

CHỦ ĐỀ 5: GIẢI QUYẾT VẤN ĐỀ VỚI SỰ TRỢ GIÚP CỦA MÁY TÍNH

BÀI 16

NGÔN NGỮ LẬP TRÌNH BẬC CAO PYTHON (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết khái niệm ngôn ngữ lập trình bậc cao và ngôn ngữ lập trình bậc cao Python.
- Phân biệt được chế độ gõ lệnh trực tiếp và chế độ soạn thảo chương trình trong môi trường lập trình Python
- Biết cách tạo và thực hiện một chương trình Python.

I. NGÔN NGỮ LẬP TRÌNH BẬC CAO

1. Khái niệm:

- Lập trình: Là việc sử dụng cấu trúc dữ liệu và các câu lệnh của ngôn ngữ lập trình cụ thể để mô tả dữ liệu và diễn đạt các thao tác của thuật toán.
- Ngôn ngữ lập trình bậc cao: có các câu lệnh được viết gần với ngôn ngữ tự nhiên giúp cho việc đọc, hiểu chương trình dễ dàng hơn.
- Một số ngôn ngữ lập trình cao là: Java, C/C++, Python,...

2. Giới thiệu Python:

- Python là một ngôn ngữ lập trình bậc cao phổ biến.
- Python có mã nguồn mở.
- Vì Python có các thư viện chương trình phong phú nên được dùng phổ biến trong nghiên cứu và giáo dục.

II. MÔI TRƯỜNG LẬP TRÌNH PYTHON

Môi trường lập trình Python có hai chế độ:

- Chế độ gõ lệnh trực tiếp thường được dùng để tính toán và kiểm tra nhanh các lệnh.
- Chế độ soạn thảo dùng để viết các chương trình có nhiều dòng lệnh.

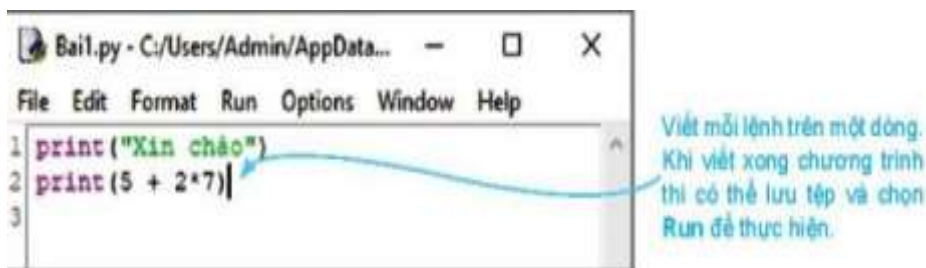
1. Chế độ gõ lệnh trực tiếp:

Gõ lệnh trực tiếp sau dấu nhắc >>> và nhấn phím Enter để thực hiện lệnh như sau:

```
>>> <lệnh python>
```

2. Chế độ soạn thảo:

- Mở màn hình soạn thảo bằng cách vào File/NewFile



Hình 16.3. Màn hình soạn thảo trong môi trường Python

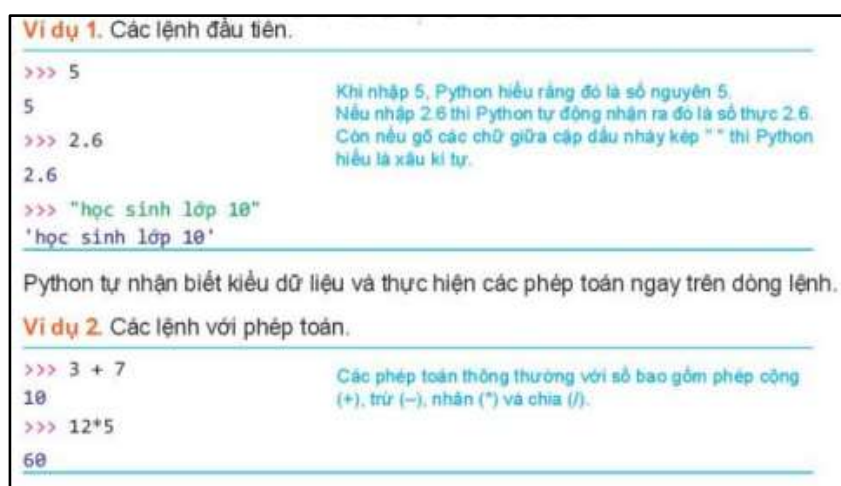
- Chú ý: Có thể soạn thảo chương trình Python bằng phần mềm soạn thảo văn bản hoặc phần mềm lập trình python như Wingware, Pycharm, Thonny, VisualStudio,....

3. Một số lệnh Python đầu tiên:

- Trong Python, lệnh print() có chức năng đưa dữ liệu ra (xuất dữ liệu).
- Cú pháp lệnh print() như sau:

print(v1, v2, ..., vn)

trong đó v1, v2, ..., vn là các giá trị cần đưa ra màn hình.



Ví dụ 3. Lệnh `print()`.

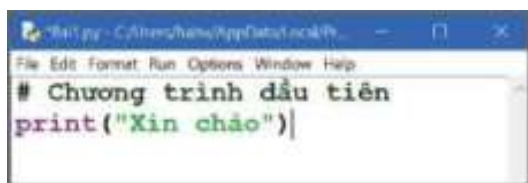
```
>>> print(12)
12
>>> print(10, 3.4 + 4.1, "hoà bình")
10 7.5 hoà bình
>>> print("Đây ba số chẵn:", 2, 4, 6)
Đây ba số chẵn: 2 4 6
>>> print("3 + 7 =", 3+7)
3 + 7 = 10
```

Chú ý: nếu lệnh `print()` nhiều giá trị thì các dữ liệu này sẽ được đưa ra trên một dòng, giữa các dữ liệu sẽ có dấu cách.

Lệnh `print()` có thể tính toán và đưa ra kết quả của biểu thức.

III. THỰC HÀNH

Nhiệm vụ 1: Sử dụng chế độ soạn thảo chương trình của Python để tạo, nhập và chạy chương trình đầu tiên có tên `Bai1.py` như sau:



Hình 16.4

Nhiệm vụ 2: Viết chương trình in ra màn hình:

“Không có việc gì khó

Chỉ sợ lòng không bền

Đào núi và lấp biển

Quyết chí ắt làm nên”

.....

.....

.....

.....

.....

BÀI 17

BIẾN VÀ LỆNH GÁN (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết cách thiết lập biến. Phân biệt được biến và từ khóa.
- Biết sử dụng lệnh gán và thực hiện một số phép toán trên kiểu số nguyên, số thực và xâu kí tự.

I. BIẾN VÀ LỆNH GÁN

1. Khái niệm biến:

- Biến là tên (định danh) của một vùng nhớ dùng để lưu trữ giá trị (dữ liệu) và giá trị đó có thể được thay đổi khi thực hiện chương trình.
- Biến trong Python được tạo ra khi thực hiện lệnh gán.

2. Quy tắc đặt tên biến:

- Chỉ gồm các chữ cái tiếng Anh, các chữ số từ 0 đến 9 và kí tự gạch dưới “_”.
- Không bắt đầu bằng chữ số.
- Phân biệt chữ hoa và chữ thường.

Ví dụ:

.....
.....

3. Lệnh gán:

- Cú pháp của lệnh gán:

<biến> = <giá trị/biểu thức>

Ví dụ:

$x=5$

$y=x+6$

- Khi thực hiện lệnh gán, <giá trị/biểu thức> bên phải sẽ được gán cho <biến>. Nếu biến chưa được khai báo thì nó sẽ được khởi tạo khi thực hiện câu lệnh gán.

- Biến trong Python được xác định kiểu dữ liệu tại thời điểm gán giá trị nên không cần khai báo trước kiểu dữ liệu cho biến.
- Có thể thực hiện tất cả các phép toán thông thường như: +, -, *, /, ... trên các biến có cùng kiểu dữ liệu.

Ví dụ:

- a="Học Python" # biến a kiểu xâu (str)
- a=5 # biến a đổi sang kiểu số nguyên (int)
- a=5.5 # biến a đổi sang kiểu số thực (float)
- a=True # biến a đổi sang kiểu logic (bool)

Chú ý 1: Ta có thể ép kiểu cho biến:

Ví dụ:

```
a = str(5)            # a là một biến kiểu xâu
b = int(5.5)        # b là một biến kiểu nguyên
c = float(5)        # c là một biến kiểu thực
```

Nếu thực hiện lệnh print(a+b) thì Python sẽ báo lỗi vì không thể cộng một xâu với một số nguyên.

Chú ý 2: Có thể gán đồng thời các giá trị cho nhiều biến:

<var1>, <var2>, ..., <varn> = <gt1>, <gt2>, ..., <gtn>

(số biến bên trái phải đúng bằng số giá trị bên phải)

Ví dụ:

.....

Chú ý 3: Nếu các biến đều cùng kiểu và cùng giá trị, có thể khai báo gộp như sau:

<var1>=<var2>= ...=<varn> = <gt>

Ví dụ:

.....

II. CÁC PHÉP TOÁN TRÊN MỘT SỐ KIỂU DỮ LIỆU CƠ BẢN

1. Phép toán trên dữ liệu kiểu số:

Sau đây là các phép toán trên dữ liệu kiểu số bao gồm: số nguyên và số thực

Toán tử	Ý nghĩa	Ví dụ
+	Cộng	$x + y + 2$
-	Trừ	$x - y - 2$
*	Nhân	$x * y$
/	Chia	x / y
%	Lấy phần dư của phép chia (mod)	$x \% y$
//	Lấy phần nguyên của phép chia (div)	$x // y$
**	Lũ thừa	$x ** y (x^y)$

- Tất cả các phép toán đều được thực hiện từ trái sang phải, riêng phép lũy thừa (**)

thì thực hiện từ phải sang trái. Ví dụ: **4**2**3 sẽ được thực hiện**

4(2**3)=4**8=65536**

- Thứ tự thực hiện các phép tính như sau: phép lũy thừa ** có ưu tiên cao nhất, sau đó là các phép toán /, *, //, %, cuối cùng là các phép toán +, -. Chú ý: Nếu có ngoặc thì biểu thức trong ngoặc được ưu tiên thực hiện trước.

