

**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**

**TÓM TẮT LUẬN VĂN THẠC SĨ**

**HÀ NỘI - 2022**

Luận văn được hoàn thành tại:

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

Người hướng dẫn khoa học: **TS. Vũ Văn Thỏa**

Phản biện 1: **PGS.TS. Phạm Thanh Giang**

Phản biện 2: **PGS.TS. Phạm Văn Cường**

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: 10 giờ 45 ngày 02 tháng 07 năm 2022

Có thể tìm hiểu luận văn tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông

## 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 thế 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 ở 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**

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

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.

#### **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.

### **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*

Nội dung môn Tin học phát triển ba mạch kiến thức hoà quyền: 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
- Giai đoạn giáo dục định hướng nghề nghiệp

### *1.2.2 Quan điểm xây dựng chương trình*

#### **Tính kế thừa và phát triển**

- a) Kế thừa chương trình môn Tin học hiện hành
- b) Khai thác chương trình môn Tin học phổ thông của các nước tiên tiến

#### **Tính khoa học, hiện đại và sư phạm**

#### **Tính thiết thực**

- a) Phục vụ định hướng nghề nghiệp
- b) Thực hiện giáo dục STEM

#### **Tính mở**

- a) Nội dung chương trình mở
- b) Hình thức giáo dục đa dạng

### *1.2.3 Mục tiêu của chương trình*

#### **Mục tiêu chung**

#### **Mục tiêu cấp trung học phổ thông**

### *1.2.4 Yêu cầu cần đạt*

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

#### **Yêu cầu cần đạt về năng lực đặc thù**

- 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.**

#### **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.

### **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.

### **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**

### **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.

### **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.

## **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.

## 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. 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.

#### **2.1.2 Đặc điểm của ngôn ngữ lập trình Python**

- 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.
- Python là một ngôn ngữ lập trình đa mẫu hình
- Python sử dụng kiểu động và một dạng kết hợp giữa đếm tham chiếu và bộ dọn rác kiểm tra theo chu kì để quản lí bộ nhớ.
- Python được thiết kế để dễ dàng mở rộng.
- Python có thư viện lớn để hỗ trợ NLT, trong đó có thể dễ dàng thực hiện các chức năng lập trình.

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

#### **a. 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 lí 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.

#### **b. Các kiểu dữ liệu đơn giản chuẩn**

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

#### **c. Câu lệnh và khối lệnh**

Trong Python có các lệnh đơn và các lệnh có cấu trúc.

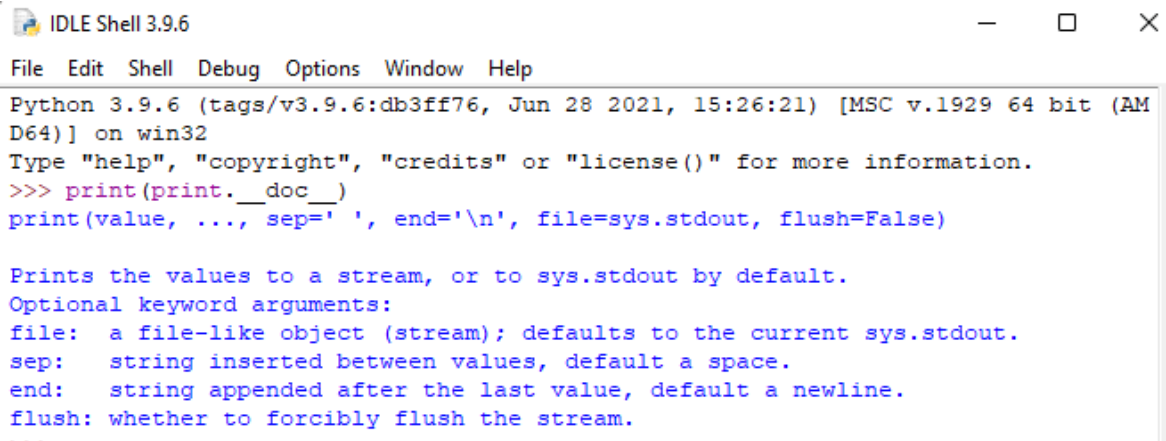


Các câu lệnh đơn bao gồm phép gán (=), nhập dữ liệu từ bàn phím *input()* và xuất dữ liệu ra màn hình *print()*.

Thực hiện ép kiểu dữ liệu bằng câu lệnh gán:

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

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()***

#### d. Các kiểu dữ liệu có cấu trúc

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.

