất và phim 3 đạt giải đạo diễn xuất sắc nhất có tổng điểm dự đoán đúng lớn nhất tìm được là $3 + 5 + 9 = 17$.

6.30 Bộ ba số đặc biệt (Thời gian chạy chương trình ≤ 1 giây)

Cho dãy (A) gồm m số nguyên dương a_1, a_2, \dots, a_m và dãy (B) gồm n số nguyên dương b_1, b_2, \dots, b_n . Từ dãy (A) lấy ra hai số a_k, a_j ($k \neq j, 1 \leq k, j \leq m$); từ dãy (B) lấy ra một số b_i ($1 \leq i \leq n$) để tạo thành bộ ba số thỏa mãn $a_k < b_i < a_j$ hoặc $a_j < b_i < a_k$. Bộ ba số như vậy gọi là bộ ba số đặc biệt. Đối với các số còn lại của hai dãy (A) và (B) lại tiếp tục quá trình trên để chọn các bộ ba số đặc biệt khác.

Yêu cầu: Xác định số lượng các bộ ba số đặc biệt nhiều nhất có thể chọn được từ hai dãy (A) và (B).

Dữ liệu: Vào từ tệp SDB.INP :

- Dòng đầu chứa hai số nguyên dương m và n , $1 \leq m, n \leq 10^5$;
- Dòng tiếp theo chứa m số nguyên dương a_1, a_2, \dots, a_m , $a_i \leq 10^9, 1 \leq i \leq m$;
- Dòng cuối cùng chứa n số nguyên dương b_1, b_2, \dots, b_n , $b_j \leq 10^9, 1 \leq j \leq n$;

Kết quả: Ghi ra tệp SDB.OUT số lượng các bộ ba số đặc biệt chọn được từ hai dãy (A) và (B).

Ví dụ:

SDB.INP	SDB.OUT	Giải thích
5 3 5 1 3 2 2 2 4 3	2	Bộ ba thứ nhất: $a_1 = 5, a_4 = 2, b_2 = 4$; Bộ ba thứ hai: $a_2 = 1, a_3 = 3$ và $b_1 = 2$.

6.31 Máy tính (Thời gian chạy chương trình ≤ 1 giây)

Có n máy tính đánh số từ 1 đến n . Mỗi máy tính thứ i ($1 \leq i \leq n$) cần trang bị 1 đĩa cứng với dung lượng tối thiểu là a_i (GB). Trên thị trường có m loại đĩa cứng đánh số

từ 1 đến m . Mỗi đĩa cứng loại j ($1 \leq j \leq m$) có dung lượng b_j (GB) và giá tiền c_j (ngàn đồng). Máy tính thứ i có thể trang bị đĩa cứng thứ j nếu $b_j \geq a_i$.

Yêu cầu: Tìm số tiền nhỏ nhất để trang bị đĩa cứng cho n máy tính.

Dữ liệu: Vào từ tệp MT.INP:

- Dòng đầu chứa số nguyên dương n không vượt quá 10^5 .
- Dòng tiếp theo chứa n số nguyên dương a_i , $1 \leq i \leq n$, $a_i \leq 1000$.
- Dòng tiếp theo chứa số nguyên dương m không vượt quá 10^5 .
- Trong m dòng tiếp theo, mỗi dòng thứ j ($1 \leq j \leq m$) chứa hai số nguyên dương b_j , c_j , mỗi số không vượt quá 1000.

Kết quả: Ghi ra tệp MT.OUT số tiền nhỏ nhất cần đầu tư.

Ví dụ:

MT.INP	MT.OUT	Giải thích
3 3 1 2 4 1 10 1 7 3 7 2 3	13	Máy tính thứ 1 trang bị đĩa cứng loại 3, máy tính thứ 2 và thứ 3 trang bị đĩa cứng loại 4. Tổng số tiền cần đầu tư là $7 + 3 + 3 = 13$.

6.32 Quảng cáo (Thời gian tối đa chạy chương trình ≤ 1 giây)

Trong một siêu thị liên tục phát quảng cáo về các mặt hàng mới qua loa phóng thanh. Để sắp xếp được một lịch quảng cáo tối ưu, ban lãnh đạo siêu thị đã tiến hành điều tra khách hàng và biết được hàng ngày mỗi khách hàng đến và rời siêu thị vào những thời điểm nào. Siêu thị muốn tiến hành quảng cáo sao cho mỗi khách hàng khi đến siêu thị nghe được không dưới 2 quảng cáo trong thời gian khách hàng đó ở trong siêu thị. Đồng thời hai quảng cáo bất kỳ không được phát đồng thời và số lượng quảng cáo trong một ngày là nhỏ nhất. Các quảng cáo được bắt đầu phát và kết thúc vào thời điểm là các số nguyên. Khi một khách hàng đến siêu thị hoặc rời siêu thị vào đúng thời điểm bắt đầu phát quảng cáo thì có thể xem như khách hàng nghe được quảng cáo đó.

