

TÀI LIỆU HƯỚNG DẪN THÍ THỰC HÀNH, THÍ NGHIỆM HỌC PHẦN
KỸ THUẬT LẬP TRÌNH
MÃ HỌC PHẦN: ELE20004

BÀI 1. Thực hiện các thuật toán bằng phần mềm Flowgorithm

1.1. Mục đích

- Sử dụng phần mềm Flowgorithm trong thiết kế và biểu diễn thuật toán;
- Xây dựng thuật toán cho các bài toán cụ thể trên Flowgorithm

1.2. Cơ sở lý thuyết

1.2.1. Giới thiệu Flowgorithm

Flowgorithm là ngôn ngữ lập trình dành cho người mới bắt đầu miễn phí dựa trên sơ đồ đồ họa đơn giản.

Thông thường, khi một sinh viên lần đầu tiên học lập trình, họ thường sử dụng một trong những ngôn ngữ lập trình dựa trên văn bản. Tùy thuộc vào ngôn ngữ, điều này có thể dễ dàng hoặc khó khăn. Nhiều ngôn ngữ yêu cầu bạn viết các dòng mã khó hiểu chỉ để hiển thị văn bản "Xin chào, thế giới!".

Bằng cách sử dụng sơ đồ, bạn có thể tập trung vào các khái niệm lập trình hơn là tất cả các sắc thái của ngôn ngữ lập trình thông thường. Bạn cũng có thể chạy các chương trình của mình trực tiếp trong Flowgorithm.

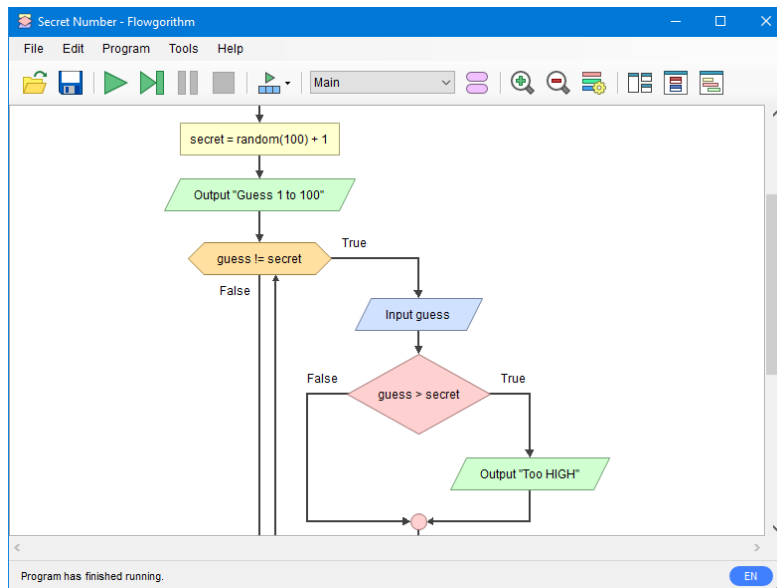
Một khi bạn hiểu logic lập trình, bạn sẽ dễ dàng học một trong những ngôn ngữ chính. Flowgorithm có thể tương tác chuyển đổi sơ đồ của bạn sang hơn 18 ngôn ngữ. Chúng bao gồm: C #, C ++, Java, JavaScript, Lua, Perl, Python, Ruby, Swift, Visual Basic .NET và VBA (được sử dụng trong Office).

1.2.2. Đặc tính của Flowgorithm

- Đầu ra dễ hiểu
- Cửa sổ xem biến đồ họa
- Tương tác tạo mã thật
- Phối màu tùy chỉnh
- Hỗ trợ đa ngôn ngữ
- Hỗ trợ đầy đủ các toán tử và hơn 20 hàm cơ bản

1.2.3. Các cửa sổ chính khi làm việc với Flowgorithm

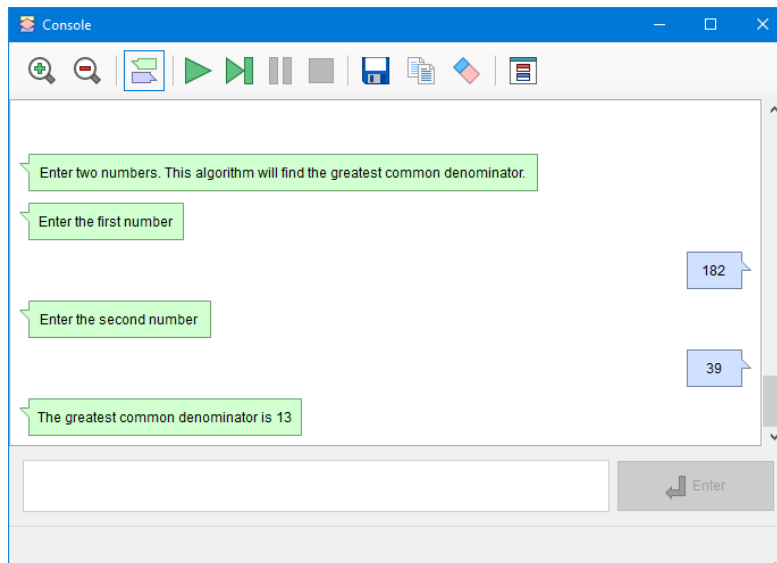
- a. Giao diện chính*



Hình 1. Main window

b. Cửa sổ điều khiển

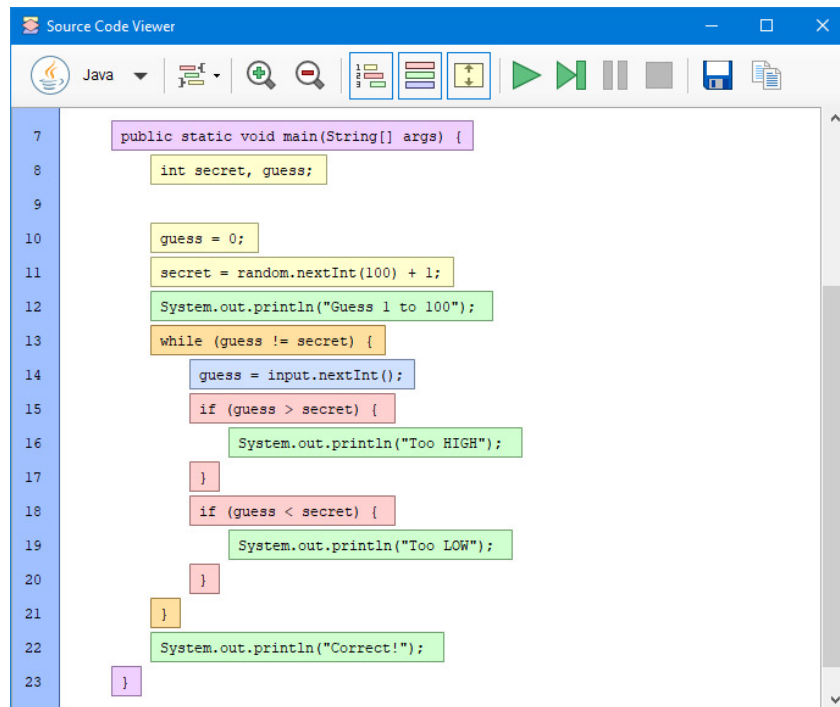
Phương pháp cổ điển để tương tác với máy tính là sử dụng "Bảng điều khiển". Thuật toán lưu lượng cố gắng làm cho nó trông giống như một cửa sổ nhắn tin tức thời điển hình. "Bong bóng trò chuyện" được mã hóa màu để khớp với hình dạng Đầu vào và Đầu ra được sử dụng trong sơ đồ. Nếu bạn không muốn sử dụng bong bóng trò chuyện, bạn cũng có thể chuyển đổi giữa chúng và văn bản thuần túy cổ điển.



Hình 2. Console Window

c. Giao diện mã nguồn

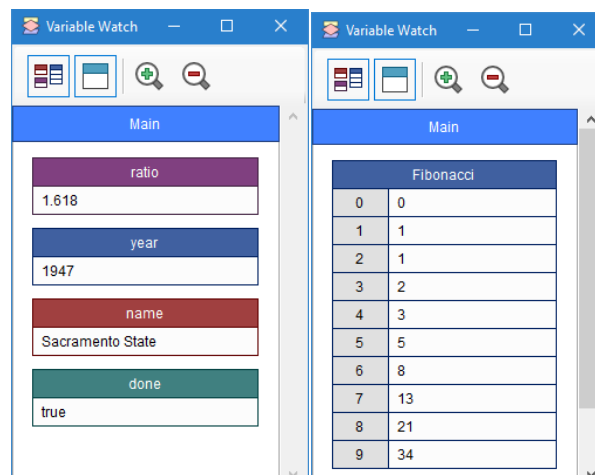
Trình xem mã nguồn có thể chuyển đổi sơ đồ của bạn sang một số ngôn ngữ lập trình chính.