#### e. Các hàm

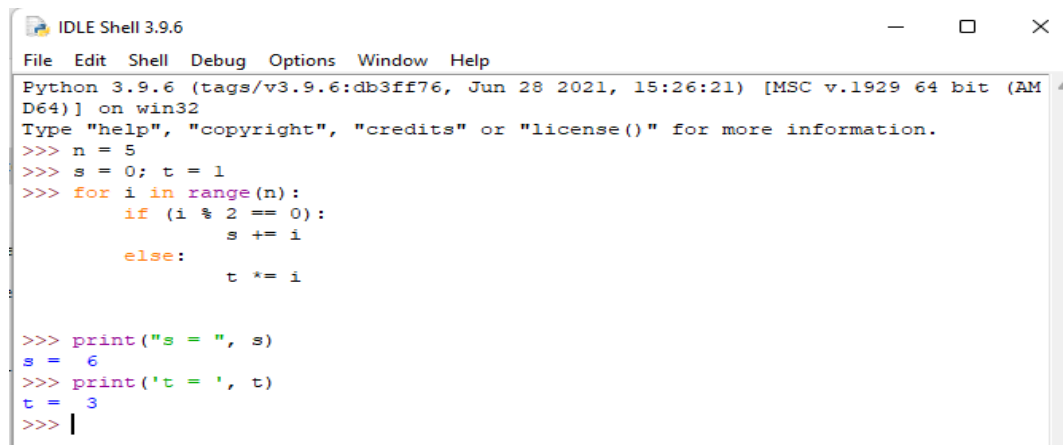
Trong Python không mô tả nội dung về chương trình con (CTC) riêng mà chỉ đề cập đến các hàm mà NLT tự xây dựng.

