

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.

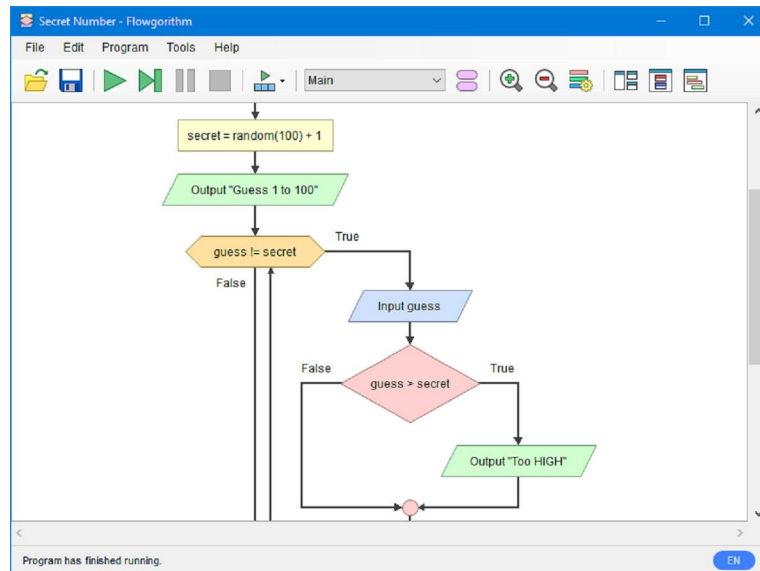
Một khi bạn hiểu logic lập trình, bạn sẽ dễ dàng học một trong những ngôn ngữ chính. Flowgorithm có thể tương tác chuyển đổi sơ đồ của bạn sang hơn 18 ngôn ngữ. Chúng bao gồm: C #, C ++, Java, JavaScript, Lua, Perl, Python, Ruby, Swift, Visual Basic .NET và VBA (được sử dụng trong Office).