Hình 3. Source Code Viewer Window

d. Giao diện xem các biến của chương trình

Cửa sổ theo dõi biến được sử dụng để theo dõi cách các biến của bạn thay đổi khi chương trình của bạn thực thi. Mỗi biến được mã hóa màu dựa trên kiểu dữ liệu của nó. Nhìn thoáng qua, bạn có thể cho biết chính xác loại dữ liệu nào đang được lưu trữ - và nắm bắt nơi bạn có thể muốn sử dụng một loại dữ liệu khác.



Hình 4. Variable Watch Window

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao:

1.3.1. Thiết bị thực hành thí nghiệm

Máy tính cài phần mềm Flowgorithm

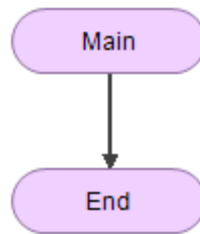
1.3.2. Vật tư tiêu hao: Không

1.4. CÁC BƯỚC TIẾN HÀNH THÍ NGHIỆM

1.4.1. Xây dựng thuật toán hiển thị chuỗi ký tự với Flowgorithm

B1: Tạo chương trình mới

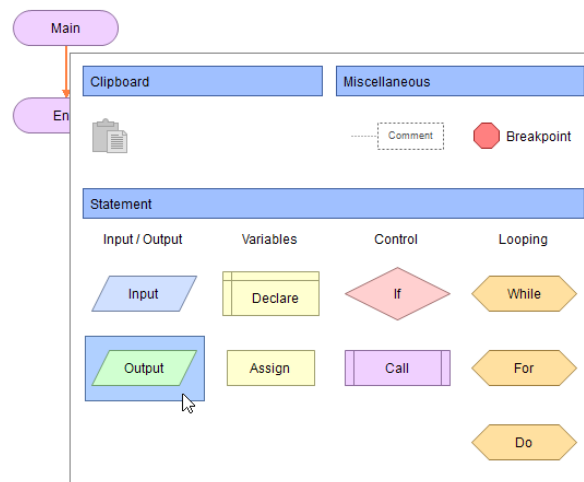
- Khi bắt đầu một sơ đồ mới, chúng ta sẽ thấy hai hình chữ nhật tròn được gọi là "terminals". Những biểu tượng này đại diện cho sự bắt đầu và kết thúc chương trình của bạn.
- Nhiều sơ đồ thuật toán hiển thị văn bản "Begin" trong terminal. Flowgorithm sử dụng văn bản "Main". Hầu hết các ngôn ngữ lập trình bắt đầu với các nỗ lực "Main" và Flowgorithm cũng vậy.



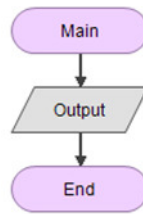
Hình 5. Bắt đầu và kết thúc chương trình

B2: Thêm các khối chức năng

- Tất cả mọi thứ trong một sơ đồ được thể hiện bằng các khối hình. Các khối được thêm vào giữa các terminals Main và End.
- Để thêm hình dạng, di chuyển con trỏ chuột của bạn trên một dòng. Nếu có thể thêm một khối, dòng sẽ chuyển sang màu cam.
- Bấm đúp hoặc bấm chuột phải để thêm hình.
- Hiển thị giao diện với các khối cần thêm
- Lựa chọn và click vào khối cần thêm sẽ được chương trình như hình 7.



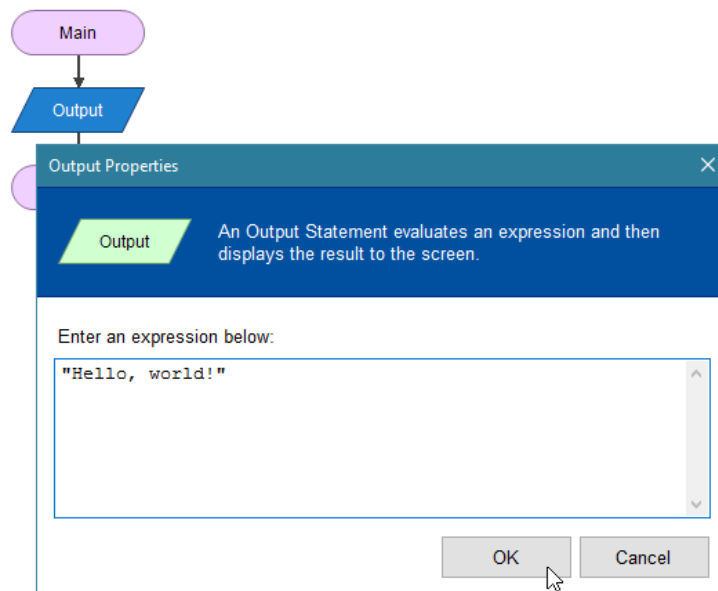
Hình 6. Thêm các khối chức năng cho chương trình



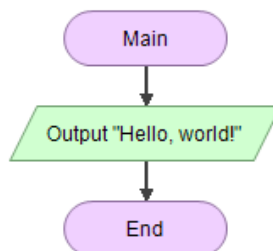
Hình 7. Chương trình đã thêm các khối chức năng

B3: Định nghĩa chức năng cho khối

- Click đúp chuột vào khối cần định nghĩa chức năng
- Cửa sổ “Output Properties” xuất hiện, chúng ta có thể nhập vào các chứng năng theo cú pháp quy định, chương trình sau khi định nghĩa chức năng được chỉ ra trong hình 9.



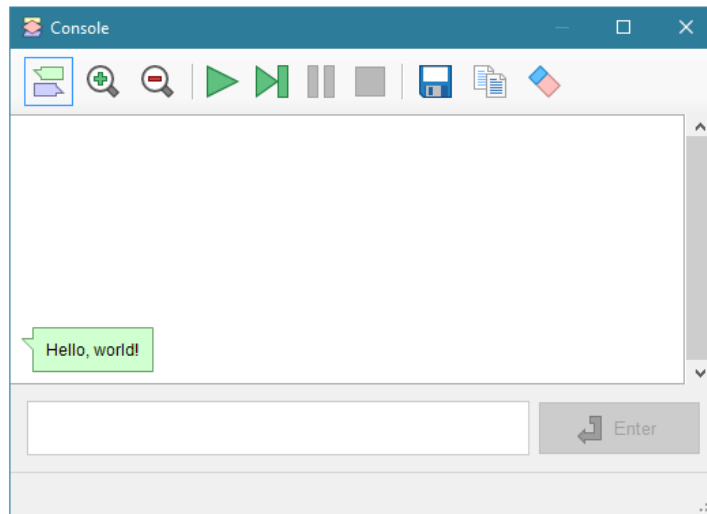
Hình 8. Định nghĩa chức năng cho khối được thêm vào



Hình 9. Chương trình sau khi định nghĩa cho các khối chức năng thêm vào

B4: Khởi chạy chương trình đã thiết kế

- Bấm F5 hoặc nút “Run” trên menu của chương trình

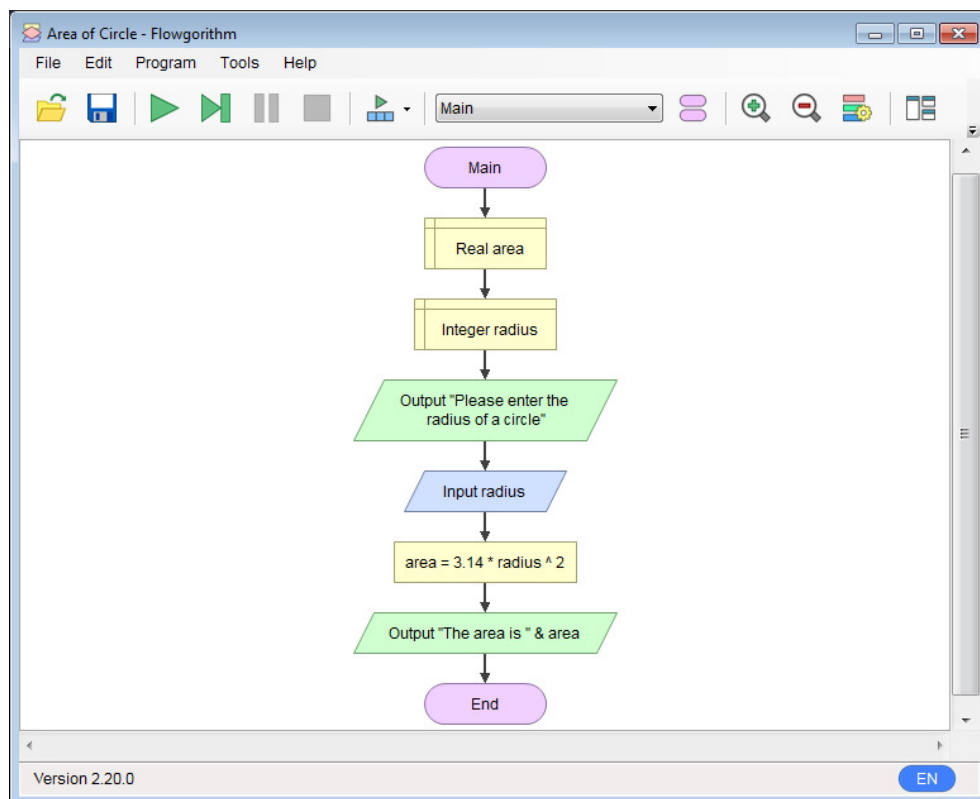


Hình 10. Cửa sổ điều khiển hiển thị kết quả

B5: Xem mã nguồn hoặc các biến sử dụng trong chương trình sử dụng menu “Tools” ở thanh công cụ.