Yêu cầu: Tìm một lịch phát quảng cáo thoả mãn tất cả các yêu cầu trên.

Dữ liệu: Vào từ tệp QC.INP:

- Dòng đầu chứa số nguyên dương $n \leq 3000$ là số lượng khách hàng đến siêu thị trong vòng một ngày.
- Trong n dòng tiếp theo, mỗi dòng chứa cặp số nguyên dương A và B là thời điểm đến và rời khỏi siêu thị của một khách hàng.

Kết quả: Ghi ra tệp QC.OUT số lượng quảng cáo ít nhất cần phát.

Ví dụ:

QC.INP	QC.OUT	Giải thích
5 1 10 10 12	5	Quảng cáo 1 phát vào thời điểm 9. Quảng cáo 2 phát vào thời điểm 10. Quảng cáo 3 phát vào thời điểm 12.

1 10		Quảng cáo 4 phút vào thời điểm 23.
1 10		Quảng cáo 5 phút vào thời điểm 24.
23 24		

6.33 Chép kinh (Thời gian chạy chương trình ≤ 1 giây)

Trong chùa Thiếu lâm, các nhà sư phạm giới sẽ bị phạt đưa vào Tàng Kinh Các để chép kinh. Hiện tại có tất cả n nhà sư phạm giới được đánh số thứ tự từ 1 đến n và phải chép m cuốn kinh. Nhà sư thứ i ($1 \leq i \leq n$) chép một cuốn kinh bất kỳ sẽ hoàn thành sau d_i ngày. Các nhà sư phải làm việc liên tục cho đến khi chép xong m cuốn kinh. Tại Tàng Kinh Các, các nhà sư sẽ nhận các cuốn kinh để chép lần lượt theo số thứ tự của mình.

Yêu cầu: Xác định nhà sư được giao chép cuốn kinh cuối cùng và ngày diễn ra sự kiện đó.

Dữ liệu: Vào từ tệp văn bản CK.INP:

- Dòng đầu chứa hai số nguyên dương $m \leq 10^4$ và $n \leq 10^3$;
- Dòng tiếp theo chứa n số nguyên dương $d_i \leq 15$ ($1 \leq i \leq n$).

Kết quả: Ghi ra tệp văn bản CK.OUT gồm hai số nguyên dương s và k , trong đó s là số thứ tự của nhà sư được giao chép cuốn kinh cuối cùng và k là ngày diễn ra sự kiện đó.

Ví dụ:

CK.INP	CK.OUT	Giải thích
22 5 1 2 3 4 5	4 9	Đến ngày thứ 9, nhà sư thứ nhất chép cuốn kinh thứ 20, nhà sư thứ hai chép cuốn kinh thứ 21, nhà sư thứ tư chép cuốn kinh thứ 22.

6.34 Vùng đất phẳng (Thời gian chạy chương trình ≤ 1 giây)

Một vùng đất có dạng hình chữ nhật được chia thành lưới các ô vuông đơn vị gồm m hàng và n cột. Tại mỗi ô vuông ghi một số tự nhiên không vượt quá 100 là độ cao của ô đất.

Yêu cầu: Tìm vùng đất phẳng gồm một số ô kề cạnh và có cùng độ cao sao cho diện tích của vùng là lớn nhất.

Dữ liệu: Vào từ tệp VDP.INP::

- Dòng đầu chứa hai số nguyên dương m, n không vượt quá 100.
- Trong m dòng tiếp theo, dòng thứ i ($1 \leq i \leq m$) chứa n số tự nhiên mô tả độ cao các ô trong hàng i . Các số cách nhau ít nhất 1 dấu cách.

Kết quả: Ghi ra tệp VDP.OUT gồm một số là diện tích của vùng đất lớn nhất tìm được trên cùng 1 dòng.

Ví dụ:

VDP.INP	VDP.OUT	Giải thích
4 4 0 8 10 0 1 1 1 1	7	Vùng đất lớn nhất gồm 7 ô: (2,1), (2,2), (2,3), (2,4), (3,3), (3,4) và (4,2).