Ví dụ:

.....

.....

.....

.....

.....

2. Phép toán trên dữ liệu kiểu chuỗi ký tự:

Các phép toán trên dữ liệu kiểu chuỗi: + (nối chuỗi) và * (lặp)

Ví dụ:

```
>>> s1 = "Hà Nội"
```

```
>>> s2 = "Việt Nam"
```

```
>>> s1 + s2 # Phép nối + nối hai chuỗi ký tự.
```

```
" Hà Nội Việt Nam"
```

```
>>> "123" * 5 # Phép * n lặp n lần chuỗi gốc.
```

```
"123123123123123"
```

```
>>> s*0 # Phép *n với số  $n \leq 0$  thì được kết quả là chuỗi rỗng.
```

III. TỪ KHÓA

- Từ khóa là các từ đặc biệt tham gia vào cấu trúc của ngôn ngữ lập trình
- Không được phép đặt tên biến hay các định danh trùng với từ khóa
- Một số từ khóa trong Python phiên bản 3.x.

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

BÀI 18

CÁC LỆNH VÀO RA ĐƠN GIẢN (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết và thực hiện được một số lệnh vào ra đơn giản
- Thực hiện được một số chuyển đổi dữ liệu giữa các kiểu dữ liệu cơ bản

I. CÁC CÂU LỆNH VÀO RA ĐƠN GIẢN

Các lệnh vào ra đơn giản của Python bao gồm lệnh `input()` và lệnh `print()`

1. Đưa dữ liệu vào:

- Trong Python, để nhập dữ liệu từ bàn phím ta dùng hàm `input()`
- Cú pháp hàm **`input()`** như sau:

.....<Tên biến>=input(<Câu thông báo>)

- Giá trị nhập vào của hàm `input()` mặc định là kiểu xâu.
- Nếu ta muốn giá trị nhập vào là kiểu số nguyên thì ta dùng thêm câu lệnh ép kiểu `int()` để chuyển xâu nhập vào thành số nguyên. (số thực sẽ dùng hàm `float()`)

Ví dụ 1:

```
a=input().....# nhập giá trị cho biến a (mặc định là kiểu xâu)

b=input('Nhập b=').....# nhập giá trị cho biến b (mặc định là kiểu xâu)

c=int(input('Nhập c=')).....# nhập giá trị cho biến c (ép thành kiểu int ngay khi nhập)
```

Ví dụ 2:

Để nhập số nguyên M từ bàn phím ta có thể thực hiện như sau:

Cách 1:

```
.....M=input("Mời bạn nhập số M=")

.....M=int(M)
```

Cách 2:

```
.....M=int(input("Mời bạn nhập số M="))
```

2. Xuất dữ liệu ra:

- Lệnh `print()` có chức năng đưa dữ liệu ra thiết bị chuẩn, thường là màn hình. Thông tin cần đưa ra có thể bao gồm một hay nhiều dữ liệu với kiểu khác nhau, cho phép cả biểu thức tính toán.

- Cú pháp cơ bản như sau:

`print(<danh sách dữ liệu ra>)`

- Nếu <danh sách dữ liệu ra> gồm nhiều đối tượng, thì các đối tượng đó phân cách nhau bởi dấu ,

Ví dụ:

```
n=int(input("Lớp bạn có bao nhiêu người?"))
```

```
print("Vậy bạn có ", n-1, "người bạn")
```

II. CHUYỂN ĐỔI KIỂU DỮ LIỆU CƠ BẢN CỦA PYTHON

- Một số kiểu dữ liệu cơ bản của Python bao gồm: `int` (số nguyên), `float` (số thực), `str` (xâu kí tự), `bool` (logic).
- Lệnh `type()` dùng để nhận biết kiểu dữ liệu của biến trong Python.

Quan sát các lệnh sau để biết kiểu dữ liệu của mỗi biến.

```
>>> n,x,s = 10,1.8,"One"  ← Gán n = 10, x = 1.8, s = "One" đồng thời.
>>> n
10
>>> type(n)
<class 'int'>  ← Biến n thuộc kiểu int – số nguyên.
>>> type(x)
<class 'float'>  ← Biến x thuộc kiểu float – số thực.
>>> type(s)
<class 'str'>  ← Biến s thuộc kiểu str – xâu kí tự.
```

Xác định kiểu và ghi giá trị của các biểu thức sau:

a. "15+20-7"

b. 32>45

c. $13! = 8 + 5$

d. $1 == 2$

✂ Cách chuyển đổi dữ liệu từ kiểu này sang kiểu khác

Lệnh int () có chức năng chuyển đổi số thực hoặc xâu chứa số nguyên (không chuyển đổi được xâu chứa số thực) thành số nguyên.

Ví dụ:

.....

.....

.....

Lệnh float () dùng để chuyển đổi số nguyên và xâu kí tự thành số thực.

Ví dụ:

.....

.....

.....

Lệnh str () dùng để chuyển đổi các kiểu dữ liệu khác thành xâu kí tự.

Ví dụ:

.....

.....

.....

Chú ý: Các lệnh int (), float () chỉ có thể chuyển đổi các xâu ghi giá trị số trực tiếp, không chuyển đổi xâu có công thức.

Ví dụ:

.....

.....

.....

✂ Cách nhập dữ liệu số nguyên hoặc số thực từ bàn phím

<biến> = int(input())

<biến> = float(input())

Ví dụ:.....
.....
.....
.....

CỦNG CỐ KIẾN THỨC BÀI 16 + 17 +18

Bài 1. Hãy viết lệnh để tính giá trị các biểu thức sau trong chế độ gõ lệnh trực tiếp của Python:

- a) $10+13$ b) $20-7$ c) $3 \times 10 - 16$ d) $12/5 + 13/6$

.....
.....
.....
.....

Bài 2. Các lệnh sau có lỗi không? Vì sao?

>>> 3 + * 5

>>> "Bạn là học sinh, bạn tên là "Nguyễn Việt Anh" "

.....
.....
.....
.....

Bài 3. Viết các lệnh in ra màn hình thông tin như sau:

- a) $1 \times 3 \times 5 \times 7 = 105$ b) Bạn Hoa năm nay 16 tuổi

.....
.....
.....

Bài 4: Lệnh sau sẽ in ra kết quả là gì?

```
>>> print("13 + 10*3/2 - 3*2 = ", 13 + 10*3/2 - 3*2)
```

.....
.....

Bài 5: Các tên biến nào sau đây là hợp lệ:

A. _name

B. 12abc

C. My country

D. m123&b

E. xyzABC

.....

.....

Bài 6: Sau các lệnh dưới đây, các biến x,y nhận giá trị bao nhiêu?

```
>>> x = 10
```

```
>>> y = x**2 - 1
```

```
>>> x = x/2 + y
```

.....

.....

Bài 7:

Mỗi lệnh sau là đúng hay sai? Nếu đúng thì cho kết quả là bao nhiêu?

```
>>> (12 - 10//2)**2 - 1
```

```
>>> (13 + 45**2)(30//12 - 5/2)
```

.....

.....

Bài 8:

Mỗi lệnh sau cho kết quả là xâu kí tự như thế nào?

```
>>> ""*20 + "010"
```

```
>>> "10"+"0"*5
```

.....

.....

Bài 9: Các tên biến sau có hợp lệ không?

a) _if

b) global

c) nlocal

d) return

e) true

.....

.....

Bài 10: Lệnh sau có lỗi gì?

```
>>> x = 1
```

```
>>> 123a = x + 1
```

.....

.....

Bài 11: Lệnh sau sẽ in ra kết quả gì?

```
>>> print("đồ rê mi " *3 + "pha son la si đô " *2)
```

Bài 12:

Hãy cho biết trước và sau khi thực hiện các lệnh sau, giá trị các biến x, y là bao nhiêu? Em có nhận xét gì về kết quả nhận được?

```
>>> x, y = 10, 7
>>> x, y = y, x
```

Bài 13:

Mỗi lệnh sau sẽ trả lại các giá trị nào?

a) `str(150)` b) `int("1110")` c) `float("15.0")`

Bài 14:

Lệnh nào sau đây sẽ báo lỗi?

A. `int("12.0")` B. `float(13+1)` C. `str(17.001)`

Bài 15: Nhập giá trị cho biến R từ bàn phím là bán kính hình tròn. Viết chương trình tính và in ra kết quả theo mẫu:

Mời nhập vào bán kính hình tròn:

Chu vi hình tròn là:

Diện tích hình tròn là:

Bài 9: Sử dụng Python, hãy viết chương trình tính chu vi và diện tích hình chữ nhật.

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bài 10: Sử dụng Python, hãy viết chương trình nhập vào 3 số nguyên a,b,c và đưa ra màn hình tổng của 3 số vừa nhập.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bài 11: Thực hiện lệnh gán: $x=2, y=5$

Rồi tính giá trị của biểu thức : $(x+y)(x^2+y^2-1)$

.....

.....

.....

.....

.....

Bài 12: Thực hiện lệnh gán $a=2, b=3, c=4$ rồi tính giá trị của biểu thức:

$(a+b+c)(c+b-c)$

.....

.....

.....

.....

.....