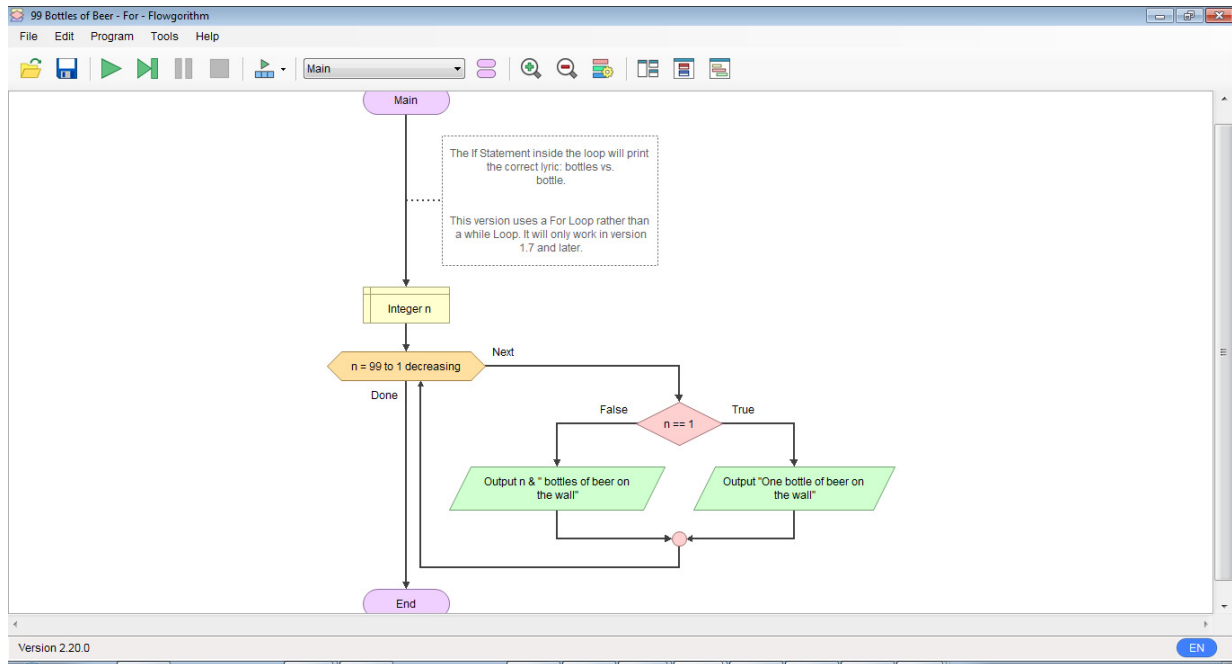
1.4.2. Sử dụng Flowgorithm xây dựng chương trình giải quyết các bài toán

a. Tính diện tích hình tròn



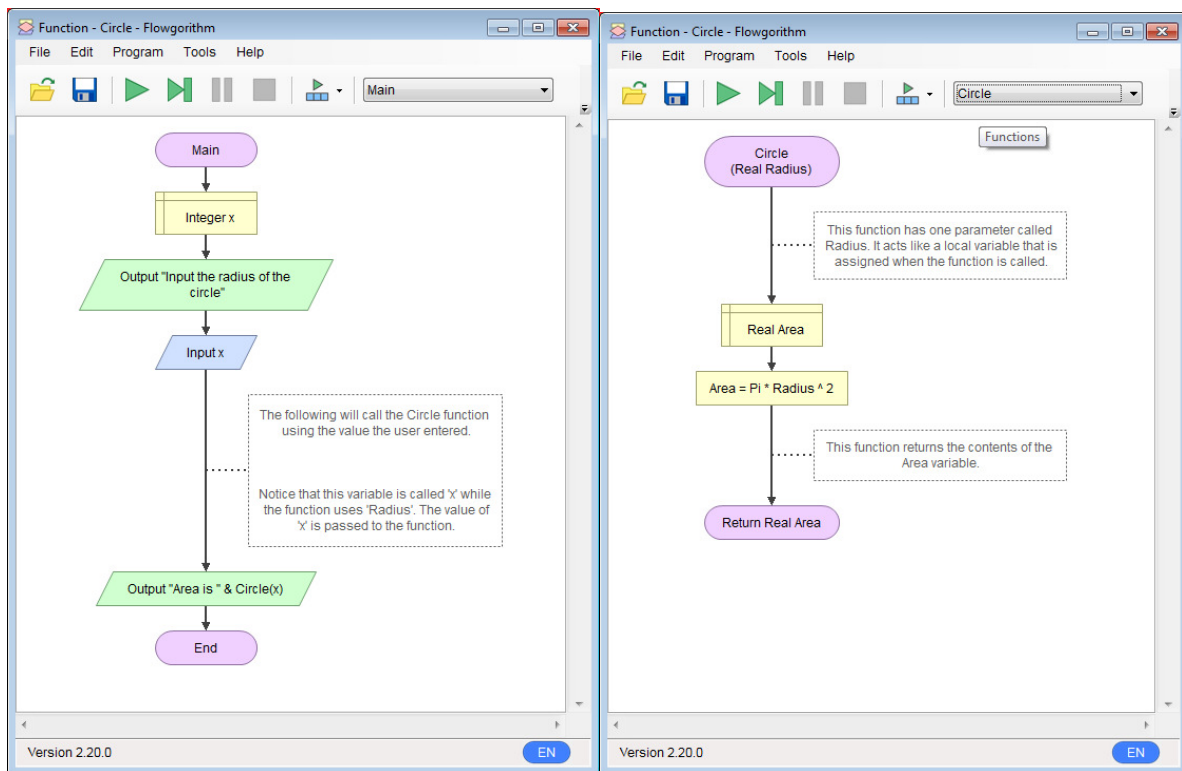
Hình 11. Sơ đồ giải thuật bài toán tính diện tích hình tròn

b. Nhập và in số lượng chai sử dụng vòng lặp for



Hình 12. Giải thuật bài toán nhập và in số lượng chai sử dụng vòng lặp while

c. Tính diện tích hình tròn sử dụng hàm (chương trình con)



Hình 13. Sử dụng hàm cho giải thuật tính diện tích hình tròn

- d. *Viết chương trình nhập và cạnh và chiều cao tương ứng của một tam giác và in ra màn hình diện tích tam giác.*
- e. *Xây dựng thuật toán kiểm tra tính nguyên tố (một số nguyên dương n là số nguyên tố khi chỉ có hai ước là 1 và chính nó)*

Gợi ý:

Bước 1: nhập số nguyên dương n ;

Bước 2: nếu $n=1$ thông báo “ n không là số nguyên tố”, kết thúc;

Bước 3: nếu $n < 4$ thông báo “ n là số nguyên tố”, kết thúc;

Bước 4: khai báo biến i và gán $i=2$;

Bước 5: nếu i là ước của n thì đến bước 7;

Bước 6: $i=i+1$ (tăng i lên 1 đơn vị) rồi quay lại bước 5;

Bước 7: nếu $i=n$ thì thông báo “ n là số nguyên tố”, ngược lại thông báo “ n không là số nguyên tố”. Kết thúc;

1.5. Câu hỏi kiểm tra

- Ý nghĩa các khối sử dụng trong chương trình
- Cách khai báo nhập dữ liệu cho các biến, các khối chức năng và vòng lặp
- Xây dựng hàm thực hiện chương trình con trong Flowgorithm

1.6. Tài liệu tham khảo

<http://www.flowgorithm.org/documentation/index.htm>

Bài 2. Các cú pháp, kiểu dữ liệu, lệnh điều khiển trong lập trình Python

1.1. Mục đích

Giúp sinh viên nắm bắt các kiến thức cơ bản như kiểu dữ liệu, các kiểu cú pháp, các lệnh điều khiển khi viết một chương trình bằng ngôn ngữ python.

