**BÀI TẬP PYTHON**

**DẠNG 0: BÀI TẬP KHỞI ĐỘNG**

**Bài 1:** Viết chương trình tính cạnh huyền của một tam giác vuông cho trước.

import math

a = float(input("Nhập giá trị a: "))

b = float(input("Nhập giá trị b: "))

c = math.sqrt(math.pow(a,2)+ math.pow(b,2))

print ("Độ dài cạnh huyền là: ",c)

**Bài 2:** Viết một chương trình tính giá trị của a+ab+abc+abcd với a,b,c,d là số được nhập vào bởi người dùng.

a = input("Nhập số a: ")

b = input("Nhập số b: ")

c = input("Nhập số c: ")

d = input("Nhập số d: ")

n1 = int( "%s" % a )

n2 = int( "%s%s" % (a,b) )

n3 = int( "%s%s%s" % (a,b,c) )

n4 = int( "%s%s%s%s" % (a,b,c,d) )

print ("Tổng là: ",n1+n2+n3+n4)

**Bài 3:** Viết hàm tính âm lịch với số dương lịch cho trước? VD: Namam(2022) => Nhâm Dần

can = ['Canh', 'Tân', 'Nhâm', 'Quý', 'Giáp', 'Ất', 'Bính', 'Đinh','Mậu','Kỷ']

chi = ['Thân', 'Dậu', 'Tuất', 'Hợi','Tí','Sửu','Dần', 'Mão', 'Thìn', 'Tị', 'Ngọ', "Mùi"]

nam = int(input("Nhap vao nam: "))

vitri\_can = nam%10

vitri\_chi = nam%12

print(can[vitri\_can]+’ ‘+chi[vitri\_chi])

**Bài 4: Tính chính xác tuổi của bạn thông qua import datetime**

# Tính tuổi dựa trên ngày tháng năm sinh nhập vào

import datetime

a = int(input("Nhập vào ngày sinh của bạn: "))

b = int(input("Nhập vào tháng sinh của bạn: "))

c = int(input("Nhập vào năm sinh của bạn: "))

current\_year = datetime.date.today().year

current\_month = datetime.date.today().month

current\_day = datetime.date.today().day

age\_year = current\_year - c

age\_month = abs(current\_month-b)

age\_day = abs(current\_day-a)

print("Tuổi chính xác của bạn là ", age\_day,"ngày" , age\_month,"tháng", age\_year, "năm tuổi")

**Bài 5: Nhập dữ liệu trên hệ thống mức lương tối thiểu của công ty trả cho nhân viên, từ đó biết được lương của nhân viên:**

**Biết:**

**- Tổng giám đốc : Mức lương tối thiểu x 2.14**

**- Phó giám đốc: Mức lương tối thiểu x 2.03**

**- Quản lý: Mức lương tối thiểu x 1.82**

**- Còn lại Mức lương tối thiểu x 1.56**

total\_salary = float(input("Nhập mức lương tối thiểu của công ty: "))

CEO\_salary = total\_salary \* 2.14

Deputy\_CEO\_salary = total\_salary \* 2.03

Manager\_salary = total\_salary \* 1.82

Other\_salary = total\_salary \* 1.56

print("Mức lương của Tổng giám đốc là: ", CEO\_salary)

print("Mức lương của Phó giám đốc là: ", Deputy\_CEO\_salary)

print("Mức lương của Quản lý là: ", Manager\_salary)

print("Mức lương của nhân viên còn lại là: ", Other\_salary)

**Bài 6: Viết chương trình tính căn bậc N của 1 số bất kì.**

import math

number = float(input("Nhập số cần tính căn bậc n: "))

power = float(input("Nhập số mũ n: "))

result = math.pow(number, 1.0 / power)

print(f"Căn bậc {power} của {number} là {result}")

**DẠNG 1: Dạng if và if … else**

**Bài 7:** Viết chương trình để tìm số chẵn hay số lẻ. **(Có sử dụng khai báo hàm def)**

n = int(input("Nhập số tự nhiên n: "))

if n%2 == 0:

print ("Đây là một số chẵn")

else:

print ("Đây là một số lẻ")