Bài 13: Hoàn thành các câu hỏi trắc nghiệm sau:

1. Ngôn ngữ lập trình đang học là gì?

☐ A Python

☐ B C

☐ C C#

☐ D Lua

2. Câu lệnh print() có thể hiện thị được những nội dung gì?

☐ A Ký tự, số, biến số

☐ B Số

☐ C Ký tự, số

☐ D Biến số

3. Có tất cả bao nhiêu kiểu dữ liệu đã học.

☐ A 1

☐ B 2

☐ C 4

☐ D 3

4. Biến a = 1, thuộc kiểu dữ liệu nào.

☐ A float

☐ B bool

☐ C int

☐ D str

5. Hãy chọn phương án ghép phù hợp nhất. Ngôn ngữ lập trình là gì?

☐ A Ngôn ngữ Pascal hoặc C

☐ B Phương tiện diễn đạt thuật toán

☐ C Phương tiện để soạn thảo văn bản trong đó có chương trình

☐ D Phương tiện diễn đạt thuật toán để máy tính thực hiện công việc

6. Phát biểu nào dưới đây là hợp lí nhất khi nói về biến?

- | | | | |
|----------------------------|---|----------------------------|--|
| ☐ A | Biến có thể đặt hoặc không đặt tên gọi | ☐ B | Biến là đại lượng được đặt tên và có giá trị thay đổi trong quá trình thực hiện chương trình |
| ☐ C | Biến là đại lượng nhận giá trị trước khi chương trình thực hiện | ☐ D | Biến có thể lưu trữ nhiều loại giá trị khác nhau |

7. Các thành phần của ngôn ngữ lập trình là?

- | | | | |
|----------------------------|--|----------------------------|---|
| ☐ A | Tên dành riêng, tên chuẩn và tên do người lập trình định nghĩa | ☐ B | Bảng chữ cái, cú pháp, ngữ nghĩa |
| ☐ C | Chương trình thông dịch và chương trình biên dịch | ☐ D | Chương trình dịch, bảng chữ cái, cú pháp, ngữ nghĩa |

8. Phát biểu nào dưới đây là hợp lí nhất

- | | | | |
|----------------------------|---|----------------------------|--|
| ☐ A | Biến là đại lượng nhận giá trị trước khi chương trình thực hiện | ☐ B | Biến dùng trong chương trình phải khai báo |
| ☐ C | Biến có thể lưu trữ nhiều loại giá trị khác nhau | ☐ D | Biến được chương trình dịch bỏ qua |

9. Tên chuẩn là gì?

- | | | | |
|----------------------------|---|----------------------------|-------------|
| ☐ A | Là đại lượng được đặt tên | ☐ B | Là từ khóa |
| ☐ C | Là những tên đã được định nghĩa thuộc thư viện chuẩn của Python | ☐ D | Đáp án khác |

10. Tên trong Python không bắt đầu bằng chữ số?

- | | | | |
|----------------------------|-----|----------------------------|------|
| ☐ A | Sai | ☐ B | Đúng |
|----------------------------|-----|----------------------------|------|

11. Tên trong Python dùng các ký tự đặc biệt như !, @, #, \$, %, &

- | | | | |
|----------------------------|-----|----------------------------|------|
| ☐ A | Sai | ☐ B | Đúng |
|----------------------------|-----|----------------------------|------|

12. Tên trong Python được phân biệt chữ hoa, chữ thường.

- | | | | |
|----------------------------|-----|----------------------------|------|
| ☐ A | Sai | ☐ B | Đúng |
|----------------------------|-----|----------------------------|------|

13. Chọn tên đúng trong Python

☐ A ví@dụ

☐ B Bài tập

☐ C #4ví dụ

☐ D 3_phuongtrinh

14. Các từ khoá nào trong Python được viết hoa ở đầu từ?

☐ A True, False, None

☐ B Math, False, None

☐ C True, And, None

☐ D True, False, In

15. Hãy chọn câu lệnh dùng để tính chu vi hình tròn trong Python

☐ A $c = 3.14 * 2 * r$

☐ B $c == 3.14 ** 2 ** r$

☐ C $c == 3.14 * 2 * r$

☐ D $c = 3.14 ** 2 ** r$

16. Phát biểu nào đúng về ngôn ngữ lập trình Python

☐ A Là ngôn ngữ MicroBit

☐ B Là ngôn ngữ lập trình cấp cao

☐ C Là ngôn ngữ lập trình cấp thấp

☐ D Là ngôn ngữ máy

17. Kết quả in ra màn hình của lệnh **print("Xin chào", "123")** là gì?

☐ A 123

☐ B Lệnh sai cú pháp

☐ C Xin chào 123

☐ D Xin chao

18. Nhóm lệnh **print ("Hà Nội"); print ("Việt Nam")** sẽ in ra kết quả như thế nào ?

☐ A Hà Nội Việt Nam

☐ B Hà NộiViệt Nam

☐ C "Hà Nội" "Việt Nam"

☐ D Hà NộiViệt Nam

19. Trong các lệnh sau, lệnh nào là lệnh python đúng ?

☐ A print ("Hello Word")

☐ B print ("Hello Word")

☐ C Print('Hello Word')

☐ D print (Hello Word)

☐ E Print ("Hello Word")

20. Trong Python kết quả của phép chia $9 / 2$ là?

- | | | | |
|----------------------------|-----|----------------------------|---|
| ☐ A | 0 | ☐ B | 1 |
| ☐ C | 4.5 | ☐ D | 4 |

21. Ngôn ngữ lập trình là gì?

- | | | | |
|----------------------------|---|----------------------------|---|
| ☐ A | Ngôn ngữ dùng để viết chương trình máy tính gọi là ngôn ngữ lập trình | ☐ B | Là ngôn ngữ dưới dạng nhị phân để máy tính có thực hiện trực tiếp |
| ☐ C | Chính là chương trình dịch | ☐ D | Chỉ là một quy ước để diễn tả thuật toán |

22. Tên một số ngôn ngữ lập trình bậc cao?

- | | | | |
|----------------------------|------------------------|----------------------------|----------------------------------|
| ☐ A | Pascal, Python, C, C++ | ☐ B | Python, C, C#, Google Chrome |
| ☐ C | Word, C, C++, Cốc Cốc | ☐ D | Notepad, Excel, C, Control Panel |

23. Chương trình được tạo bởi ngôn ngữ lập trình Python có đuôi là:

- | | | | |
|----------------------------|------|----------------------------|-------|
| ☐ A | .gif | ☐ B | .pas |
| ☐ C | .py | ☐ D | .docx |

24. In thông điệp "Hello" ra màn hình sử dụng câu lệnh nào trong Python?

- | | | | |
|----------------------------|------------------------------|----------------------------|-------------------------------|
| ☐ A | <code>write ('Hello')</code> | ☐ B | <code>print ("Hello");</code> |
| ☐ C | <code>print ("Hello")</code> | ☐ D | <code>input (Hello);</code> |

25. Đây là quy tắc đúng khi đặt tên cho biến trong Python?

- | | | | |
|----------------------------|---|----------------------------|--|
| ☐ A | Tên biến có thể có các ký hiệu như !, @, #, \$, % | ☐ B | Tên biến có thể bắt đầu bằng dấu gạch dưới "_" |
| ☐ C | Có thể sử dụng keyword làm tên biến | ☐ D | Tên biến có thể bắt đầu bằng một chữ số |

26. Khẳng định nào sau đây về Python là đúng?

- | | | | |
|----------------------------|--|----------------------------|--|
| ☐ A | Python là một ngôn ngữ lập trình bậc cao | ☐ B | Python là ngôn ngữ lập trình hướng đối tượng |
| ☐ C | Python gần với ngôn ngữ tự nhiên | ☐ D | Tất cả các ý trên đều đúng |

27. Tên biến đúng trong Python là?

- | | |
|-------------------------------------|-------------------------------------|
| ☐ A bai-2 | ☐ B 3bai |
| ☐ C bai2 | ☐ D Bai?2 |

28. Viết câu lệnh hiển thị dòng chữ "Lập trình cùng NNLT Python thật dễ "

- | | |
|--|--|
| ☐ A print('Lập trình cùng NNLT Python thật dễ') | ☐ B print(Lập trình cùng NNLT Python thật dễ) |
| ☐ C print('Lập trình cùng NNLT Python thật dễ'); | ☐ D prin('Lập trình cùng NNLT Python thật dễ') |

29. Cho bài toán sau

Tìm tuổi của mẹ. Biết rằng năm nay con 12 tuổi và con ít hơn mẹ 21 tuổi
Viết câu lệnh hiển thị tuổi của mẹ trên màn hình bằng NNLT Python.

- | | |
|--|--|
| ☐ A print(12+21); | ☐ B 12+21 |
| ☐ C print('12+21') | ☐ D print(12+21) |

30. Hãy chỉ ra tên biến SAI trong các tên dưới đây

- | | |
|----------------------------------|-----------------------------------|
| ☐ A _a | ☐ B a1 |
| ☐ C 1a | ☐ D _1a |

31. Ký hiệu phép toán chia lấy phần dư trong Python là

- | | |
|--|---|
| ☐ A % (phần trăm) | ☐ B // |
| ☐ C : (dấu 2 chấm) | ☐ D / (xuyệt) |

32. Biểu thức số học nào sau đây là SAI trong Python?

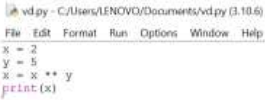
- | | |
|--|---|
| ☐ A 20 % 3 * (5 + 8) | ☐ B 20 ** 3*(5 + 8) |
| ☐ C (20 - 3)(20 + 2) | ☐ D (20 // 3)*(5 + 8) |

33. Câu lệnh gán nào sau đây là đúng trong Python?

- | | |
|--|--|
| ☐ A x == 5*8 | ☐ B x != 5*8 |
| ☐ C x = 5*8 | ☐ D x := 5*8 |

34.  Thực hiện chương trình sau đây, nhận được kết quả trên màn hình là gì?

- | | | | |
|----------------------------|-----|----------------------------|-----------|
| ☐ A | 6 | ☐ B | 0 |
| ☐ C | 7.5 | ☐ D | 0.8333334 |

35.  Thực hiện chương trình sau đây, nhận được kết quả trên màn hình là gì?

- | | | | |
|----------------------------|----|----------------------------|----|
| ☐ A | 2 | ☐ B | 0 |
| ☐ C | 10 | ☐ D | 32 |

36. Câu lệnh nào đúng trong các câu lệnh sau?

- | | | | |
|----------------------------|---------------|----------------------------|---------------|
| ☐ A | Print("N="5) | ☐ B | print("N="5) |
| ☐ C | Print("N=",5) | ☐ D | print("N=",5) |

37. Tập hợp các ký tự để viết chương trình được gọi là

- | | | | |
|----------------------------|-----------|----------------------------|--------------|
| ☐ A | Ngữ nghĩa | ☐ B | Cú pháp |
| ☐ C | Tập hợp | ☐ D | Bảng chữ cái |

38. Trong toán học ghi $8^4 - \frac{7}{2+8} - 40$, trong python sẽ ghi là

- | | | | |
|----------------------------|-----------------|----------------------------|----------------------------|
| ☐ A | 8**4 - 7/2+8-40 | ☐ B | 8**4-7/(2+8-40) |
| ☐ C | 8**4-7/(2+8)-40 | ☐ D | $8^4 - \frac{7}{2+8} - 40$ |

39.