#### **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**

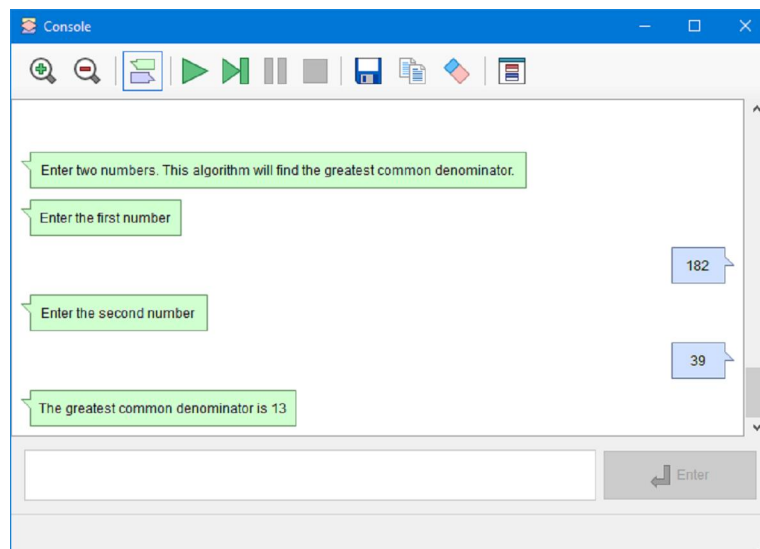
*a. Giao diện chính*



Hình 1. Main window

### b. Cửa sổ điều khiển

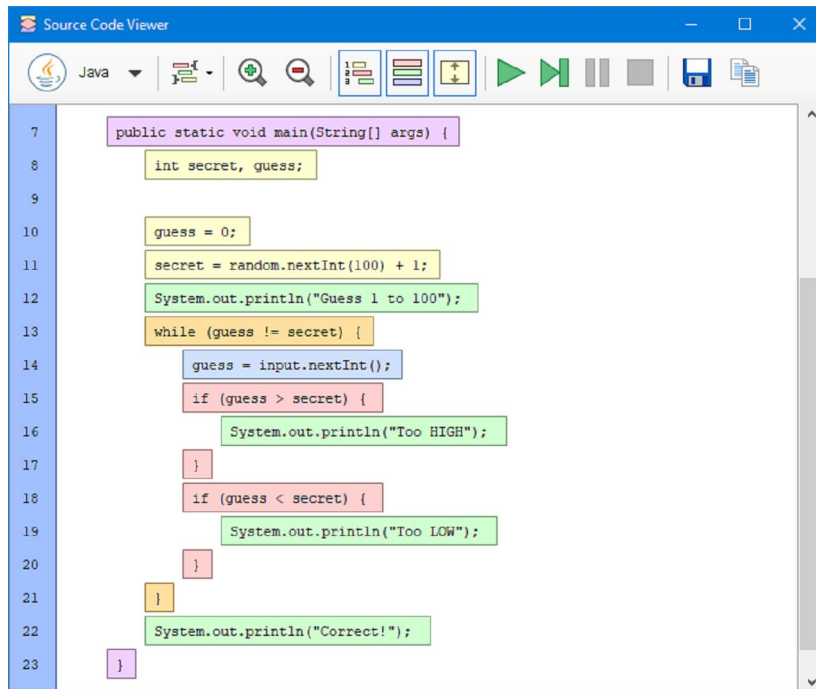
Phương pháp cổ điển để tương tác với máy tính là sử dụng "Bảng điều khiển". Thuật toán lưu lượng cố gắng làm cho nó trông giống như một cửa sổ nhắn tin tức thời điển hình. "Bong bóng trò chuyện" được mã hóa màu để khớp với hình dạng Đầu vào và Đầu ra được sử dụng trong sơ đồ. Nếu bạn không muốn sử dụng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

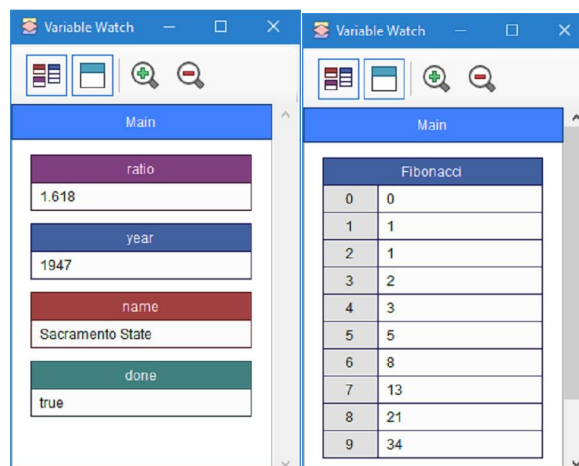
Trình xem mã nguồn có thể chuyển đổi sơ đồ của bạn sang một số ngôn ngữ lập trình chính.



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

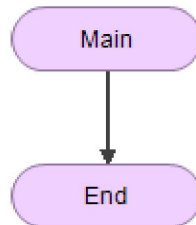
#### 1.3.2. Vật tư tiêu hao: Không

### 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

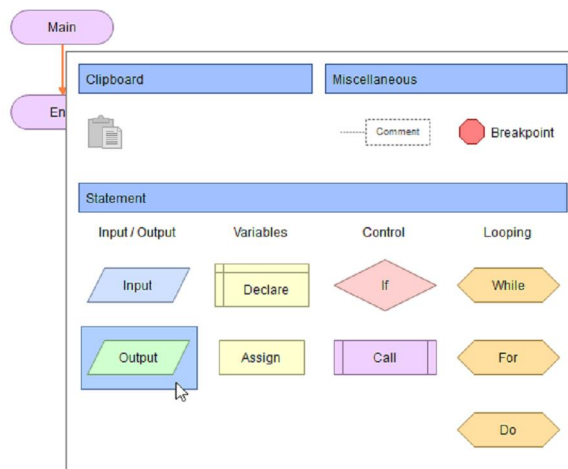
- Khi bắt đầu một sơ đồ mới, chúng ta sẽ thấy hai hình chữ nhật tròn được gọi là "terminals". Những biểu tượng này đại diện cho sự bắt đầu và kết thúc chương trình của bạn.
- Nhiều sơ đồ thuật toán hiển thị văn bản "Begin" trong terminal. Flowgorithm sử dụng văn bản "Main". Hầu hết các ngôn ngữ lập trình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