1.2. Cơ sở lý thuyết

Xem các khái niệm: Định danh, các từ khóa, dòng lệnh, khối lệnh, trích dẫn, comment, biến, toán tử, cấu trúc điều khiển trong python.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

- Viết đoạn chương trình sau và sửa lỗi

```
n1=int(input("enter n1 value"))
n2=int(input("enter n2 value"))
if n1>n2:
    print("n1 is big")
else:
    print("n2 is big")
```

- Viết chương trình nhập hai điểm và tính khoảng cách

```
import math;
x1=int(input("Enter x1--->"))
y1=int(input("Enter y1--->"))

x2=int(input("Enter x2--->"))
y2=int(input("Enter y2--->"))

d1 = (x2 - x1) * (x2 - x1);
d2 = (y2 - y1) * (y2 - y1);
res = math.sqrt(d1+d2)
print ("Distance between two points:",res);
```

- Viết chương trình nhập vào một số và kiểm tra số đó là chẵn hay lẻ, in thông báo ra màn hình

```
n=int(input("Enter a number---->"))
if n % 2 == 0:
    print ("EVEN Number");
else:
    print ("ODD Number");
```

- Viết chương trình in ra màn hình số nghịch đảo và kết quả dưới dạng thập phân của một dãy số tự nhiên trong khoảng (a,b)

```
i=1;
for j in range(2,10):
    print("i:",i,"j:",j)
    print(i,"/",j)
    print (i/j);
```

5. Viết chương trình nhập vào một số tự nhiên $n > 0$, in ra màn hình các số tự nhiên giảm dần từ n đến 0, mỗi ký tự in trên 1 hàng

```
n = int(input("Enter A Number--->"));
while n >=0:
    print (n);
    n = n - 1;
```

6. Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 2000 và 3200 (tính cả 2000 và 3200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

Gợi ý:

- Sử dụng range(begin, end)

Code mẫu:

```
j=[]
for i in range(2000, 3201):
    if (i%7==0) and (i%5!=0):
        j.append(str(i))
print (','.join(j))
```

7. Với số nguyên n nhất định, hãy viết chương trình để tạo ra một dictionary chứa $(i, i*i)$ như là số nguyên từ 1 đến n (bao gồm cả 1 và n) sau đó in ra dictionary này. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}.

Gợi ý:

- Viết lệnh yêu cầu nhập số nguyên n .

Code mẫu:

```
n=int(input("Nhập vào một số:"))
d=dict()
for i in range(1,n+1):
    d[i]=i*i

print (d)
```

8. Viết chương trình in ra màn hình dãy số Fibonacci nhỏ hơn 4.000.000, tìm tổng các số chẵn trong dãy đã in

```

a, b = 1, 2
total = 0
print(a,end=" ")
while (a <=4000000-1):
    if a % 2 == 0:
        total += a
        a, b = b, a+b
        print(a,end=" ")
print("\n sum of prime numbers term in fibonacci series: ",total)

```

9. Viết chương trình đếm số ký tự trong 1 xâu ký tự nhập vào từ bàn phím, lưu các ký tự vào cấu trúc từ điển

```

str=input("Enter a String:")
dict = {}
for n in str:
    keys = dict.keys()
    if n in keys:
        dict[n] += 1
    else:
        dict[n] = 1
print (dict)

```

Hoặc

```

str=input("Enter a String")
dict = {}
for i in str:
    dict[i] = str.count(i)
print (dict)

```

10. Viết chương trình sử dụng các phương thức split và join để tách nhập xâu ký tự

```

a="hi i am python programmer"
b=a.split()
print (b)
c=" ".join(b)
print (c)

```

11. Viết chương trình kết nối các danh sách vào từ điển

```

l=[1, 'python', 4, 7]
k=['cse', 2, 'guntur', 8]
m=[]
m.append(l);
m.append(k);
print (m)
d={1:l, 2:k, 'combine_list':m}
print (d)

```

12. Một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký.
Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.

Các tiêu chí kiểm tra mật khẩu bao gồm:

- Ít nhất 1 chữ cái nằm trong [a-z]
- Ít nhất 1 số nằm trong [0-9]
- Ít nhất 1 ký tự nằm trong [A-Z]

- Ít nhất 1 ký tự nằm trong [\$ # @]
- Độ dài mật khẩu tối thiểu: 6
- Độ dài mật khẩu tối đa: 12

Chương trình phải chấp nhận một chuỗi mật khẩu phân tách nhau bởi dấu phẩy và kiểm tra xem chúng có đáp ứng những tiêu chí trên hay không. Mật khẩu hợp lệ sẽ được in, mỗi mật khẩu cách nhau bởi dấu phẩy.

Ví dụ mật khẩu nhập vào chương trình là: ABd1234@1,a F1#,2w3E*,2We3345

Thì đầu ra sẽ là: ABd1234@1

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

```
import re
value = []
items=[x for x in input("Nhập mật khẩu: ").split(',')]
# #####
for p in items:
    if len(p)<6 or len(p)>12:
        continue
    else:
        pass
    if not re.search("[a-z]",p):
        continue
    elif not re.search("[0-9]",p):
        continue
    elif not re.search("[A-Z]",p):
        continue
    elif not re.search("[$#@]",p):
        continue
    elif re.search("\s",p):
        continue
    else:
        pass
    value.append(p)
print (",".join(value))
```

13. Viết chương trình giải phương trình bậc 2: $ax^2+bx+c=0$, với các hệ số a, b, c nhập từ bàn phím

1.5. Câu hỏi kiểm tra

- Các kiểu biến, khai báo và đặt tên biến trong python;
- Cấu trúc điều khiển trong python
- Giải thích các cú pháp lệnh, và các phương thức sử dụng trong các chương trình python đã viết

1.6. Tài liệu tham khảo

Bài 3. Lập trình hàm trong Python

1.1. Mục đích

Giúp sinh viên nắm bắt các kiến thức trong lập trình python sử dụng hàm.

1.2. Cơ sở lý thuyết

Xem các quy tắc khai báo hàm, gọi hàm, giá trị trả về, tham số truyền vào, tham số mặc định, phạm vi của biến trong python, sử dụng các hàm có sẵn trong các thư viện của python.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

- Viết hàm sum() tính tổng hai số

```
def sum(a, b):  
    print("sum = " + str(a + b))  
# tính tổng 2 số 4,5  
sum(4, 5)  
# tính tổng 2 số 3,7  
sum(3, 7)
```

- Viết hàm sum() với kết quả trả về

```
def sum(a, b):  
    return a + b  
c = sum(4, 5);  
print("Tổng của 4 và 5 = " + str(c))
```

- Tìm và sửa lỗi chương trình

```
def say_hello():  
    a = "Hello"  
    print(a)  
print(a)
```

- Viết chương trình có phạm vi biến như sau

```
a = "Hello Guy!"  
def say(a):  
    a = "Vinh University"  
    print(a)  
say(a)
```

```
print(a)
```

5. Viết chương trình sau và xem sự thay đổi của biến

```
a = "Hello Guy!"
```

```
def say():
```

```
    global a
```

```
    a = "Vinh University "
```

```
    print(a)
```

```
say()
```

```
print(a)
```

6. Viết chương trình sau và giải thích việc truyền tham số của hàm

```
def get_sum(*num):
```

```
    tmp = 0
```

```
    # duyệt các tham số
```

```
    for i in num:
```

```
        tmp += i
```

```
    return tmp
```

```
result = get_sum(1, 2, 3, 4, 5)
```

```
print(result)
```

7. Định nghĩa hàm có thể chấp nhận input là số nguyên và in "Đây là một số chẵn" nếu nó chẵn và in "Đây là một số lẻ" nếu là số lẻ.

Gợi ý:

Sử dụng toán tử % để kiểm tra xem số đó chẵn hay lẻ.

Code mẫu:

```
def checkValue(n):
```

```
    if n%2 == 0:
```

```
        print ("Đây là một số chẵn")
```

```
    else:
```

```
        print ("Đây là một số lẻ")
```

```
checkValue(7)
```

8. Một Robot di chuyển trong mặt phẳng bắt đầu từ điểm đầu tiên (0,0). Robot có thể di chuyển theo hướng UP, DOWN, LEFT và RIGHT với những bước nhất định. Dấu di chuyển của robot được đánh hiển thị như sau:

UP 5

DOWN 3

LEFT 3

RIGHT 3

Các con số sau phía sau hướng di chuyển chính là số bước đi. Hãy viết chương trình để tính toán khoảng cách từ vị trí hiện tại đến vị trí đầu tiên, sau khi robot đã di chuyển một quãng đường. Nếu khoảng cách là một số thập phân chỉ cần in số nguyên gần nhất.

Ví dụ: Nếu tuple sau đây là input của chương trình:

UP 5
DOWN 3
LEFT 3
RIGHT 2

thì đầu ra sẽ là 2.

Gợi ý:

Trong trường hợp dữ liệu đầu vào được nhập vào chương trình nó nên được giả định là dữ liệu được người dùng nhập vào từ giao diện điều khiển.