Bạn gõ lệnh `print(8*8-100)` khi chạy chương trình sẽ cho kết quả bao nhiêu

☐ A

64

☐ B

100

☐ C

84

☐ D

-36

40.

Trong toán học ghi $7 \times 2 - \frac{4}{6}$, trong python sẽ ghi là

☐ A

$7 \cdot 2 - 4:6$

☐ B

$7*2-4/6$

☐ C

$7 \times 2 - \frac{4}{6}$

☐ D

$7 * 2 - 4:6$

41.

$15 // 5$ cho kết quả?

☐ A

0

☐ B

15

☐ C

5

☐ D

3

42.

Câu lệnh sau đây cho kết quả `print('70+9=',70+9)`

☐ A

$70+9=79$

☐ B

$70+9=70+9$

43.

Kết quả $23 // 5 = ?$

☐ A

7

☐ B

4.60

☐ C

4

☐ D

3

44.

Hãy chọn biểu thức trong Pascal tương ứng với biểu thức toán học sau

$$\frac{2x^2 - 3x + 1}{x^2 + 1}$$

☐ A

$(2*x*x-3*x+1)/x*x+1$

☐ B

$2*x*x-3*x+1/x*x+1$

☐ C

$(2*x*x-3*x+1)/(x*x+1)$

☐ D

$2*x*x-3*x+1/(x*x+1)$

BÀI 19: CÂU LỆNH RỄ NHÁNH IF (1 LT + 1TH)

Sau bài này em sẽ:

23

- Biết và trình bày được các phép toán với kiểu dữ liệu logic
- Biết sử dụng được lệnh rẽ nhánh if trong lập trình

I. BIỂU THỨC LOGIC

- Trong Python, biểu thức logic là biểu thức chỉ nhận giá trị True (đúng) hoặc False (sai). Biểu thức logic đơn giản nhất là các biểu thức so sánh số hoặc xâu kí tự.

- Các phép so sánh giá trị số trong Python

<	Nhỏ hơn	>	Lớn hơn	==	Bằng nhau
<=	Nhỏ hơn hoặc bằng	>=	Lớn hơn hoặc bằng	!=	Khác nhau

- Các phép toán trên kiểu dữ liệu logic bao gồm phép and (và), or (hoặc) và not (phủ định).
- Bảng các phép toán logic như sau:

Phép toán or		
X	Y	X or Y
True	True	
True	False	
False	True	
False	False	

đoán

Phép toán and		
X	Y	X and Y
True	True	
True	False	
False	True	
False	False	

Phép toán not	
X	
True	
False	

- Ví dụ 1: dự kết quả của các giá

trị logic a,b,c

```
>>> x, y, z = 10, 5, 9
>>> b = x < 11 and z > 5
>>> c = x > 15 or y < 9
>>> a = not b
```

- Ví dụ 2: Mỗi biểu thức sau có giá trị True hay False?
 - 100%4 == 0
 - 111//5 != 20 or 20%3 != 0

II. LỆNH IF

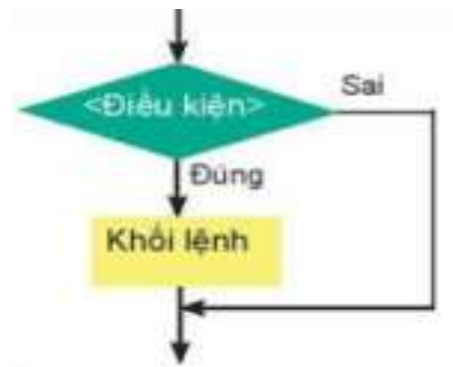
Cấu trúc rẽ nhánh chia làm 2 loại: dạng thiếu và dạng đủ

1. Cấu trúc if dạng thiếu

if <điều kiện>:

<Khởi lệnh>

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> đúng thì thực hiện <khởi lệnh>, ngược lại thì bỏ qua sang lệnh tiếp theo sau lệnh if.



Hình 19.2

nếu
chuyển

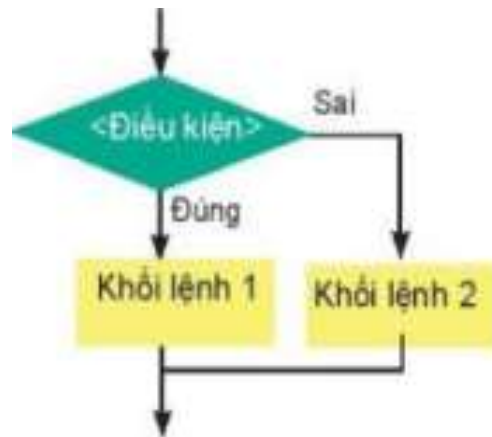
2. Cấu trúc if dạng đủ

if <điều kiện>:

<khởi lệnh 1>

else:

<khởi lệnh 2>



Hình 19.3

Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện> nếu đúng thì thực hiện <khởi lệnh 1>, ngược lại thì thực hiện <khởi lệnh 2>.

Ví dụ: Viết chương trình nhập số nguyên dương n từ bàn phím. Hãy kiểm tra và đưa ra màn hình thông báo n là số chẵn hay số lẻ?

- Dạng thiếu:

```
n=int(input("Mời bạn nhập số n="))
if n%2==0:
    print(n,"là số chẵn")
if n%2 != 0:
    print(n,"là số lẻ")
```

- Dạng đủ:

```
n=int(input("Mời bạn nhập số n="))
if n%2==0:
    print(n,"là số chẵn")
else:
    print(n,"là số lẻ")
```

III. THỰC HÀNH

Bài 1: Viết chương trình nhập số nguyên dương n từ bàn phím. Hãy kiểm tra và đưa ra màn hình thông báo n có chia hết cho cả 3 và 7 hay không?

.....

.....

.....

.....

.....

.....

.....

.....

Bài 2: Nhập số năm từ bàn phím, đưa ra màn hình thông báo số ngày của năm đó, biết rằng năm nhuận là năm chia hết cho 400 hoặc **chia hết cho 4 và không chia hết cho 100**

.....

.....

.....

.....

.....

.....

.....

.....

.....

Bài 3: Giá bán cam tại siêu thị tính như sau: nếu khối lượng cam mua dưới 5 kg thì giá bán là 12.000 đồng/kg, nếu khối lượng mua lớn hơn hoặc bằng 5 kg thì giá bán là 10.000 đồng/kg. Viết chương trình nhập số lượng mua (tính theo kg) sau đó tính số tiền phải trả.

.....

.....

.....

.....

.....

.....

.....

.....

.....

BÀI 20: CÂU LỆNH LẶP FOR (1 LT + 1TH)

Sau bài này em sẽ:

- Biết được ý nghĩa của vùng giá trị tạo bởi lệnh `ranger()`.
- Biết được chức năng của lện lặp `for` và cách dùng trong Python.

I. LỆNH FOR

- `for` là lệnh lặp với số lần biết trước. Số lần lặp thường được xác định bởi vùng giá trị của lệnh `range()`.
- Cú pháp của lệnh lặp với số lần biết trước `for` trong Python như sau:

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

```
<khối lệnh lặp>
```

- Khi thực hiện, ở mỗi vòng lặp biến `i` sẽ được gán lần lượt các giá trị trong vùng giá trị của lệnh `range()` và thực hiện <khối lệnh lặp>

- Lệnh `range(n)` trả lại vùng giá trị gồm n số từ 0 đến $n - 1$.

Ví dụ 1: Tính tổng các số tự nhiên chẵn nhỏ hơn n , với n cho trước ($n=10$).

.....

.....

.....

.....

.....

.....

Ví dụ 2: Đếm các số nguyên nhỏ hơn n ($n=20$) và là bội của 3.

.....

.....

.....

.....

.....

II. LỆNH RANGE

- Lệnh tạo vùng giá trị range() có dạng như sau :

+ range(stop) trả lại vùng giá trị từ 0 đến stop – 1.

+ range(start, stop) trả lại vùng giá trị từ start đến stop – 1.

Ví dụ:

.....

.....

.....

Hãy biểu diễn các dãy số sau bằng lệnh range()

– 1,2,3,...50.....

– 5,6,7,8,9,10.....

– 0,1.....

– 10.....

III. THỰC HÀNH

Bài 1: Đoạn chương trình sau in ra kết quả là gì?

```
n = int(input("Nhập số tự nhiên n:"))
S = 0
for k in range(n+1):
    S = S + k
print(S*S)
```

Bài 2: Viết đoạn chương trình tính tích $1 \times 2 \times 3 \times \dots \times n$ với n được nhập từ bàn phím

.....

.....

.....

.....

.....

.....

.....