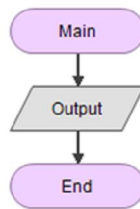
Hình 5. Bắt đầu và kết thúc chương trình

#### 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.
- Hiện thị giao diện với các khối cần thêm
- Lựa chọn và click vào khối cần thêm sẽ được chương trình như hình 7.



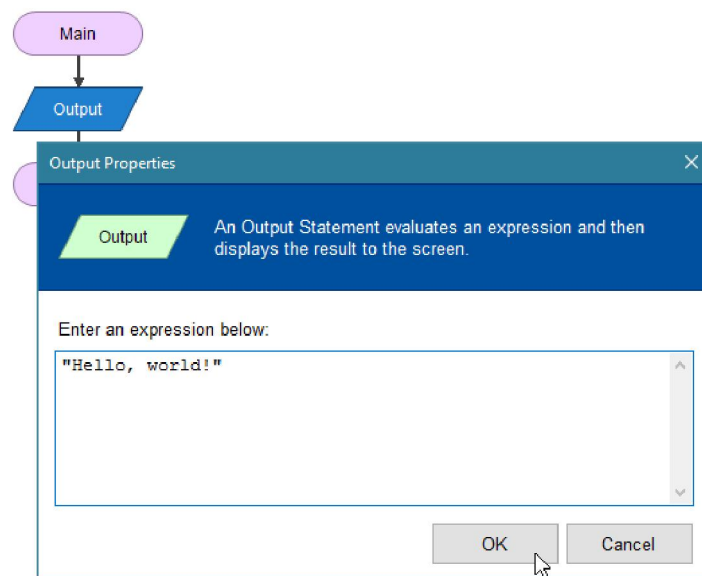
Hình 6. Thêm các khối chức năng cho chương trình



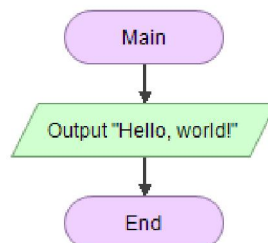
Hình 7. Chương trình đã thêm các khối chức năng

*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.



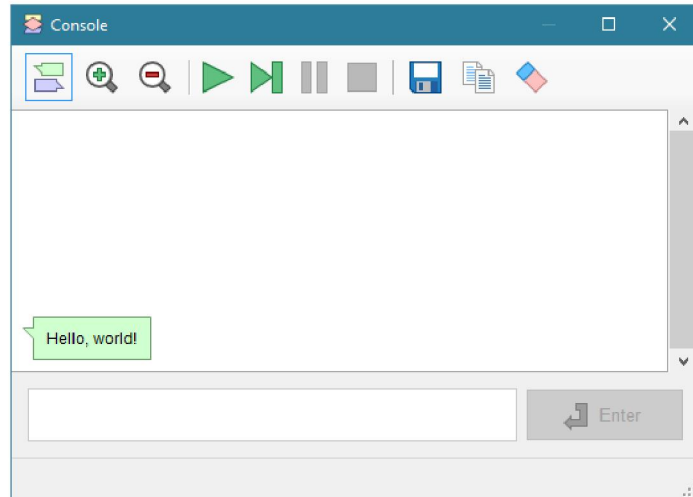
Hình 8. Định nghĩa chức năng cho khối được thêm vào