Code mẫu:

```
import math
pos = [0,0]
while True:
    s = input()
    if not s:
        break
    movement = s.split(" ")
    direction = movement[0]
    steps = int(movement[1])
    if direction=="UP":
        pos[0]+=steps
    elif direction=="DOWN":
        pos[0]-=steps
    elif direction=="LEFT":
        pos[1]-=steps
    elif direction=="RIGHT":
        pos[1]+=steps
    else:
        pass
#####
print (int(round(math.sqrt(pos[1]**2+pos[0]**2))))
```

9. Chương trình máy tính thực hiện các phép tính đơn giản

```
“""" Program make a simple calculator that can add, subtract, multiply and
divide using functions"""”
# This function adds two numbers
def add(x, y):
    return x + y
# This function subtracts two numbers
def subtract(x, y):
    return x - y
# This function multiplies two numbers
def multiply(x, y):
```



```

    return x * y
# This function divides two numbers
def divide(x, y):
    return x / y
print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

# Take input from the user
choice = input("Enter choice(1/2/3/4):")
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

if choice == '1':
    print(num1,"+",num2,"=", add(num1,num2))
elif choice == '2':
    print(num1,"-",num2,"=", subtract(num1,num2))
elif choice == '3':
    print(num1,"*",num2,"=", multiply(num1,num2))
elif choice == '4':
    print(num1,"/",num2,"=", divide(num1,num2))
else:
    print("Invalid input")

```

10. Viết hàm “def Tinh(R):” tính chu vi và diện tích hình tròn, với bán kính R được nhập từ bàn phím, và kiểm tra giá trị bán kính đầu vào là hợp lệ.

Gợi ý: sử dụng thư viện “import math” và math.pi cho số pi nếu cần

11. Biết lãi suất tiết kiệm là $t\%$ /tháng (nhập t từ bàn phím). Nhập số vốn ban đầu n và số tháng gửi k . Tính số tiền nhận được sau k tháng sử dụng cấu trúc hàm

```
def benefit(t,n,k):
```

1.5. Câu hỏi kiểm tra

- Giải thích quy tắc khai báo hàm, gọi hàm, giá trị trả về, tham số truyền vào, tham số mặc định, phạm vi của biến trong python, sử dụng các hàm có sẵn trong các thư viện của python trong các chương trình đã thực thi

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

Bài 4. Các kiểu dữ liệu có cấu trúc trong Python

1.1. Mục đích

Giúp sinh viên nắm bắt các kiểu dữ liệu có cấu trúc trong lập trình python.

1.2. Cơ sở lý thuyết

Xem các quy tắc sử dụng các kiểu dữ liệu: chuỗi, số, list, tuple, set và dictionary trong python.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

1. Nhập chuỗi S và in ra từng kí tự của S, mỗi kí tự trên một dòng.

```
S = input('Nhập chuỗi:')  
for ch in S:  
    print(ch)
```

2. Chỉnh sửa ví dụ trên: hãy bỏ qua không in ra những kí tự “không nhìn thấy” (dấu space và dấu tab).
3. Chỉnh sửa ví dụ ở bài 1: hãy các kí tự ở dạng IN HOA.
4. Nhập một danh sách trên một dòng, mỗi phần tử cách nhau bởi dấu trống hoặc tab, sau đó in ra dãy vừa nhập ra màn hình

```
ds = input('Danh sách: ').split()  
# in cả dãy vừa nhập  
print(ds)  
# in dãy vừa nhập, mỗi phần tử trên một dòng  
for so in ds:  
    print(so)
```

5. Chỉnh sửa ví dụ ở bài 4: nhập 1 danh sách các từ từ bàn phím, in ra các từ đó theo thứ tự ngược lại thứ tự vừa nhập (ví dụ nhập “mot hai ba” thì in ra “ba hai mot”)
6. Nhập một tên người từ bàn phím, hãy tách phần họ và tên riêng của người đó và in chúng ra màn hình (giả thiết họ và tên riêng chỉ gồm một âm).
7. Nhập một chuỗi từ bàn phím, hãy loại bỏ tất cả các chữ số khỏi chuỗi và in lại nội dung chuỗi mới ra màn hình.
8. Nhập một dãy các từ từ bàn phím, hãy in ra từ dài nhất trong dãy vừa nhập, in ra mọi từ có cùng độ dài nhất.
9. Nhập một list từ bàn phím

```
ds = input('Nhập chuỗi:').split()
```

10. Cắt list: lấy list nhưng bỏ phần tử đầu và cuối

```
x = ds[0:-1]
```

```
for c in x:
```

```
    print(c)
```

11. Thêm phần tử vào list

```
ds.append('abc')
```

```
for ch in ds:
```

```
    print(ch)
```

12. Bỏ phần tử khỏi list

```
ds.remove('123')
```

```
for ch in ds:
```

```
    print(ch)
```

13. Tìm kiếm phần tử trong list

```
print("vị trí của chuỗi abc là", ds.index('abc'))
```

14. Sắp xếp các phần tử trong list

```
ds.sort()
```

```
for ch in ds:
```

```
    print(ch)
```

15. Người dùng nhập từ bàn phím liên tiếp các từ tiếng Anh viết tách nhau bởi dấu cách. Hãy nhập chuỗi đầu vào và tách thành các từ sau đó in ra màn hình các từ đó theo thứ tự từ điển.

16. Người dùng nhập từ bàn phím chuỗi các số nhị phân viết liên tiếp được nối nhau bởi dấu phẩy. Hãy nhập chuỗi đầu vào sau đó in ra những giá trị được nhập.

17. Nhập số n, in ra màn hình các số nguyên dương nhỏ hơn n có tổng các ước số lớn hơn chính nó.

18. Hãy nhập số nguyên n, tạo một list gồm các số fibonacci nhỏ hơn n và in ra màn hình.

19. Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu.

20. Nhập n, in n dòng đầu tiên của tam giác pascal.

21. Viết một chương trình chấp nhận đầu vào là chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy, kiểm tra xem chúng có chia hết cho 5 không. Sau đó in các số chia hết cho 5 thành dãy phân tách bởi dấu phẩy.

Ví dụ đầu vào là: 0100,0011,1010,1001

Đầu ra sẽ là: 1010

22. Viết một chương trình tìm tất cả các số trong đoạn 1000 và 3000 (tính cả 2 số này) sao cho tất cả các chữ số trong số đó là số chẵn. In các số tìm được thành chuỗi cách nhau bởi dấu phẩy, trên một dòng.

23. Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó. Giả sử đầu vào sau được cấp cho chương trình: hello world! 123

Thì đầu ra sẽ là:

Số chữ cái là: 10

Số chữ số là: 3

24. Viết một chương trình chấp nhận đầu vào là một câu, đếm chữ hoa, chữ thường.

Giả sử đầu vào là: Dai Hoc Vinh

Thì đầu ra là:

Chữ hoa: 3

Chữ thường: 7

25. Sử dụng một danh sách để lọc các số lẻ từ danh sách được người dùng nhập vào.

Giả sử đầu vào là: 1,2,3,4,5,6,7,8,9 thì đầu ra phải là: 1,3,5,7,9

26. Viết chương trình tính số tiền thực của một tài khoản ngân hàng dựa trên nhật ký giao dịch được nhập vào từ giao diện điều khiển.

Định dạng nhật ký được hiển thị như sau:

D 100

W 200

(D là tiền gửi, W là tiền rút ra).

Giả sử đầu vào được cung cấp là:

D 300

D 300

W 200

D 100

Thì đầu ra sẽ là:

500

1.5. Câu hỏi kiểm tra

- Giải thích chương trình đã viết
- Các hàm và các phương thức xây dựng sẵn cho các kiểu dữ liệu có cấu trúc
- Định nghĩa các kiểu dữ liệu có cấu trúc, phân biệt sự khác nhau, giải thích

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

Bài 5. Thiết kế module trong Python

1.1. Mục đích

Giúp sinh viên nắm bắt việc thiết kế module trong lập trình python, sử dụng module thư viện numpy trong các ứng dụng.