Viết chương trình nhập từ bàn phím số tự nhiên n và in ra kết quả $S = 1 + \frac{1}{2} + \dots + \frac{1}{n}$.

Bài 3:

.....

.....

.....

.....

.....

.....

.....

.....

Bài 4: Viết chương trình nhập từ bàn phím số tự nhiên n và in ra kết quả là tổng sau:

$$S = 1^3 + 2^3 + \dots + n^3.$$

.....

.....

.....

.....

.....

.....

.....

BÀI 21: CÂU LỆNH LẶP WHILE (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết và thực hành giải các bài toán sử dụng lệnh lặp while với số lần không biết trước
- Biết ba cấu trúc lập trình cơ bản: tuần tự, rẽ nhánh, lặp,...

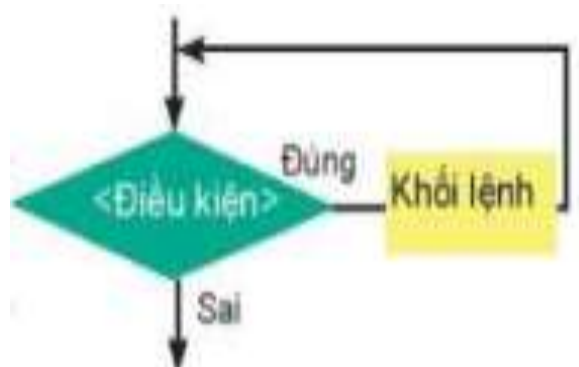
I. LỆNH LẶP WHILE

- Lệnh lặp while thực hiện khối lệnh với số lần lặp không biết trước. Khối lệnh lặp được thực hiện cho đến khi <điều kiện> = False
- Cú pháp của lệnh while như sau:

while <điều kiện>:

<khối lệnh lặp>

- **Chú ý:** sau dấu “:” khối lệnh lặp cần được viết lùi vào và thẳng hàng. Mặc định các lệnh sẽ lùi vào 1 tab hoặc 4 dấu cách.



Hình 21.1

- Trong đó <điều kiện> là biểu thức logic. Khi thực hiện lệnh, Python sẽ kiểm tra <điều kiện>, nếu đúng thì thực hiện khối lệnh lặp, nếu sai thì kết thúc lệnh while.
- Vì lệnh while không biết trước số lần lặp, mà phụ thuộc vào điều kiện. Do đó, cần chú ý đến điều kiện của lệnh while để tránh bị lặp vô hạn.

II. CẤU TRÚC LẬP TRÌNH

Ba cấu trúc lập trình cơ bản của các ngôn ngữ lập trình bậc cao gồm: cấu trúc tuần tự, cấu trúc rẽ nhánh, cấu trúc lặp.

III. THỰC HÀNH

Bài 1: Viết chương trình in toàn bộ dãy các số tự nhiên từ 1 đến 100 trên một hàng ngang

.....

.....

.....

.....

.....

.....

.....

Bài 2: Viết chương trình in ra màn hình dãy các chữ cái tiếng Anh từ “A” đến “Z” theo ba hàng ngang trên màn hình, hai hàng ngang đầu có 10 chữ cái, hàng thứ ba có 6 chữ cái.

.....

.....

.....

.....

.....

.....

.....

Bài 3: Cho dãy số 1, 4, 7, 10, ... Tìm phân tử lớn nhất của dãy nhưng nhỏ hơn 100

.....

.....

.....

.....

.....

.....

.....

Bài 4: Viết chương trình đếm trong dãy 100 số tự nhiên đầu tiên có bao nhiêu số thỏa mãn điều kiện: hoặc chia hết cho 5 hoặc chia cho 3 dư 1.

.....

.....

.....

.....

.....

.....

.....

Bài 5: Viết chương trình in ra các số tự nhiên từ 1 đến 100 ra màn hình thành 10 hàng, mỗi hàng 10 số, có dạng như sau:

1 2 3 ... 10

11 12 ... 20

.....

91 92 ... 100

.....

.....

.....

.....

.....

BÀI 22: KIỂU DỮ LIỆU DANH SÁCH (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết được kiểu dữ liệu danh sách (list), cách khởi tạo và truy cập từng phần tử của danh sách
- Biết và thực hiện được cách duyệt các phần tử của danh sách bằng lệnh for
- Thực hành được một số phương thức đơn giản trên dữ liệu danh sách

I. KIỂU DỮ LIỆU DANH SÁCH

- List là kiểu dữ liệu danh sách (dãy, mảng) trong Python. Tạo list bằng lệnh gán với các phần tử trong cặp dấu ngoặc []. Các phần tử của danh sách có thể có các kiểu dữ liệu khác nhau.
- Truy cập hoặc thay đổi giá trị của từng phần tử thông qua chỉ số:
`<danh sách>[<chỉ số>]`
- Chỉ số của danh sách bắt đầu từ 0 đến `len() - 1`, trong đó `len()` là lệnh tính độ dài danh sách.
- Khởi tạo kiểu dữ liệu danh sách trong Python:

<code><tên list> = [<v₁>, <v₂>, ..., <v_n>]</code>
--

Trong đó:

các giá trị `<vk>` có thể có kiểu dữ liệu khác nhau (số nguyên, số thực, xâu kí tự...). Ví dụ:

.....
.....
.....
.....
.....
.....
.....

✂ Lệnh `del` để xóa một phần tử của danh sách

Ví dụ:
.....
.....

✂ Lệnh `len()` để tính độ dài danh sách

Ví dụ:
.....
.....
.....

✂ Ghép danh sách bằng cách sử dụng phép cộng (+)

Ví dụ:
.....
.....

II. DUYỆT CÁC PHẦN TỬ CỦA DANH SÁCH

- Có thể duyệt lần lượt các phần tử của danh sách bằng lệnh for kết hợp với vùng giá trị của lệnh range().
- Cho biết kết quả của từng ví dụ sau:

Ví dụ 1: Duyệt và in ra từng phần tử của danh sách.

```
>>> A = [1,2,3,4,5]
>>> for i in range (len(A)):
    print (A[i], end=" ")
.....
```

Ví dụ 2: Duyệt và in một phần của danh sách.

```
>>> A = [3, 2, 1, 5, 6, 10, 7, 12, 18]
>>> for i in range(2,5):
    print (A[i], end=" ")
.....
```

III. THÊM PHẦN TỬ VÀO DANH SÁCH

- Python có những lệnh đặc biệt để thêm phần tử vào một danh sách. Các lệnh này được thiết kế riêng cho kiểu dữ liệu danh sách và còn được gọi là phương thức (method) của danh sách.
- Python có một số lệnh dành riêng (phương thức) cho dữ liệu kiểu danh sách. Cú pháp các lệnh đó như sau:

<danh sách>.<phương thức>

- Lệnh thêm phần tử vào cuối danh sách là **<danh sách>.append()**

Ví dụ:

.....

.....

.....

.....

IV. THỰC HÀNH

Bài 1: Cho danh sách $A = [1, 0, \text{"One"}, 9, 15, \text{"Two"}, \text{True}, \text{False}]$. Hãy cho biết giá trị các phần tử:

a) $A[0]$

b) $A[2]$

c) $A[7]$

d) $A[\text{len}(A)]$

Bài 2: Giả sử A là một danh sách các số, mỗi lệnh sau thực hiện gì?

a) $A=A+[10]$

b) $A=100+A$

c) $\text{del } A[7]$

d) $A=A[1]*25$

Bài 3: Giải thích các lệnh ở mỗi câu sau thực hiện công việc gì?

a) `>>> S = 0`

`>>> for i in range(len(A)):`

`if A[i] > 0: S = S + A[i]`

`>>> print(S)`

.....

.....

b) `>>> C = 0`

`>>> for i in range(len(A)):`

`if A[i] > 0: C = C + 1`

`>>> print(C)`

.....

.....

.....

Bài 4: Cho dãy các số nguyên A , viết chương trình in ra các số chẵn của A .

.....

.....

.....

.....

.....

.....

.....

Bài 5: Sau khi thêm một phần tử vào danh sách A bằng lệnh `append()` thì độ dài danh sách A thay đổi như thế nào?

.....

.....

.....

Bài 6: Danh sách A sẽ như thế nào sau các lệnh sau?

```
>>> A = [2,4,10,1,0]
>>> A.append(100)
>>> del(A[1])
```

.....

.....

.....

Bài 7: Cho dãy số A. Viết chương trình tìm giá trị và chỉ số của phần tử lớn nhất của A

.....

.....

.....

.....

.....

.....

BÀI 23: MỘT SỐ LỆNH LÀM VIỆC VỚI DỮ LIỆU DANH SÁCH (1 LT + 2 TH)

Sau bài này em sẽ:

- Biết cách duyệt danh sách bằng toán tử `In`

- Biết và thực hiện được một số phương thức thường dùng với danh sách

I. DUYỆT DANH SÁCH VỚI TOÁN TỬ IN

- Tính toán tử in dùng để kiểm tra một phần tử có nằm trong danh sách đã cho không. Kết quả trả lại True (Đúng) hoặc False (Sai).

<giá trị> in <danh sách>

- Có thể duyệt nhanh từng phần tử của danh sách bằng toán tử in và lệnh for mà không cần sử dụng lệnh range ().

Ví dụ:.....

.....

.....

.....

.....

.....

.....

.....

.....