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

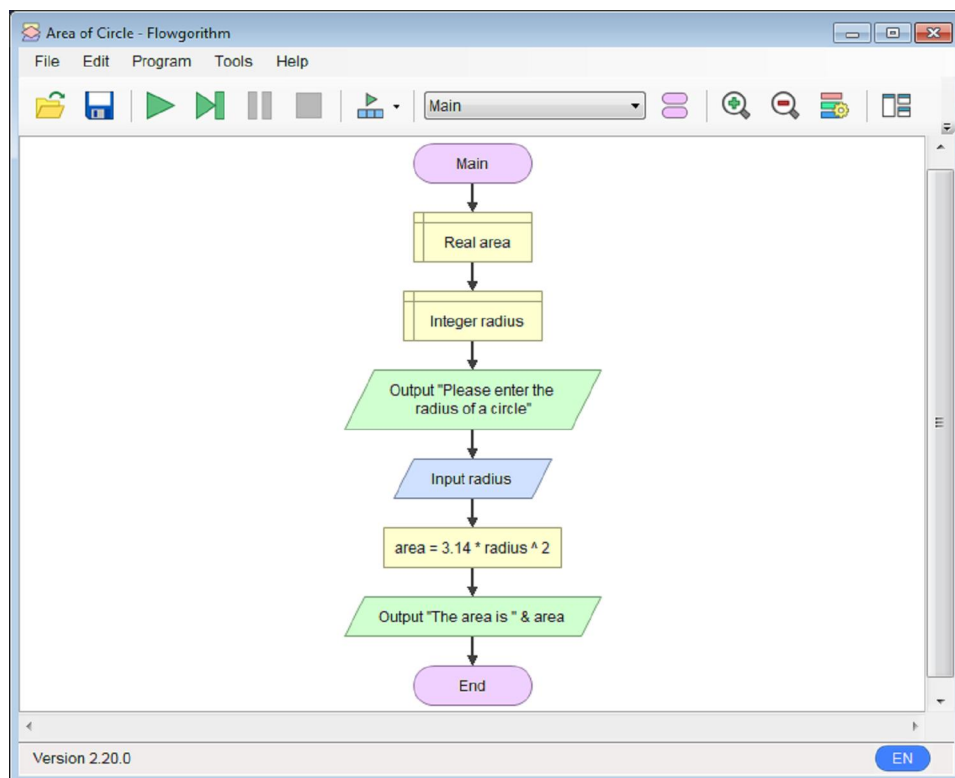


Hình 10. Cửa sổ điều khiển hiển thị kết quả

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

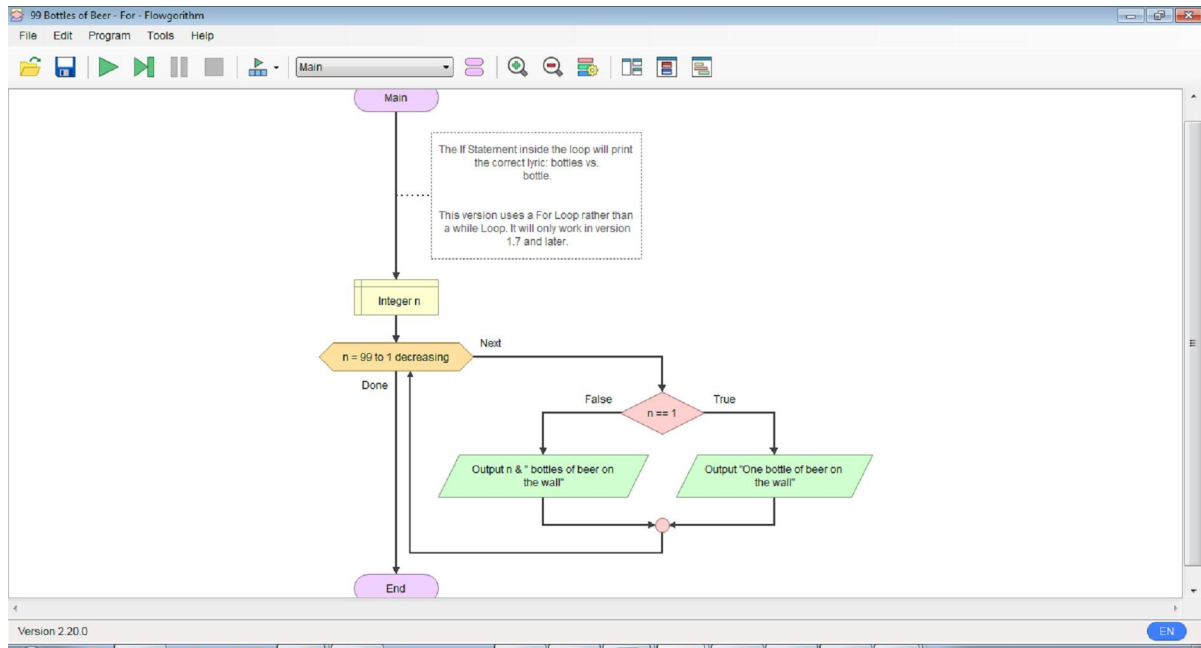
#### 1.4.2. Sử dụng Flowgorithm xây dựng chương trình giải quyết các bài toán