1.2. Cơ sở lý thuyết

Xem các quy tắc khai báo, thiết kế và sử dụng module trong python, các thuật toán tìm kiếm, sắp xếp, cài đặt và sử dụng thư viện numpy.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

1. Sử dụng module. Định nghĩa một module toán học gọi là mymath và sử dụng module này từ một tập lệnh riêng biệt.

```
## File: mymath.py ##
def square(n):
    return n*n

def cube(n):
    return n*n*n

def average(values):
    nvals = len(values)
    sum = 0.0
    for v in values:
        sum += v
    return float(sum)/nvals
```

Sử dụng các hàm bằng cách sử dụng “import” gọi hàm theo cấu trúc “module.functionname”.

```
## My script using the math module ##
import mymath # Note no .py

values = [2,4,6,8,10]
print('Squares:')
for v in values:
    print(mymath.square(v))
print 'Cubes:'
for v in values:
    print(mymath.cube(v))

print('Average: ' + str(mymath.average(values)))
```

Cũng có thể sử dụng “import module as new-name”

```
import mymath as mt
```

```
print(mt.square(2))
print(mt.square(3))
```

2. Sử dụng thư viện tiêu chuẩn của python (datetime)

```
import datetime as dt
format = '%Y-%m-%dT%H:%M:%S'
t1 = dt.datetime.strptime('2008-10-12T14:45:52', format)
print('Day ' + str(t1.day))
print('Month ' + str(t1.month))
print('Minute ' + str(t1.minute))
print('Second ' + str(t1.second))

# Define today's date and time
t2 = dt.datetime.now()
diff = t2 - t1
print('How many days difference? ' + str(diff.days))
```

3. Viết chương trình sử dụng thư viện NumPy để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 38

Đầu ra:

[12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37]

```
import numpy as np
x = np.arange(12, 38)
print(x)
```

4. Viết chương trình để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 38 và đảo ngược mảng đã tạo (phần tử đầu tiên trở thành cuối cùng)

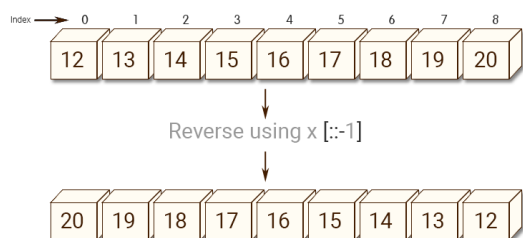
Mảng được tạo:

[12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37]

Mảng đảo ngược:

[37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12]

Gợi ý:



5. Viết chương trình tìm phần tử lớn nhất và nhỏ nhất của một danh sách

- Số lượng và giá trị của list được nhập từ bàn phím
- Phương thức sắp xếp và tìm phần tử lớn nhất được viết thành module

6. In ra vị trí phần tử lớn nhất và nhỏ nhất tìm được ở bài tập trên

7. Viết chương trình sử dụng thư viện NumPy để tạo một mảng có cấu trúc từ tên sinh viên, chiều cao, lớp và các kiểu dữ liệu của họ. Bây giờ sắp xếp các mảng theo chiều cao.

```
import numpy as np
data_type = [('name', 'S15'), ('class', int), ('height', float)]
students_details = [('James', 5, 48.5), ('Nail', 6, 52.5), ('Paul', 5, 42.10), ('Pit', 5, 40.11)]
# create a structured array
students = np.array(students_details, dtype=data_type)
print("Original array:")
print(students)
print("Sort by height")
print(np.sort(students, order='height'))
```

8. Xây dựng hàm “Sequential_Search(dlist, item)” (giải thuật tìm kiếm tuyến tính) dưới dạng module. Viết chương trình nhập một dlist n phần tử từ bàn phím và tìm kiếm phần tử item bất kỳ.

Sequential_Search([11,23,58,31,56,77,43,12,65,19],31) -> (True, 3)

Gợi ý:

Giải thuật tìm kiếm tuyến tính (Mảng A, Giá trị x)

Bước 1: Thiết lập i thành 1

Bước 2: Nếu i > n thì chuyển tới bước 7

Bước 3: Nếu A[i] = x thì chuyển tới bước 6

Bước 4: Thiết lập i thành i + 1

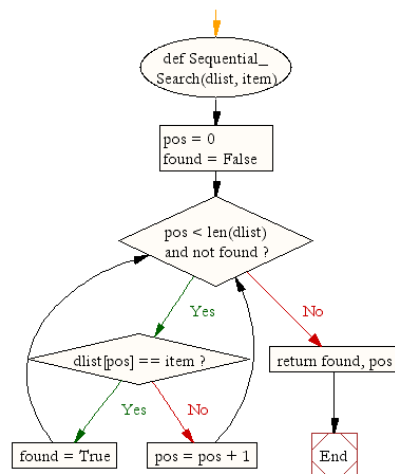
Bước 5: Tới bước 2

Bước 6: In phần tử x được tìm thấy tại chỉ mục i và tới bước 8

Bước 7: In phần tử không được tìm thấy

Bước 8: Thoát

Flowchart:



9. Xây dựng hàm “binary_search(list, value)” (giải thuật tìm kiếm nhị phân) dưới dạng module. Viết chương trình nhập một list n phần tử từ bàn phím và tìm kiếm phần tử value bất kỳ.

binary_search([1,2,3,5,8], 6) -> False

binary_search([1,2,3,5,8], 5) -> True

Gợi ý:

Giải thuật kiểm nhị phân (Binary Search)

$A \leftarrow$ một mảng đã được sắp xếp

$n \leftarrow$ kích cỡ mảng

$x \leftarrow$ giá trị để tìm kiếm trong mảng

gán $\text{lowerBound} = 1$

gán $\text{upperBound} = n$

while x not found

if $\text{upperBound} < \text{lowerBound}$

EXIT: x không tồn tại.

gán $\text{midPoint} = \text{lowerBound} + (\text{upperBound} - \text{lowerBound}) / 2$

if $A[\text{midPoint}] < x$

gán $\text{lowerBound} = \text{midPoint} + 1$

if $A[\text{midPoint}] > x$

gán $\text{upperBound} = \text{midPoint} - 1$

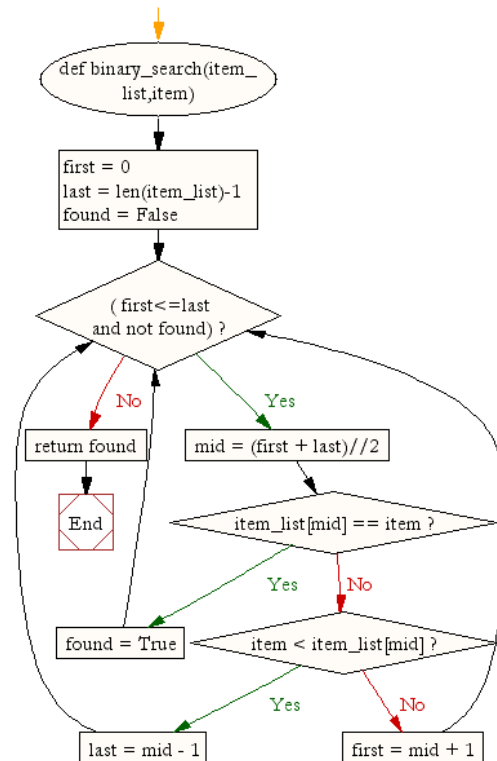
if $A[\text{midPoint}] = x$

EXIT: x được tìm thấy tại midPoint

kết thúc while

kết thúc giải thuật

Flowchart:



10. Xây dựng hàm “*bubbleSort (nlist)*” (giải thuật sắp xếp nổi bọt) dưới dạng module.

Viết chương trình nhập một *nlist* n phần tử từ bàn phím và sắp xếp.

Sample Data: [14,46,43,27,57,41,45,21,70]

Expected Result: [14, 21, 27, 41, 43, 45, 46, 57, 70]

Gợi ý:

Bắt đầu hàm bubbleSort(list : mảng các phần tử)

```
loop = list.count;
```

```
for i = 0 tới loop-1 thực hiện:
```

```
    swapped = false
```

```
    for j = 0 tới loop-1 thực hiện:
```

```
        /* so sánh các phần tử cạnh nhau */
```

```
        if list[j] > list[j+1] then
```

```
            /* trao đổi chúng */
```

```
            swap( list[j], list[j+1] )
```

```
            swapped = true
```

```
        kết thúc if
```

```
    kết thúc for
```

```
    /*Nếu không cần trao đổi phần tử nào nữa thì  
    tức là mảng đã được sắp xếp. Thoát khỏi vòng lặp.*/
```

```
    if(not swapped) then
```

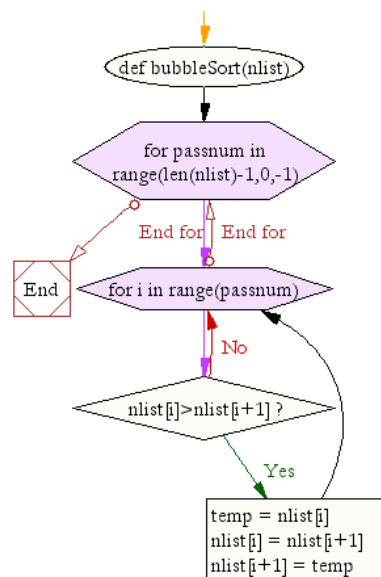
```
        break
```

```
    kết thúc if
```

```
    kết thúc for
```

```
    Kết thúc hàm return list
```

Flowchart:



11. Viết chương trình sử dụng thư viện NumPy để tạo một mảng có cấu trúc từ tên sinh viên, chiều cao, lớp và các kiểu dữ liệu của họ. Bây giờ sắp xếp theo lớp, sau đó chiều cao nếu lớp bằng nhau.