II. MỘT SỐ LỆNH LÀM VIỆC VỚI DANH SÁCH

A.append(x)	<u>Bổ sung phần tử x vào cuối danh sách A</u>
A.insert(k, x)	<u>Chèn phần tử x vào vị trí k của danh sách A</u>
A.clear()	<u>Xóa toàn bộ dữ liệu của danh sách A</u>
A.remove(x)	<u>Xóa phần tử x từ danh sách</u>

Hãy điền kết quả của sau khi thực hiện các câu lệnh sau trên danh sách A

A=[1,2,6,10,16,0]		
1	A.remove(1)	
2	A.remove(10)	
3	A.append(9)	

4	A.insert(2,5)	
5	A.clear()	
6.	A.insert (-10,8)	
7.	A.insert (100,7)	

(nếu $k < 0$ thì chèn vào đầu danh sách, nếu $k > \text{len}()$ thì chèn vào cuối danh sách)

III. THỰC HÀNH

Bài 1: Cho dãy số [1, 2, 2, 3, 4, 5, 5]. Viết lệnh thực hiện:

A) Chèn số 1 vào ngay sau giá trị 1 của dãy.

B) Chèn số 3 và số 4 vào danh sách để dãy có số 3 và số 4 liền nhau hai lần.

.....

.....

.....

.....

Bài 2: Cho trước dãy số A. Viết chương trình thực hiện dãy công việc sau:

- Xóa đi một phần tử ở chính giữa dãy nếu số phần tử của dãy là số lẻ
- Xóa đi hai phần tử ở chính giữa của dãy nếu số phần tử của dãy là số chẵn

.....

.....

.....

.....

BÀI 24: XÂU KÍ TỰ (1 LT + 1 TH)

Sau bài này em sẽ:

- Hiểu được chuỗi ký tự là kiểu dữ liệu cơ bản của Python
- Biết và thực hiện được lệnh for để xử lý chuỗi ký tự

I. XÃU LÃ MỘT DÃY CÃC KÍ TỰ

- Chuỗi ký tự trong Python là dãy các ký tự Unicode.
- Chuỗi có thể được coi là danh sách các ký tự nhưng **không thay đổi** từng ký tự của chuỗi.
- Truy cập từng ký tự của chuỗi qua chỉ số, chỉ số từ 0 đến độ dài `len()` – 1.
- Chuỗi rỗng được định nghĩa như sau: **empty**= “ ”

Ví dụ:

.....

.....

.....

.....

.....

.....

II. LỆNH DUYỆT KÍ TỰ CỦA XÃU

- **Cách thứ nhất**, biến `i` lần lượt chạy theo chỉ số của chuỗi ký tự `s`, từ 0 đến `len(s) – 1`. Ký tự tại chỉ số `i` là `s[i]`.
- **Cách thứ hai** duyệt theo từng ký tự của chuỗi `s`. Biến `ch` sẽ được gán lần lượt các ký tự của chuỗi `s` từ đầu đến cuối.
- **Chú ý:** Từ khóa `in`, tùy trường hợp cụ thể, hoặc là toán tử logic dùng để kiểm tra một giá trị có mặt hay không trong một vùng giá trị/danh sách/ chuỗi, hoặc để chọn lần lượt từng phần tử trong một vùng giá trị/danh sách/ chuỗi.

Ví dụ:

.....

.....

.....

.....

.....

.....

Có thể duyệt các kí tự của xâu bằng lệnh for tương tự với danh sách. s_1 in s_2 trả lại giá trị True nếu s_1 là xâu con của s_2

III. THỰC HÀNH

Bài 1: Các xâu kí tự sau có hợp lệ không?

- a. “123&*()+-ABC”
- b. “1010110&0101001”
- c. “Tây Nguyên”
- d. $11111111 = 256$

Bài 2: Mỗi xâu hợp lệ ở câu 1 có độ dài bằng bao nhiêu?

.....

.....

Bài 3: Sau khi thực hiện các lệnh sau, biến skq sẽ có giá trị bao nhiêu?

>>> s = “81723”

>>> skq = “”

>>> for ch in s:

if int(ch) % 2 != 0:

skq = skq + ch

.....

.....

.....

Bài 4: Cho $s_1 = \text{“abc”}$, $s_2 = \text{“ababcabca”}$. Các biểu thức logic sau cho kết quả là đúng hay sai?

- a. s_1 in s_2
- b. $s_1 + s_1$ in s_2
- c. “abcabca” in s_2
- d. “abc123” in s_2

.....

.....

.....

.....

.....

.....

BÀI 25:MỘT SỐ LỆNH LÀM VIỆC VỚI XÂU KÍ TỰ (1 LT + 2 TH)

Sau bài này em sẽ:

- Biết và thực hiện được một số lệnh thường dùng với chuỗi ký tự.

I. XÂU CON VÀ LỆNH TÌM VỊ TRÍ XÂU CON

✂ Biểu thức kiểm tra <chuỗi 1> nằm trong <chuỗi 2> là:

<chuỗi 1> in <chuỗi 2>

Nếu đúng thì trả lại giá trị True, nếu sai trả lại giá trị False.

Ví dụ:
.....
.....
.....

✂ Tìm vị trí xuất hiện của một chuỗi trong chuỗi khác

<chuỗi mẹ>find (<chuỗi con>,start)

Ý nghĩa: Tìm vị trí đầu tiên của chuỗi con trong chuỗi mẹ bắt đầu từ vị trí start. Nếu không tìm thấy thì trả về -1

<chuỗi mẹ>find (<chuỗi con>)

Ý nghĩa: Nếu không ghi vị trí start, thì lệnh trên sẽ tìm vị trí xuất hiện đầu tiên của chuỗi con trong chuỗi mẹ và trả về vị trí đó.

Ví dụ:
.....
.....
.....
.....
.....
.....

II. MỘT SỐ LỆNH THƯỜNG DÙNG VỚI XÂU KÍ TỰ

✂ Lệnh split () tách một chuỗi thành danh sách các từ:

Cú pháp của lệnh split()

<xâu mẹ>.split(<kí tự cách>)

Ví dụ:.....
.....
.....
.....
.....
.....

✂ **Lệnh join() nối danh sách gồm các từ thành một xâu.**

Cú pháp của lệnh join() là:

“kí tự nối”. join(<danh sách>)

Ví dụ:.....
.....
.....
.....
.....
.....
.....

III. THỰC HÀNH

Bài 1: Biểu thức logic sau là đúng hay sai?

>>> “010” in “00100”

.....
.....

Bài 2: Lệnh sau trả lại giá trị gì?

>>> “ababababab”.find (“ab”, 4)

.....
.....

Bài 3: Viết chương trình nhập nhiều số (số nguyên hoặc số thực) từ bàn phím, các số cách nhau bởi dấu cách. Sau đó in ra màn hình tổng các số đã nhập.

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Bài 4:Viết chương trình nhập họ tên đầy đủ của người dùng, sau đó in thông báo tên và họ đệm của người đó.

.....
.....
.....
.....
.....
.....
.....

BÀI 26: HÀM TRONG PYTHON (1 LT + 1TH)

Sau bài này em sẽ:

- Biết được chương trình con là hàm
- Biết cách tạo hàm

I. MỘT SỐ HÀM THIẾT KẾ SẴN CỦA PYTHON

abs()	len()	range()	bool()
list()	round()	chr()	input()
str()	divmod()	int()	print()
float()	ord()	type()	

- Python cung cấp sẵn nhiều hàm thực hiện những công việc khác nhau cho phép người dùng được tùy ý sử dụng khi viết chương trình bằng các câu lệnh gọi hàm tương ứng.
- Cú pháp câu lệnh gọi hàm trong Python có dạng chung như sau:

<tên hàm>(<danh sách tham số hàm>)

Ví dụ:.....
.....
.....
.....
.....

II. THIẾT LẬP CÁC HÀM TỰ ĐỊNH NGHĨA

- Hàm trong Python được định nghĩa bằng từ khóa **def**, theo sau là tên hàm (tên hàm sẽ theo quy tắc đặt tên định danh).
- Hàm có thể có hoặc không có tham số.
- Khối lệnh mô tả hàm được viết sau dấu “:” và viết lùi vào, thẳng hàng.
- Hàm có thể có hoặc không có giá trị trả lại sau từ khóa return.

✂ Cú pháp thiết lập hàm có trả lại giá trị

```
def <tên hàm> (<tham số>):
```

```
    <khởi lệnh>
```

```
    return <giá trị>
```

Cần có lệnh **return** <giá trị>. Hàm số kết thúc khi gặp lệnh **return** và trả lại <giá trị>

✂ Cú pháp thiết lập hàm không có trả lại giá trị

```
def <tên hàm> (<tham số>):
```

```
    <khởi lệnh>
```

```
    return
```

Lệnh **return** không có giá trị trả lại. Hàm số kết thúc khi gặp lệnh **return**. Nếu hàm không trả lại giá trị thì có thể không cần lệnh **return**

Ví dụ:

.....

.....

.....

.....

.....

.....

III. THỰC HÀNH

Bài 1: Quan sát các hàm sau, giải thích cách thiết lập và chức năng của mỗi hàm

a) def Nhap_xau() :

```
    msg = input("Nhập một xâu: ")
```

```
    return msg
```

.....

.....

b)def Inday(n) :

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

```
print(k, end = “ “)
```

.....

.....

.....

Bài 2: Trong khi viết hàm có thể có nhiều lệnh return. Quan sát hàm sau và giải thích ý nghĩa của những lệnh return. Hàm này có điểm gì khác so với hàm prime () đã được mô tả trong phần thực hành.

```
1 def prime(n):
2     if n < 2: return False
3     c = 0
4     k = 2
5     while k < n:
6         if n%k == 0: return False
7         k = k + 1
8     return True
```

.....

.....

.....

BÀI 27: THAM SỐ CỦA HÀM (1 LT + 2 TH)

Sau bài này em sẽ:

- Biết cách thiết lập các tham số của hàm. Hiểu được cách truyền giá trị thông qua đối số hàm.
- Biết viết chương trình có sử dụng chương trình con .