4 7 1 1		
3 5 1 0		

6.35 Tìm hình (Thời gian chạy chương trình ≤ 1 giây)

Cuối có một mảnh giấy thần hình chữ nhật được chia thành m hàng và n cột. Các hàng được đánh số từ dưới lên trên theo thứ tự từ 1 đến m . Các cột được đánh số từ trái qua phải theo thứ tự từ 1 đến n . Các ô của mảnh giấy thần đều được tô cùng một màu 0. Bờm lấy k mảnh giấy hình chữ nhật có kích thước sao cho có thể đặt vào mảnh giấy thần và không có phần thừa ra ngoài. Các mảnh giấy của Bờm được đánh số từ 1 đến k và tô các màu khác nhau tương ứng từ 1 đến k . Mỗi mảnh giấy thứ i ($1 \leq i \leq k$) được đặt sao cho ô tận cùng phía dưới bên trái ở hàng $h_1[i]$, cột $c_1[i]$ và ô tận cùng phía trên bên phải ở hàng $h_2[i]$, cột $c_2[i]$.

Sau khi Bờm đặt xong k mảnh giấy, mảnh giấy thần ban đầu được chia thành các hình, mỗi hình gồm các ô vuông cùng màu có chung cạnh.

Yêu cầu: Tính số hình trên mảnh giấy thần.

Dữ liệu vào từ trong tệp TH.INP:

- Dòng đầu chứa hai số nguyên dương m, n không vượt quá 10^2 .
- Dòng thứ hai chứa số nguyên dương k không vượt quá 10^3 .
- Trong k dòng tiếp theo, mỗi dòng thứ i chứa 4 số nguyên dương $h_1[i], c_1[i], h_2[i], c_2[i]$, trong đó $h_1[i], h_2[i]$ không vượt quá $m, c_1[i], c_2[i]$ không vượt quá n .

Kết quả ghi ra tệp TH.OUT gồm một dòng chứa số hình tính được.

Các số trên cùng một dòng cách nhau ít nhất 1 dấu cách.

Ví dụ:

TH.INP	TH.OUT	Giải thích
5 4 3 1 1 4 3 1 1 2 2 3 3 5 4	6	Sau khi Bờm đặt 3 mảnh giấy, các ô có màu như sau: 0 0 3 3 1 1 3 3 1 1 3 3 2 2 1 0 2 2 1 0 Do đó có tất cả 6 hình.

6.36 Xâu đối xứng (Thời gian chạy chương trình ≤ 1 giây)

Xâu kí tự x được gọi là đối xứng nếu viết các kí tự của x theo thứ tự ngược lại cũng nhận được x . Cho trước xâu kí tự s chỉ gồm các chữ cái thường tiếng Anh, có độ dài không quá 255.

Yêu cầu: Tìm xâu đối xứng x có độ dài lớn nhất và thứ tự từ điển bé nhất nhận được từ s bằng cách thay đổi vị trí các kí tự và có thể loại bỏ bớt một số kí tự.

Dữ liệu: Vào từ tệp văn bản XDX.INP chứa xâu s .

Kết quả: Ghi ra tệp văn bản XDX.OUT là xâu x tìm được.

Ví dụ:

XDX.INP	XDX.OUT	Giải thích
---------	---------	------------

acbacca	acaca	Xâu <i>acaca</i> là xâu đối xứng có độ dài lớn nhất và thứ tự từ điển bé nhất nhận được từ xâu <i>acbac</i> bằng cách thay đổi vị trí các kí tự và bỏ bớt 1 kí tự b và 1 kí tự c.
---------	-------	---

6.37 Hoán vị xâu (Thời gian chạy chương trình ≤ 1 giây)

Cho trước xâu kí tự *s* chỉ bao gồm các chữ cái tiếng Anh in hoa. Khi thực hiện phép hoán vị vòng quanh các kí tự của *s* sẽ nhận được *m* xâu khác nhau s_1, s_2, \dots, s_m .

Yêu cầu: Tìm xâu có thứ tự từ điển nhỏ nhất và lớn nhất trong *m* xâu trên.

Dữ liệu: Vào từ tệp văn bản HV.INP: Xâu kí tự *s* có độ dài không quá 10^5 .

Kết quả: Ghi ra tệp văn bản HV.OUT:

- Dòng đầu ghi số *m* tìm được;
- Dòng thứ hai ghi xâu có thứ tự từ điển nhỏ nhất tìm được;
- Dòng thứ ba ghi xâu có thứ tự từ điển lớn nhất tìm được.

Ví dụ:

HV.INP	HV.OUT	Giải thích
ABC	3 ABC CAB	Từ xâu 'ABC' khi hoán vị vòng quanh các kí tự nhận được 3 xâu khác nhau: 'ABC', 'BCA', 'CAB', trong đó xâu 'ABC' có thứ tự từ điển nhỏ nhất và xâu 'CAB' có thứ tự từ điển lớn nhất.