*a. Tính diện tích hình tròn*



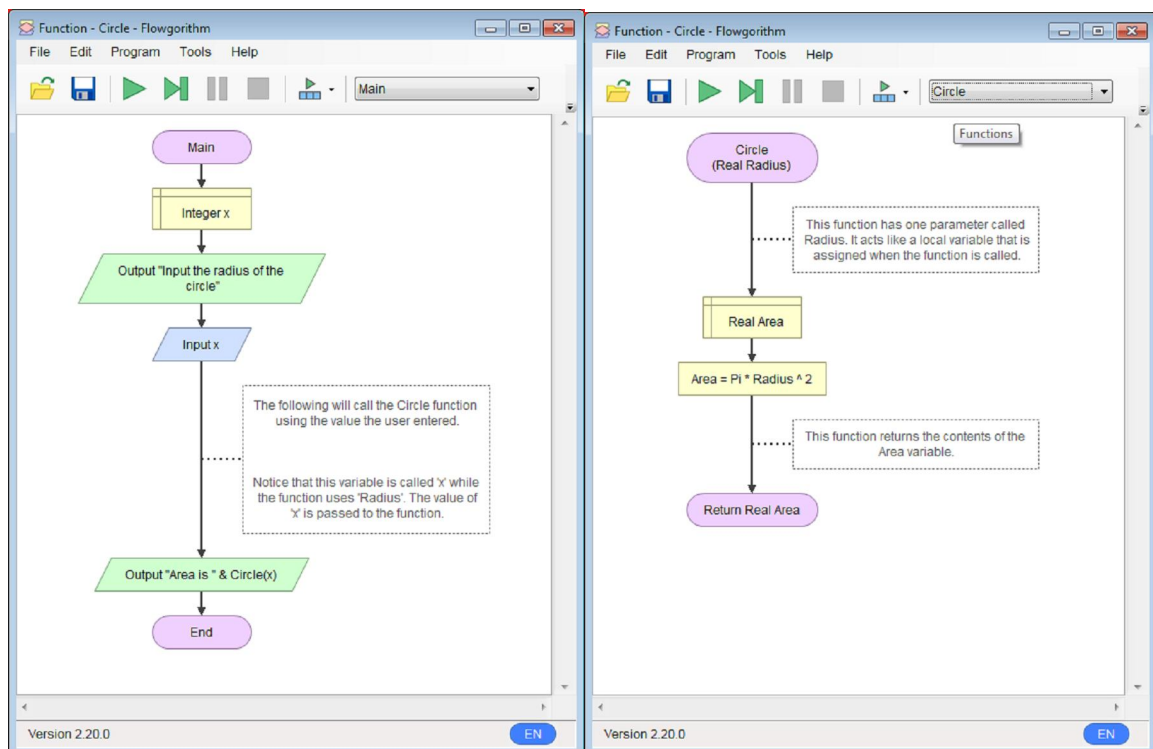
Hình 11. Sơ đồ giải thuật bài toán tính diện tích hình tròn

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 chuỗi 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 chuỗi 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 Tính(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.