I. THAM SỐ VÀ ĐỐI SỐ CỦA HÀM

- Tham số của hàm được định nghĩa khi khai báo hàm và được dùng như biến trong định nghĩa hàm. Đối số là giá trị được truyền vào khi gọi hàm.
- Khi gọi hàm, các tham số (parameter) sẽ được truyền bằng giá trị thông qua đối số (argument) của hàm, số lượng giá trị được truyền vào hàm bằng với số tham số trong khai báo của hàm.

```

1 >>> def f(a,b,c): ← Hàm f() có ba tham số a, b, c.
2     return a+b+c
3 >>> f(1,2,3) ← Hàm f() được gọi với ba giá trị cụ thể.
4 6
5 >>> x,y,z = 10,20,5
6 >>> f(x,y,z) ← Hàm f() được gọi với ba biến đã có giá trị.
7 35
8 >>> f(a,b,c) ← Lỗi gọi hàm bị lỗi nếu các tham số được truyền
    vào chưa có giá trị.
9 Traceback (most recent call last):
10 File "<pyshell#6>", line 1, in <module>
11     f(a,b,c)
12 NameError: name 'a' is not defined
13 >>>

```

II. CÁCH SỬ DỤNG CHƯƠNG TRÌNH CON

- Sử dụng chương trình con có thể giúp phân chia việc giải một bài toán lớn thành giải quyết các bài toán nhỏ và phát huy được tinh thần làm việc nhóm.
- Chương trình chính có cấu trúc rõ ràng, dễ hiểu hơn. Nếu cần hiệu chỉnh, phát triển và nâng cấp cũng thuận tiện hơn.

```

1 def prime(n):
2     C = 0
3     k = 1
4     while k < n:
5         if n%k == 0: C = C + 1
6         k = k + 1
7     if C == 1: return True
8     else: return False
9 # Chương trình chính
10 n = int(input("Nhập số tự nhiên n: "))
11 for k in range(1, n+1) :
12     if prime(k):
13         print(k, end = " ")

```

Trên đây là đoạn chương trình dùng để:

.....

.....

```

1 def tongduong(A) :
2     S=0
3     for k in A:
4         if k > 0: S = S + k
5     return S
6 # Chương trình chính
7 A = [0, 2, -1, 5, 10, -3]
8 B = [1, -10, -11, 8, 2, 0, -5]
9 # Sử dụng hàm tongduong tính tổng các số dương của dãy A
10 print("Tổng các số dương trong dãy A = ",tongduong(A))
11 # Sử dụng hàm tongduong tính tổng các số dương của dãy B
12 print("Tổng các số dương trong dãy B = ",tongduong(B))
13

```

Trên đây là đoạn chương trình dùng để:

.....

.....

III. THỰC HÀNH

Bài 1:Viết chương trình thực hiện. Nhập hai số tự nhiên từ bàn phím, hai số cách nhau bởi dấu phẩy, in ra ước chung lớn nhất (UCLN) của hai số.

.....

.....

.....

.....

.....

.....

Bài 2:Viết chương trình thực hiện. Nhập n số tự nhiên từ bàn phím, hai số cách nhau bởi dấu cách. Tính và in ra tổng của các số này.

.....

.....

.....

.....

.....

.....

BÀI 28: PHẠM VI CỦA BIẾN (1 LT + 2 TH)

Sau bài này em sẽ:

- Biết và trình bày được ý nghĩa của phạm vi hoạt động của biến trong chương trình và hàm

I. PHẠM VI CỦA BIẾN KHAI BÁO TRONG HÀM

- Trong Python tất cả các biến khai báo bên trong hàm đều có tính địa phương (cục bộ), không có hiệu lực ở bên ngoài hàm.
- Ví dụ:.....

.....

.....

.....

.....

II. PHẠM VI CỦA BIẾN KHAI BÁO NGOÀI HÀM

- Biến khai báo bên ngoài hàm không có tác dụng bên trong hàm.
- Bên trong hàm có thể truy cập để sử dụng giá trị của biến đã khai báo trước đó ở bên ngoài hàm.
- **Lưu ý:** nếu muốn biến bên ngoài vẫn có tác dụng bên trong hàm thì chỉ cần khai báo lại biến này bên trong hàm với từ khóa **global**.
- Ví dụ:.....

.....

.....

.....

.....

.....

.....

III. THỰC HÀNH

Bài 1: Giả sử có các lệnh sau:

```
>>> a, b - 1, 2
```

```
>>> def f(a, b):
    a = a+b
    b = b*a
    return a + b
```

Giá trị của a, b bằng bao nhiêu sau khi thực hiện lệnh sau?

a) f(1, 2) b) f(10, 20)

.....

.....

.....

Bài 2: Ta có thể khai báo một biến bên trong hàm trùng tên với biến đã khai báo trước đó bên ngoài hàm không?

.....

.....

.....

Bài 3: Giả sử hàm f(x, y) được định nghĩa như sau:

```
>>> def f (x, y) :
    a = 2* (x + y)
    print(a + n)
```

Kết quả nào được in ra khi thực hiện các lệnh sau?

n = 10

f(1, 2)

.....

.....

.....

BÀI 29: NHẬN BIẾT LỖI CHƯƠNG TRÌNH (1 LT + 1 TH)

Sau bài này em sẽ:

- Biết và phân biệt được một số loại lỗi chương trình
- Biết được một vài lỗi ngoại lệ thường gặp

I. NHẬN BIẾT LỖI CHƯƠNG TRÌNH

✂ Trường hợp 1:

SyntaxError: Invallid syntax

Người lập trình viết sai cú pháp lệnh, chương trình lập tức dừng lại và thông báo lỗi cú pháp

Ví dụ: `>>> While true print("Hello")`

✂ Trường hợp 2:

Traceback (most recent call last):

File "<pyshell#0>", line 1, in <module>

Người dùng nhập dữ liệu sai, chương trình thông báo lỗi nhập dữ liệu không đúng khuôn dạng

Ví dụ: `>>> n = int( input("Nhập số nguyên n: "))`

Nhập số nguyên n: 1.5

✂ Trường hợp 3:

Traceback (most recent call last):

File "C:\Python\Array_b1.py", line 3, in <module>

`print (A[1],end")`

IndexError: list index out of range

Chương trình thông báo lỗi chỉ số vượt quá giới hạn cho phép

Ví dụ:

`A = [1, 3, 10, 0]`

`for i in range(5):`

`print(A[i], end = " ")`

Khi chạy chương trình sẽ báo lỗi.

✂ Trường hợp 4:

Chương trình thực hiện bình thường nhưng kết quả không đúng với yêu cầu của bài toán. Đây là lỗi logic bên trong chương trình.

Ví dụ: # Tính tổng của ba số nguyên dương đầu tiên

`>>> s = 0`

`>>> for i in range(3):`

`s = s + i`

`>>> print(s)`

Chương trình cho kết quả là 3 mà kết quả đúng là $1 + 2 + 3 = 6$. Lí do là hàm `range(3)` trả lại vùng giá trị là 0, 1, 2 chứ không phải là 1, 2, 3

II. MỘT SỐ LỖI NGOẠI LỆ THƯỜNG GẶP:

Mã lỗi ngoại lệ	Mô tả lỗi
ZeroDivisionError	Lỗi này xảy ra khi lệnh thực hiện phép chia cho giá trị 0.
IndexError	Lỗi xảy ra khi lệnh cố gắng truy cập phần tử của danh sách nhưng chỉ số vượt quá giới hạn.
NameError	Lỗi xảy ra khi chương trình muốn tìm một tên nhưng không thấy. Ví dụ khi lệnh gọi một hàm nhưng không có hàm đó.
TypeError	Lỗi kiểu dữ liệu. Một số ví dụ lỗi loại này: - Lệnh truy cập một phần tử của danh sách nhưng chỉ số không là số nguyên - Lệnh tính biểu thức số nhưng lại có một toán hạng không phải là số
ValueError	Lỗi liên quan đến giá trị của đối tượng. Lỗi khi thực hiện lệnh chuyển đổi kiểu dữ liệu, đổi số của hàm có giá trị mà hàm không hỗ trợ. Ví dụ khi thực hiện lệnh <code>int("1.55")</code> sẽ sinh lỗi loại này.
IndentationError	Lỗi khi các dòng lệnh thụt vào không thẳng hàng hoặc không đúng vị trí
SyntaxError	Lỗi cú pháp.

CÂU HỎI GỢI Ý ÔN TẬP KTCK HK2 - TIN HỌC 10

Câu 1: Trong python, đâu là lệnh xóa phần tử đầu tiên trong danh sách A?

- A. `del (A[0])` B. `delete (A[0])` C. `A.clear()` D. `del (A[len(A)])`

Câu 2: Đoạn chương trình cho kết quả gì?

```
a = [1,2,3,4,5]
a.append(9)

del(a[3])

print(a)
```

A. [1,2,3,5,9]

C. [1,2,3,4,9]

B. [1,2,4,5,9]

D. [1,2,3,4,5]

Câu 3: Đoạn chương trình cho kết quả gì?

```
a = [1,2,3,4,5]
a.append("lop 10")
print(a)
```

A. [1,2,3,4,5, 'lop 10']

C. [1,2,3,4,5]+'lop 10']

B. [lop 10,1,2,3,4,5]

D. [1,2,3,lop 10,4,5]

Câu 4: Cho biết kết quả của đoạn lệnh sau:

```
>>> A = [1,2,3,4,5]
>>> for i in range(len(A)):
```

```
    print(A[i], end = " ")
```

A. 1,2,3,4,5

B. 1" 2" 3" 4" 5

C. lỗi

D. 1 2 3 4 5

Câu 5: Em hãy cho biết kết quả sau khi thực hiện các lệnh sau:

```
m = 2
while m < 10:
    print(m,end= " ")
    m = m + 2
```

A. 0 2 4 6 8 10

B. 3 5 7 9

C. 2 4 6 8 10

D. 2 4 6 8

Câu 6: Lệnh sau trả lại giá trị gì?

```
>> "abcdabcd".find("cd")
```

```
>> "abcdabcd".find("cd", 4)
```

A. 2,6

B. 2,2

C. 3, 3

D. 2,7

Câu 7: Trong Python, để khai báo biến x là kiểu xâu, ta khai báo như thế nào là đúng nhất:

A. x=string(input('mời nhập'))

B. x=float(input('mời nhập'))

C. x=input('mời nhập')

D. x=int(input('mời nhập'))

Câu 8: Giá trị của a sau khi kết thúc vòng lặp là bao nhiêu:

```
a=0
for i in range(10,14): a=a+6
```

A. 24

B. 18

C. 26

D. 12

Câu 9: Cho biết kết quả của đoạn lệnh sau:

```
>>> A = [1, 0, 'One', 9, 15, 'Two', True, False]
```

>>> A[3] * 5

A. lỗi

B. 20

C. 0

D. 45

Câu 10: Danh sách (list) là gì?