**Bài 8:** Viết một chương trình có thể tính giai thừa của một số cho trước. Kết quả được in thành [chuỗi](https://quantrimang.com/gioi-thieu-qua-ve-chuoi-so-list-trong-python-140881) trên một dòng, phân tách bởi dấu phẩy. Ví dụ, số cho trước là 8 thì kết quả đầu ra phải là 40320.

x=int(input("Nhập số cần tính giai thừa:"))

def fact(x):

if x == 0:

return 1

return x \* fact(x - 1)

print (fact(x))

**Bài 9: Cho các số thực A, B, C và D. Bạn hãy tính tích của 4 số để biết đó là số âm, số dương hay bằng 0.**

**\* Lưu ý: 1 là số dương, 0 là bằng 0 và – 1 là số âm.**

a = float(input('Nhập só thực A: '))

b = float(input('Nhập số thực B: '))

c = float(input('Nhập số thực C: '))

d = float(input('Nhập số thực D: '))

Tich = a \* b \* c \* d

if (a == 0) or (b == 0) or (c == 0) or (d == 0) :

print (0)

elif Tich > 0:

print(1)

else:

print(-1)

**Bài 10: Viết phương trình giải phương trình bậc hai: ax2 + b x + c = 0 (Có sử dụng khai báo hàm def)**

import math

def giaiPTBac2(a, b, c):

if (a == 0):

if (b == 0):

print ("Phương trình vô nghiệm");

else:

print ("Phương trình có một nghiệm: x = ", + (-c / b));

return;

delta = b \* b - 4 \* a \* c;

if (delta > 0):

x1 = (float)((-b + math.sqrt(delta)) / (2 \* a));

x2 = (float)((-b - math.sqrt(delta)) / (2 \* a));

print ("Phương trình có 2 nghiệm là: x1 = ", x1, " và x2 = ", x2);

elif (delta == 0):

x1 = (-b / (2 \* a));

print("Phương trình có nghiệm kép: x1 = x2 = ", x1);

else:

print("Phương trình vô nghiệm");

a = float(input("a = "));

b = float(input("b = "));

c = float(input("c = "));

giaiPTBac2(a, b, c)

**Bài 11: Nhập tên và số tiền của nhân viên đi làm theo tháng, biết:**

**Nếu đi làm dưới 2 tháng trong năm: 500000 VND**

**Nếu đi làm trên 2 tháng trong năm: Số tháng x 250000VND**

a = input ("Nhập tên của nhân viên: ")

b = int (input("Nhập số tháng đi làm: "))

if b <= 2 :

print ("Tiền lương của", a , "là: 500000VND" )

else:

x1 = b \* 250000

print ("Tiền lương của", a , "là:", x1, "VND")

**Bài 12:** Viết chương trình tính chu vi, diện tích tam giác a,b,c (với a,b,c>0) (Có thể có or không)

import math

a = float(input("Nhập giá trị a: "))

b = float(input("Nhập giá trị b: "))

c = float(input("Nhập giá trị c: "))

if (a+b>c) and (b+c>a) and (c+a>b):

print("Tạo thành tam giác")

CV = (a+b+c)

print("Chu vi hình tam giác : ", CV)

p = (a+b+c)/2

DT = math.sqrt(p\*(p-a)\*(p-b)\*(p-c))

print("Diện tích tam giác là: ", DT)

else:

print("Không tạo thành tam giác")

**DẠNG 3: Vòng lặp for**

**Bài 13:** Viết chương trình vẽ một tam giác vuông cân (Hai cạnh góc vuông là số tự nhiên)

n = 6;

print("Ve tam giac sao vuong can: ")

for i in range(1, n + 1):

for j in range(1, i + 1):

print(".", end = " ");

print()

**Bài 14:** Viết chương trình chứa các key từ 1 đến một số n thuộc N và các giá trị chính phương của chúng.

n = int(input("Nhập số tự nhiên n: "))

def printDict():

d=dict()

for i in range(1,n+1):

d[i]=i\*\*2

print (d)

printDict()