### 2.3 Lựa chọn và trình bày một số nội dung của Python phù hợp với 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, từ khóa, câu lệnh, khối lệnh



```

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

+ **Đặt tên trong Python**

+ **Hằng và biến**

+ **Chú thích**

+ **Các kiểu dữ liệu đơn giản chuẩn**

- **Kiểu số nguyên:** Tên kiểu: *int*.

- **Kiểu số thực:** Tên kiểu: *float*.

- **Kiểu logic:** Tên kiểu: *bool*.

+ **Câu lệnh gán: Dạng lệnh:** <Danh sách n tên biến> = <Danh sách n biểu thức>

+ **Nhập dữ liệu vào từ bàn phím**

**Dạng lệnh:** <tên biến> = input() (1)

<tên biến>= <tên kiểu>(input()) (2)

+ **Xuất dữ liệu ra màn hình**

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

+ **Các câu lệnh với cấu trúc rẽ nhánh**

+ **Câu lệnh if**

- **Dạng lệnh:** *if* <Điều kiện>:

<Khối lệnh của if>

- **Câu lệnh if else**

**Dạng lệnh:** *if* <Điều kiện>:

<Khối lệnh của if>

*else:*

<Khối lệnh của else>

- **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>

+ **Các câu lệnh với cấu trúc lặp**

**Câu lệnh while**

**Dạng lệnh:** *while* <Điều kiện>:

<Khối lệnh của while>

**Câu lệnh while với else**

**Dạng lệnh:** *while* <Điều kiện>:

<Khối lệnh của while>

*else:*

*<Khởi lệnh của else>*

### **Câu lệnh for**

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

+ **Hàm range()**

**Dạng lệnh:** *range(stop) # 1*

*range(start, stop) # 2*

*range(start, stop, step) # 3*

+ **Câu lệnh for sử dụng lệnh range()**

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

*<Khởi lệnh của for>*

### **Câu lệnh for với else**

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

*<Khởi lệnh của for>*

*else:*

*<Khởi 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:** Dừng để làm dấu nhắc (null), không làm gì.

### **2.3.2 Nhóm nội dung 2**

#### **Khái niệm**

#### **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>]*

#### **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>

### **Tính số phần tử hiện có của danh sách**

**Dạng lệnh:** len(<ds>)

### **Truy cập các phần tử của danh sách**

#### **Truy cập 1 phần tử**

**Dạng lệnh:** <ds>[<vị trí>]

#### **Truy cập nhiều phần tử (lát cắt – slice)**

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

**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>

### **Xóa các phần tử của danh sách**

**Dạng lệnh:** del <phần tử>

**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>

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

**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 bảng (mảng 2 chiều)**

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: <Tên bảng>[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

### **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 hàng  $i$
  - (4,2) Bổ sung hàng  $i$  vào bảng sử dụng phương thức:
 
$$\langle \text{Tên bảng} \rangle.\text{append}(\text{hang}i)$$

#### **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:  $x = \text{random.randint}(gt1, gt2)$ .
- (5) Sử dụng hàm `uniform(gt1, gt2)` để tạo từng phần tử của dãy gồm  $n$  số thực:  $x = \text{random.uniform}(gt1, gt2)$ .

### **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.

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

#### **Khai báo chuỗi ký tự**

Khai báo chuỗi ký tự rỗng:  $\langle \text{Tên biến} \rangle = ""$

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

#### **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:** `print(<Tên biến>)`

**Công dụng:** Xuất xâu kí tự <Tên biến> ra màn hình

**Lệnh gán xâu trên một dòng:** <Tên biến> = "<Các kí tự>"

**Lệnh gán xâu trên nhiều dòng:** <Tên biến> = ""<Các dòng kí tự>""

**Các phép toán trên xâu kí tự**

**Phép ghép xâu**

**Dạng lệnh:**  $str = str1 + str2$

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

**Hàm xác định chiều dài xâu**

**Dạng hàm:**  $len(<Xâu>)$

**Công dụng:** Cho giá trị là số lượng kí tự của <Xâu>.

**Phép so sánh xâu kí tự**

**Một số hàm xử lí xâu kí tự**

**Hàm in:**  $sc\ in\ sm$

Nếu xâu sc chứa trong xâu 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.

### **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.

#### **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>`.

#### **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>`.

### **Đọc dữ liệu từ tệp**

#### **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

#### **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.

#### **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)

### **Sử dụng lệnh with để đọc từ tệp**

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

*s = <Tên biến tệp>.readline()*

### **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**

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.

#### **Xây dựng chương trình con**

#### **Sử dụng chương trình con**

#### **Biến toàn cục và biến địa phương**

#### **Chương trình con đệ quy**

### **2.3.6 Nhóm nội dung 6**

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.

#### **Các thuật toán sắp xếp**

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)

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

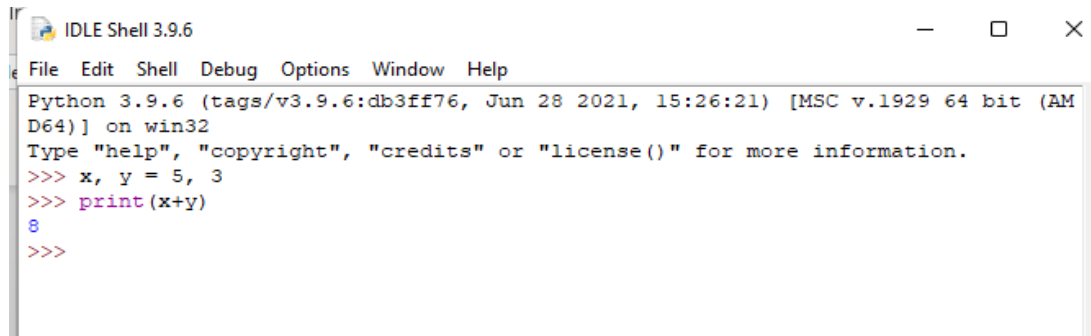
### **2.4.1 Yêu cầu về phần mềm cài đặt Python**

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



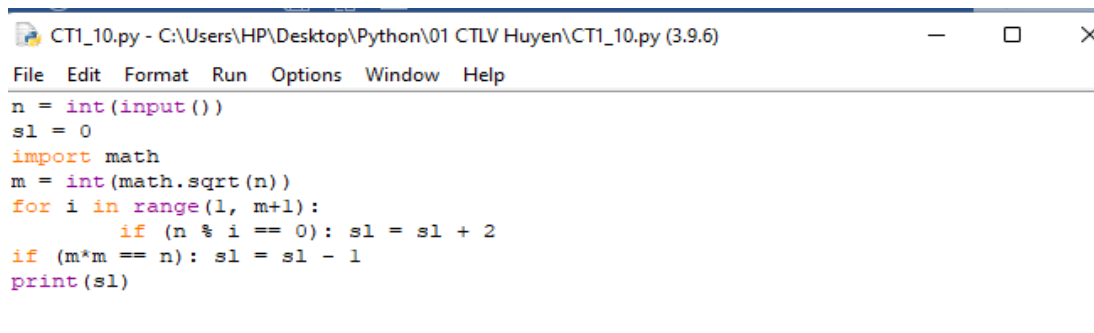
nhau. Luận văn đề xuất sử dụng IDLE Python (Python's Integrated Development and Learning Environment) để dạy lập trình cho học sinh phổ thông.

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.**

#### 2.4.2 Yêu cầu về phòng máy tính

- (1) Về số lượng máy tính:
- (2) Cấu hình máy tính:
- (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

### 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.

## CHƯƠNG 3 - 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**

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

##### **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.

#### **3.2 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. 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

##### **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 xâu kí tự như đếm kí 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.

#### **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.

### **3.3 Hướng dẫn sử dụng hệ thống bài tập**

#### **3.3.1 Phương pháp hướng dẫn học sinh giải bài tập**

**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.

### **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  $n$ .

#### **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 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.

#### **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  $n^2$ .

#### **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  $s$ .

#### **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  $d$ .

**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.

**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'  $b$ .

**Ví dụ 3.8** (Bài tập 6.25 - Phụ lục):

**6.25 Dãy và số** (Thời gian chạy chương trình  $\leq 1$  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.

**Ví dụ 3.9** (Bài tập 6.40 - Phụ lục):

**6.40 Xâu còn lại** (Thời gian chạy chương trình  $\leq 1$  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.

### 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](https://drive.google.com/file/d/1hz_QkTicnB7pHoR4M51fL5LMLucMU38l/view)

### 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.

## KẾT LUẬN

### Các kết quả đạt được của luận văn:

- 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 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>