Dữ liệu đầu vào:

```
[('James', 5, 48.5 ) ('Nail', 6, 52.5 ) ('Paul', 5, 42.1 ) ('Pit', 5, 40.11)]
```

Kết quả sắp xếp:

```
[('Pit', 5, 40.11) ('Paul', 5, 42.1 ) ('James', 5, 48.5 ) ('Nail', 6, 52.5 )]
```

12. Viết chương trình sử dụng thư viện NumPy để sắp xếp id sinh viên với chiều cao tăng dần của sinh viên từ id sinh viên và chiều cao đã cho. In các chỉ số nguyên mô tả thứ tự sắp xếp theo nhiều cột và dữ liệu được sắp xếp (sử dụng hàm *lexsort()*)

Dữ liệu đầu vào:

```
student_id = [1023, 5202, 6230, 1671, 1682, 5241, 4532]
```

```
student_height = [40., 42., 45., 41., 38., 40., 42.0]
```

Kết quả:

Chỉ số:

```
[4 0 5 3 6 1 2]
```

Dữ liệu sắp xếp:

```
1682      38.0
```

```
1023      40.0
```

```
5241      40.0
```

```
1671      41.0
```

```
4532      42.0
```

```
5202      42.0
```

```
6230      45.0
```

1.5. Câu hỏi kiểm tra

- Giải thích chương trình đã viết
- Tạo và sử dụng module trong python
- Các thuật toán sắp xếp, tìm kiếm
- Thư viện numpy

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

Bài 6: Lập trình hướng đối tượng trong Python

1.1. Mục đích

Giúp sinh viên nắm bắt về lập trình hướng đối tượng trong lập trình python.

1.2. Cơ sở lý thuyết

Xem các quy tắc xây dựng class, các phương thức và thuộc tính của đối tượng.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

1. Định nghĩa một class có tên là Circle có thể được xây dựng từ bán kính. Circle có một method có thể tính diện tích.

Gợi ý:

Sử dụng def methodName(self) để định nghĩa method.

```
class Circle(object):
    def __init__(self, r):
        self.radius = r
    #####
    def area(self):
        return self.radius**2*3.14

aCircle = Circle(2)
print (aCircle.area())
```

2. Định nghĩa class có tên là *Hinhchunhat* được xây dựng bằng chiều dài và chiều rộng. Class *Hinhchunhat* có method để tính diện tích.
3. Định nghĩa class *Nguoi* và 2 class con của nó: *Nam*, *Nu*. Tất cả các class có method "getGender" có thể in "Nam" cho class *Nam* và "Nữ" cho class *Nu*.

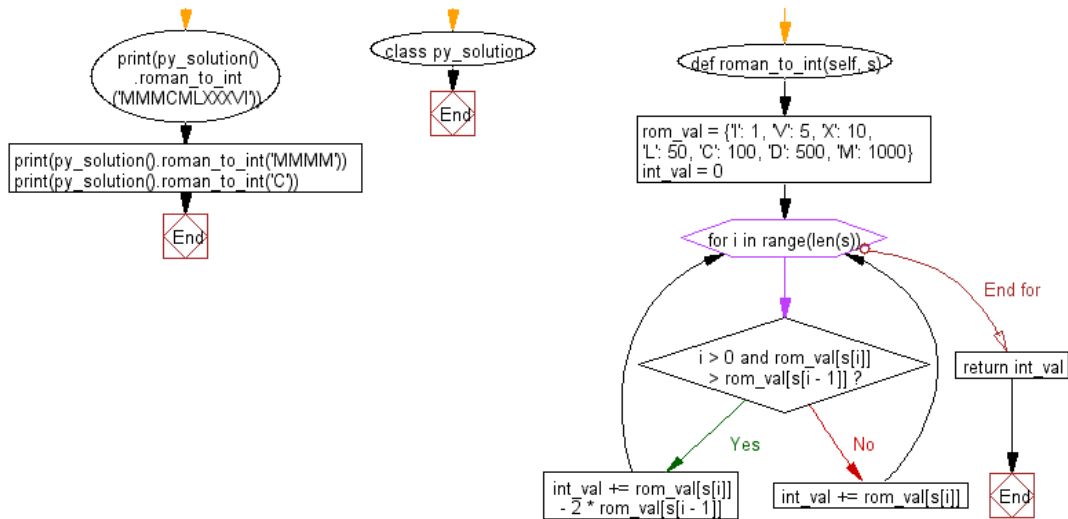
```
class Nguoi(object):
    def getGender( self ):
        return "Unknown"

class Nam( Nguoi ):
    def getGender( self ):
        return "Nam"
# Code by Quantrimang.com
class Nu( Nguoi ):
    def getGender( self ):
        return "Nữ"

aNam = Nam()
aNu= Nu()
print (aNam.getGender())
print (aNu.getGender())
```

4. Viết chương trình Python dưới dạng class để chuyển đổi một số La Mã thành một số nguyên.

Flowchart:

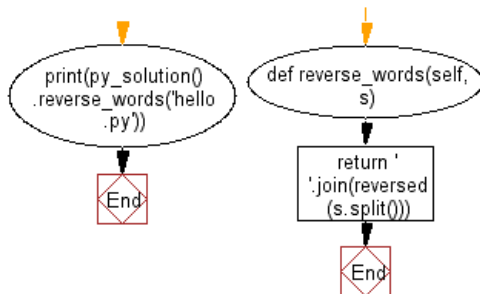


5. Viết chương trình Python dưới dạng class để đảo ngược chuỗi từ từng chữ.

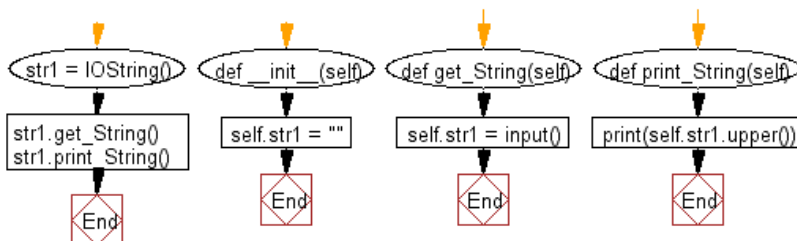
Dữ liệu vào : 'hello .py'

Đầu ra : '.py hello'

Flowchart:



6. Viết một class Python có hai phương thức get_String và print_String. get_String chấp nhận một chuỗi từ người dùng và print_String in chuỗi đó bằng chữ in hoa.



7. Viết một class Python có tên Circle được xây dựng theo bán kính và hai phương thức sẽ tính diện tích và chu vi của hình tròn.
8. Chương trình ATM đơn giản

```
class Bank:
    Account_type = "Savings"
    location = "Guntur"
    def __init__(self, name, Account_Number, balance):
        self.name = name
        self.Account_Number = Account_Number
        self.balance = balance
        self.Account_type = Bank.Account_type
        self.location = Bank.location

    def __repr__(self):
        print ("Welcome to the SBI ATM Machine ")
        print ("-----")
        account_pin = int(input("Please enter your pin number
"))
        if(account_pin==123):
            Account(self)
        else:
            print("Pin Incorrect. Please try again")
            Error(self)
        return ' '.join([self.name, self.Account_Number])

    def Error(self):

        account_pin = int(input("Please enter your pin number "))
        if(account_pin==123):
            Account(self)
        else:
            print("Pin Incorrect. Please try again")
            Error(self)

    def Account(self):
        print ("Your Card Number is:XXXX XXXX XXXX 1337")
        print ("Would you like to deposit/withdraw/Check
Balance?")
        print("""
1)      Balance
2)      Withdraw
3)      Deposit
4)      Quit
```

```

    """
    option=int(input("Please enter your choice:"))
    if(option==1):
        Balance(self)
    elif(option==2):
        Withdraw(self)
    elif(option==3):
        Deposit(self)
    elif(option==4):
        exit()
def Balance(self):
    print("Balance:",self.balance)
    Account(self)
def Withdraw(self):
    w=int(input("Please Enter Desired amount: "))
    if(self.balance>0 and self.balance>=w):
        self.balance=self.balance-w
        print("Your transaction is successfull")
        print("your Balance:",self.balance)
        print("")
    else:
        print("Your transaction is cancelled due to")
        print("Amount is not sufficient in your account")
    Account(self)
def Deposit(self):
    d=int(input("Please Enter Desired amount: "))
    self.balance=self.balance+d
    print("Your transaction is successfull")
    print("Balance:",self.balance)
    Account(self)
def Exit():
    print ("Exit")
t1 = Bank('mahesh', 1453210145,5000)

print (t1)

```

1.5. Câu hỏi kiểm tra

- Giải thích chương trình đã viết
- Các khái niệm về lớp , đối tượng, thuộc tính, phương thức

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

Bài 7: Thao tác trên tập tin và thư mục trong Python

1.1. Mục đích

Giúp sinh viên có thể thao tác với các file văn bản sử dụng python.