A. Là tập hợp các phần tử cùng kiểu dữ liệu

B. Là tập hợp các phần tử có thể có nhiều kiểu dữ liệu khác nhau

C. Là tập hợp chứa 1 phần tử đơn chiếc

D. Là một kiểu dữ liệu đặc biệt trong Python, các phần tử trong list đều là những số nguyên.

Câu 11: Với:

S1 = "12345"

S2 = "3e4r5"

S3 = "45"

Cho biết giá trị của hai biểu thức sau:

S3 in S1

S3 in S2

A. True, True.

B. True, False.

C. False, False.

D. False, True.

Câu 12: Cho list a như sau:

a = [1,2,3,4,5, "tin", "python"]

Lệnh len(a) cho giá trị là bao nhiêu?

A. 7

B. 6

C. 8

D. lỗi

Câu 13: Khi nào vòng lặp while sẽ bị lặp vô hạn?

A. Khi điều kiện trong vòng lặp luôn nhận giá trị True và không thay đổi

B. Khi điều kiện trong vòng lặp sai

C. Khi điều kiện trong vòng lặp là biểu thức tính toán

D. Khi điều kiện trong vòng lặp luôn luôn nhận giá trị False

Câu 14: Cho đoạn chương trình Python sau:

```
lopHoc = ["ban", "ghe", "hoc sinh", "giao vien"]
```

```
lopHoc.append("bang")
```

```
print(lopHoc)
```

Kết quả chương trình được in ra màn hình là:

- A. ['ban', 'ghe', 'hoc sinh', 'giao vien']
- B. lopHoc
- C. ['ban', 'ghe', 'hoc sinh', 'giao vien', 'bang']
- D. bang

Câu 15: Chọn cú pháp đúng của lệnh while trong Python:

- A. while <điều kiện>;
 <khối lệnh>
- B. while <điều kiện> do<câu lệnh>
- C. while <điều kiện>:
 <khối lệnh>
- D. while <khối lệnh>:
 <điều kiện>

Câu 16: Cho biết độ dài của xâu s = "Ket noi tri thuc"

- A. 14
- B. 16
- C. 17
- D. 15

Câu 17: Kết quả của lệnh sau:

`"Tin học"* 3`

cho ra kết quả gì?

- A. Lỗi
- B. Tin học * 3
- C. "Tin học" * 3
- D. Tin họcTin họcTin học

Câu 18: Hàm range(100) sẽ tạo ra:

- A. Một dãy số ngẫu nhiên 100
- B. 100 số ngẫu nhiên
- C. Một dãy số nguyên từ 1 đến 99
- D. Một dãy số nguyên từ 0 đến 99

Câu 19: Trong Python, có thể dùng vòng lặp nào để duyệt từng kí tự của xâu?

- A. Loop
- B. for hoặc while
- C. for
- D. while

Câu 20: Xét đoạn chương trình:

`a = [1,2,3,4,5]`

`b = ['a','b','c','d','e']`

`c = a + b`

giá trị của c là:

A. [1,'a',2,'b',3,'c',4,'d',5,'e']

B. [1,2,3,4,5][['a','b','c','d','e']]

C. [1,2,3,4,5,'a','b','c','d','e']

D. (1,2,3,4,5,'a','b','c','d','e')

Câu 21: Cú pháp lệnh thêm phần tử vào cuối danh sách là:

A. <danh sách> . append()

B. <danh sách> : append()

C. <danh sách> = append()

D. <danh sách> append()

Câu 22: Để tạo một danh sách trống, cách viết bất kỳ sau đây là đúng?

A. a = [" "]

B. một = [0]

C. a = []

D. a = [rỗng]

Câu 23: Xét bài toán tính tổng: $S = 1 + 2 + 3 + 4 + \dots + n + \dots$ cho đến khi $S \geq 10000$.

Điều kiện nào sau đây cho vòng lặp while là đúng:

A. while $S \leq 10000$

B. while $S \neq 10000$

C. while $S < 10000$

D. while $S \geq 10000$

Câu 24: Cho đoạn chương trình sau:

x=5

y=4

if x>y : f=x*x-y

else: f=3*x-y

Sau khi thực hiện đoạn chương trình, giá trị của f là:

A. 22

B. 2

C. 21

D. 12

Câu 25: Xét đoạn lệnh sau:

a=39

b=a//10

c=a%10

d=b*10+c

print(d)

Giá trị nào được xuất ra sau khi thực hiện đoạn lệnh trên?

A. 31

B. 39

C. 51

D. 9

Câu 26: Với cấu trúc rẽ nhánh

if <điều kiện>:

<câu lệnh>

thì câu lệnh được thực hiện khi?

A. Điều kiện được tính toán xong

B. Điều kiện không tính được

C. Điều kiện có giá trị đúng

D. Điều kiện có giá trị sai

Câu 27: Cho chuỗi s="2.4.1.6.7" để tách chuỗi s thành danh sách a=['2','4','1','6','7']

ta dùng lệnh nào dưới đây?

A. a=split(".")

B. a=split()

C. a=s.split()

D. a=s.split(".")

Câu 28: Trong python, Để kiểm tra 3 cạnh tạo thành tam giác hay không thì câu lệnh nào sau đây đúng?

- A. `if (a+b>c) or( a+c>b or ( b+c>a) :write('tạo thành tam giác')`
- B. `if (a+b>c) or( a+c>b or ( b+c>a) :print('tạo thành tam giác')`
- C. `if (a+b>c) and( a+c>b and ( b+c>a): write('tạo thành tam giác'`
- D. `if (a+b>c) and( a+c>b and ( b+c>a): print('tạo thành tam giác')`

Câu 29: Điểm khác nhau giữa chuỗi ký tự và danh sách là:

- A. không thể thay đổi được từng ký tự của chuỗi
- B. có thể thay đổi được từng ký tự của chuỗi
- C. truy cập đến phần tử của chuỗi thông qua tên biến chuỗi và chỉ số.
- D. các phần tử của chuỗi được đánh số bắt đầu từ 0.

Câu 30: Cho `s1 = "abc"`, `s2 = "ababcabca"`. Biểu thức sau đây nhận giá trị False?

- A. `s2 in s2`
- B. `s1 in s2`
- C. `"abc123" in s2`
- D. `"abcbca" in s2`

Câu 31: Cho chương trình

```
a = 4
b = 5
if a > b:
    a = a+b
```

Sau khi thực hiện chương trình trên thì giá trị của a và b là

- A. a = 5; b = 4
- B. a = 4; b = 5
- C. a = 9; b = 9
- D. a = 9; b = 5

Câu 32: Sử dụng phương thức (hàm) nào để tìm vị trí của một chuỗi con trong chuỗi khác không?

- A. `test()`
- B. `find()`
- C. `in()`
- D. `split()`

Câu 33: Hàm `range(4,10)` sẽ tạo ra:

- A. một dãy số nguyên từ 0 đến 10
- B. một dãy số nguyên từ 4 đến 10
- C. một dãy số nguyên từ 4 đến 11
- D. một dãy số nguyên từ 4 đến 9

Câu 34: Chọn đáp án sai?

- A. Kết quả của hàm `join()` là một chuỗi
- B. Toán tử `in` dùng để kiểm tra một chuỗi có phải là chuỗi con của chuỗi khác hay không
- C. Kết quả của hàm `split()` là một danh sách
- D. Khi dùng hàm `find()` để tìm kiếm, Chương trình sẽ báo lỗi nếu không tìm thấy chuỗi con trong chuỗi mẹ

Câu 35: Cho list a là: `a = [1,2,3,4,5]`

`a[1]` có giá trị là:

- A. 3
- B. 2
- C. 4
- D. 1

Câu 36: kết quả của biểu thức sau: $111//5 \neq 20$ or $20\%3 \neq 0$

- A. true B. False C. True D. false

Câu 37: Thực hiện đoạn lệnh sau:

`S="Hà Nội - Việt Nam"`

`i=S.find("Việt Nam")`

Giá trị của i là:

- A. 6 B. 9 C. 7 D. 8

Câu 38: Câu lệnh khởi tạo list nào sau đây đúng cú pháp

A. `a = [1, 2, 3, 4, 5]` B. `<a> = [1, 2, 3, 4, 5]`

C. `a = [1; 2; 3; 4; 5]` D. `a = (1, 2, 3, 4, 5)`

Câu 39: Vòng lặp for kết thúc khi nào?

A. Khi một điều kiện cho trước được thỏa mãn.

B. Khi tìm được Output

C. Khi điều kiện sai

D. Khi đủ số vòng lặp

Câu 40: List trong Python được viết với dấu:

- A. `[]` B. `()` C. `{}` D. `'''`

----- HẾT -----

ĐÁP ÁN:

1	A
2	A
3	A
4	D
5	D
6	A
7	C
8	A
9	D
10	B
11	B

12	A
13	A
14	C
15	C
16	B
17	D
18	D
19	B
20	C
21	A
22	C

23	C
24	C
25	B
26	C
27	D
28	D
29	A
30	C
31	B
32	B
33	D

34	D
35	B
36	C
37	B
38	A
39	D
40	A