6.38 Tách từ (Thời gian chạy chương trình ≤ 1 giây)

Cho trước xâu *S* gồm các chữ cái và dấu cách có độ dài ≤ 255 . Ta gọi một từ là một xâu gồm một số chữ cái liền nhau không chứa dấu cách và không phân biệt chữ in hoa hay thường.

Yêu cầu: Xét xem xâu *S* có chứa bao nhiêu từ khác nhau.

Dữ liệu: Vào từ tệp TU.INP chứa xâu *S*.

Kết quả: Ghi ra tệp TU.OUT:

- Dòng đầu ghi số lượng *t* các từ khác nhau trong *S*.
- Dòng thứ *i* ($1 \leq i \leq t$) trong *t* dòng tiếp theo, mỗi dòng ghi một từ in hoa và số lần xuất hiện của từ đó trong xâu *S* theo thứ tự giảm dần của số lần xuất hiện và thứ tự từ điển của các từ.

Ví dụ:

TU.INP	TU.OUT	Giải thích
Ha Ta ma an ha ta ma an an	4 AN 3 HA 2 MA 2 TA 2	Xâu đã cho chứa 4 từ khác nhau: - Từ AN xuất hiện 3 lần; - Từ HA xuất hiện 2 lần; - Từ MA xuất hiện 2 lần; - Từ TA xuất hiện 2 lần;

6.39 Biến đổi xâu (Thời gian chạy chương trình ≤ 1 giây)

Cho trước xâu kí tự *s*. Mỗi phép biến đổi xâu *s* được thực hiện như sau:

- Chọn hai số nguyên dương *c* và *d* không vượt quá *n*;

- Đổi chỗ hai kí tự trong s đứng ở vị trí c và vị trí d cho nhau.

Sau khi thực hiện m phép biến đổi như trên, nhận được k xâu khác nhau ($k \leq m$) và sắp xếp chúng theo thứ tự từ điển.

Yêu cầu: Tìm xâu x đứng ở vị trí thứ k.

Dữ liệu: Vào từ tệp BDX.INP:

- Dòng đầu chứa số nguyên n, $2 \leq n \leq 255$;

- Dòng thứ hai chứa xâu s;

- Dòng thứ ba chứa số nguyên dương m không vượt quá 10^3 .

- Mỗi dòng trong m dòng tiếp, chứa hai số nguyên dương c và d không vượt quá n.

Kết quả: Ghi ra tệp BDX.OUT:

- Dòng đầu ghi số k;

- Dòng sau ghi xâu x tìm được.

Ví dụ :

BDX.INP	BDX.OUT	Giải thích
5 ABCDE 3 2 4 1 5 3 3	3 EBCDA	Có ba xâu khác nhau được sắp theo thứ tự từ điển là ABCDE, ADCBE, EBCDA. Do đó xâu thứ ba là EBCDA.

6.40 Xâu còn lại (Thời gian chạy chương trình ≤ 1 giây)

Cho một bảng các ô vuông gồm n hàng và n cột. Tại mỗi ô của bảng người ta viết 1 chữ cái in hoa nào đó trong các chữ cái từ A đến Z. Từ bảng đã cho, người ta lần lượt chọn ra m xâu theo quy tắc: mỗi khi chọn ra một xâu cần phải xóa tất cả các chữ cái trong bảng thuộc xâu đó. Các chữ cái còn lại trong bảng được viết thành xâu s có thứ tự từ điển nhỏ nhất.

Yêu cầu: Tìm xâu s.

Dữ liệu: Vào từ tệp XAU.INP:

- Dòng đầu chứa hai số nguyên dương n, m, mỗi số không vượt quá 1000.

- Trong n dòng tiếp theo, mỗi dòng chứa n chữ cái in hoa trong phạm vi từ A đến Z.

- Trong m dòng tiếp theo, mỗi dòng chứa một xâu gồm các chữ cái trong phạm vi từ A đến Z và có độ dài không vượt quá 200. Biết rằng các xâu này có thể chọn ra từ bảng đã cho.

Kết quả : Ghi ra tệp văn bản XAU.OUT:

- Dòng đầu ghi số k là số lượng các chữ cái còn lại trong bảng.

- Trong trường hợp $k > 0$, dòng thứ hai ghi xâu s tìm được.

Ví dụ :

XAU.INP	XAU.OUT	Giải thích
5 3 POLTE RWYMS OAIPT	6 AENRSW	Khi xóa đi ba xâu đã chọn là OLYMPIAD, PROBLEM, TEST thì còn lại 6 chữ cái tạo thành xâu có thứ tự từ điển nhỏ nhất s = AENRSW.

BDARN LEMES OLYMPIAD PROBLEM TEST		
---	--	--