1.2. Cơ sở lý thuyết

Xem các quy tắc mở, đóng file văn bản, đọc và ghi nội dung của các file.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

1. Chương trình đọc file và in đảo ngược kết quả

```
input_file=open('D:/a.txt','r')
for line in input_file:
    l=len(line)
    s=' '
    while(l>=1):
        s=s+line[l-1]
        l=l-1
    print(s)
input_file.close()
```

2. Chương trình đọc một file, tính số ký tự, số từ và số dòng của file

```
k=open('D:/a.txt','r')
char,wc,lc=0,0,0
for line in k:
    for k in range(0,len(line)):
        char +=1
        if(line[k]==' '):
            wc+=1
        if(line[k]=='\n'):
            wc,lc=wc+1,lc+1
print("The no.of chars is %d\n The no.of words is %d\n The
no.of lines is %d"%(char,wc,lc))
```

3. Viết chương trình Python để đọc toàn bộ tệp văn bản

4. Chương trình Python để đọc n dòng đầu tiên của tệp

```
def file_read_from_head(fname, nlines):
    from itertools import islice
    with open(fname) as f:
        for line in islice(f, nlines):
```

```

        print(line)
file_read_from_head('test.txt',2)

```

5. Chương trình Python để nối văn bản vào tệp và hiển thị văn bản.

```

def file_read(fname):
    from itertools import islice
    with open(fname, "w") as myfile:
        myfile.write("Python Exercises\n")
        myfile.write("Java Exercises")
    txt = open(fname)
    print(txt.read())
file_read('abc.txt')

```

6. Chương trình Python để đọc n dòng cuối cùng của tệp

```

import sys
import os
def file_read_from_tail(fname,lines):
    bufsize = 8192
    fsize = os.stat(fname).st_size
    iter = 0
    with open(fname) as f:
        if bufsize > fsize:
            bufsize = fsize-1
            data = []
            while True:
                iter +=1
                f.seek(fsize-bufsize*iter)
                data.extend(f.readlines())
                if len(data) >= lines or f.tell() == 0:
                    print(''.join(data[-lines:]))
                    break

file_read_from_tail('test.txt',2)

```

7. Viết chương trình Python để đếm số dòng trong tệp văn bản

8. Viết chương trình Python để viết nội dung danh sách vào tệp.

9. Viết chương trình Python để sao chép nội dung của tệp này sang tệp khác.

10. Viết chương trình python để tìm những từ dài nhất trong văn bản

1.5. Câu hỏi kiểm tra

- Giải thích chương trình đã viết.
- Các thao tác với file văn bản.

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.

Bài 8: Lập trình giao diện trong Python

1.1. Mục đích

Giúp sinh viên làm quen với lập trình đồ họa và sử dụng thư viện Tkinter trong python.

1.2. Cơ sở lý thuyết

Xem các quy tắc quản lý layout, widget, hộp thoại, đồ họa trong python.

1.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

1.4. Các bước tiến hành

Thực hiện các bài tập sau sử dụng python

- Viết chương trình mẫu và xem kết quả

```
import turtle

window = turtle.Screen()
window.bgcolor("lightgreen")

painter = turtle.Turtle()
painter.fillcolor('blue')
painter.pencolor('blue')
painter.pensize(3)

def drawsq(t, s):
    for i in range(4):
        t.forward(s)
        t.left(90)

for i in range(1,180):
    painter.left(18)
    drawsq(painter, 200)
```

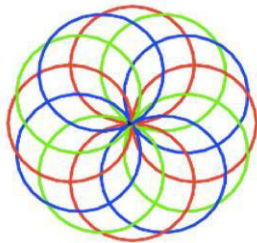
- Viết chương trình sau và xem kết quả

```

import turtle, random
colors =
["red", "green", "blue", "orange", "purple", "pink", "yellow"]
painter = turtle.Turtle()
painter.pensize(3)
for i in range(10):
    color = random.choice(colors)
    painter.pencolor(color)
    painter.circle(100)
    painter.right(30)
    painter.left(60)
    painter.setposition(0, 0)

```

3. Dựa trên các kết quả đạt được từ các chương trình trên hãy viết chương trình hiển thị hình ảnh đồ họa sau



4. Viết chương trình sau và kiểm tra kết quả, giải thích chương trình

```

from tkinter import *

window = Tk()

window.title("Welcome to LikeGeeks app")

window.geometry('350x200')

lbl = Label(window, text="Hello")

lbl.grid(column=0, row=0)

def clicked():

    lbl.configure(text="Button was clicked !!")

btn = Button(window, text="Click Me", command=clicked)

btn.grid(column=1, row=0)

window.mainloop()

```

5. Viết chương trình sau và kiểm tra kết quả, giải thích chương trình

```

from tkinter import *

canvas_width = 400
canvas_height = 400

```

```

python_green = "#476042"

def polygon_star(canvas, x,y,p,t, outline=python_green, fill='yellow',
width = 1):
    points = []
    for i in (1,-1):
        points.extend((x, y + i*p))
        points.extend((x + i*t, y + i*t))
        points.extend((x + i*p, y))
        points.extend((x + i*t, y - i * t))

    print(points)

    canvas.create_polygon(points, outline=outline,
                        fill=fill, width=width)

master = Tk()

w = Canvas(master,
            width=canvas_width,
            height=canvas_height)
w.pack()

p = 50
t = 15

nsteps = 10
step_x = int(canvas_width / nsteps)
step_y = int(canvas_height / nsteps)

for i in range(1, nsteps):
    polygon_star(w,i*step_x,i*step_y,p,t,outline='red',fill='gold',
width=3)
    polygon_star(w,i*step_x,canvas_height -
i*step_y,p,t,outline='red',fill='gold', width=3)

mainloop()

```

6. Viết chương trình sau và kiểm tra kết quả, giải thích chương trình

```

from tkinter import *

def checkered(canvas, line_distance):
    # vertical lines at an interval of "line_distance" pixel
    for x in range(line_distance,canvas_width,line_distance):
        canvas.create_line(x, 0, x, canvas_height, fill="#476042")
    # horizontal lines at an interval of "line_distance" pixel
    for y in range(line_distance,canvas_height,line_distance):
        canvas.create_line(0, y, canvas_width, y, fill="#476042")

master = Tk()
canvas_width = 200
canvas_height = 100
w = Canvas(master,
            width=canvas_width,
            height=canvas_height)
w.pack()

```

```
checkered(w,10)
```

```
mainloop()
```

7. Chương trình graphic game

```
# import the modules
import tkinter
import random

# list of possible colour.
colours = ['Red','Blue','Green','Pink','Black',
           'Yellow','Orange','White','Purple','Brown']
score = 0

# the game time left, initially 30 seconds.
timeleft = 30

# function that will start the game.
def startGame(event):

    if timeleft == 30:

        # start the countdown timer.
        countdown()

    # run the function to
    # choose the next colour.
    nextColour()

# Function to choose and
# display the next colour.
def nextColour():

    # use the globally declared 'score'
    # and 'play' variables above.
    global score
    global timeleft

    # if a game is currently in play
    if timeleft > 0:

        # make the text entry box active.
        e.focus_set()

        # if the colour typed is equal
        # to the colour of the text
        if e.get().lower() == colours[1].lower():

            score += 1

        # clear the text entry box.
        e.delete(0, tkinter.END)
```

```

        random.shuffle(colours)

        # change the colour to type, by changing the
        # text _and_ the colour to a random colour value
        label.config(fg = str(colours[1]), text = str(colours[0]))

        # update the score.
        scoreLabel.config(text = "Score: " + str(score))

# Countdown timer function
def countdown():

    global timeleft

    # if a game is in play
    if timeleft > 0:

        # decrement the timer.
        timeleft -= 1

        # update the time left label
        timeLabel.config(text = "Time left: "
                             + str(timeleft))

        # run the function again after 1 second.
        timeLabel.after(1000, countdown)

# Driver Code

# create a GUI window
root = tkinter.Tk()

# set the title
root.title("COLORGAME")

# set the size
root.geometry("375x200")

# add an instructions label
instructions = tkinter.Label(root, text = "Type in the colour"
                              "of the words, and not the word text!",
                              font = ('Helvetica', 12))
instructions.pack()

# add a score label
scoreLabel = tkinter.Label(root, text = "Press enter to start",
                             font = ('Helvetica', 12))
scoreLabel.pack()

# add a time left label
timeLabel = tkinter.Label(root, text = "Time left: " +
                             str(timeleft), font = ('Helvetica', 12))

timeLabel.pack()

```

```
# add a label for displaying the colours
label = tkinter.Label(root, font = ('Helvetica', 60))
label.pack()

# add a text entry box for
# typing in colours
e = tkinter.Entry(root)

# run the 'startGame' function
# when the enter key is pressed
root.bind('<Return>', startGame)
e.pack()

# set focus on the entry box
e.focus_set()

# start the GUI
root.mainloop()
```

8. Viết chương trình graphic sử dụng thư viện Tkinter.

1.5. Câu hỏi kiểm tra

- Giải thích chương trình đã viết.
- Sử dụng thư viện Tkinter.

1.6. Tài liệu tham khảo

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015.