**Bài 15: Nhập số tiền lãi qua từng tháng thông qua số tháng, lãi suất và số tiền, từ đó tính tiền qua từng tháng.**

a = int (input ("Nhập vào số tháng: "))

b = float (input("Nhập vào lãi suất: "))

c = int (input ("Nhập vào số tiền: "))

for i in range (1,a + 1):

lai = ((c\*(b/12))/100)

c += lai

print (" Tiền lãi tháng", i , "la: ", lai)

print (" Số tiền tháng" , i , "là: ", c)

**DẠNG 4: Vòng lặp While**

**Bài 16: Nhập n để tính tổng dãy số đã cho: S = 1 + 2 + 3 + …. + n**

n = int(input("Nhập số nguyên dương n: "))

s = 0

i = 1

while i <= n :

s+= i;

i+=1;

print("Tổng dãy số từ 1 đến", n," là:", s)

**Bài 17: Tính chuỗi dãy số sau: S =**

n = int(input("Nhập vào số n: "))

sum = 0

for i in range (1, n+1):

sum += 1/i

print ("Tổng của 1 + 1/2 + 1/3 + .... + 1/n là: ", sum)

**Bài 18: Nhập các số cần thiết để nhập các giá trị của phương trình ax + b (Có sử dụng khai báo hàm def)**

def TOAN(a,b):

while True:

x = float(input("Nhập giá trị của x (nhập -1 để kết thúc): "))

if x == -1:

break

result = a \* x + b

print("Kết quả của phương trình" ,a,"x +",b, "là: ", result)

a = float(input("Nhập giá trị của a: "))

b = float(input("Nhập giá trị của b: "))

TOAN(a,b)

**DẠNG 5: LIST\_ ARRAY**

**Bài 19:** Viết chương trình chấp nhận một chuỗi số, phân tách bằng dấu phẩy từ giao diện điều khiển, tạo ra một danh sách và một tuple chứa mọi số.

VD: 12,35,56,32,80,9,6,15.

values=input("Nhập vào các giá trị:")

l=values.split(",")

t=tuple(l)

print (l)

print (t)

**Bài 20:** Viết chương trình sắp xếp các chữ cái trong tệp. VD: abcdefg

py\_list = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

print(sorted(py\_list))

py\_string = 'abcdefg'

print(sorted(py\_string))

py\_tuple = ('a', 'b', 'c', 'd', 'e', 'f', 'g')

print(sorted(py\_tuple))

**Bài 21:** Viết một chương trình có hộp dữ liệu có nhiều keyword. Sắp xếp các số theo giá trị nhỏ dần

**C1: Dùng sorted**

participant\_list = [

('Lê An', 50, 18),

('Trần Bình', 75, 12),

('Trần Tâm', 75, 20),

('Thanh Lam', 90, 22),

('Vũ Lan', 45, 12)

]

sorted\_list = sorted(participant\_list, key=lambda item: (100-item[1], item[2]))

print(sorted\_list)

**C2: Dùng hàm def**

participant\_list = [

('Lê An', 50, 18),

('Trần Bình', 75, 12),

('Trần Tâm', 75, 20),

('Thanh Lam', 90, 22),

('Vũ Lan', 45, 12)

]

def sorter(item):

error = 100 - item[1]

age = item[2]

return (error, age)

sorted\_list = sorted(participant\_list, key=sorter)

print(sorted\_list)

**DẠNG 6: DẠNG KẾT HỢP**

**Bài 22:** Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 3, nằm trong đoạn 1000 và 2400 (tính cả 1000 và 2400). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

j=[]

for i in range(1000, 2401):

if (i%7==0) and (i%3!=0):

j.append(str(i))

print (','.join(j))

**Bài 23:** 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.

def check\_divisible\_by\_5(binary\_numbers):

value = []

items = binary\_numbers.split(',')

for p in items:

decimal\_number = int(p, 2)

if decimal\_number % 5 != 0:

value.append(p)

return ",".join(value)

binary\_numbers = input("Nhập chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy: ")

print(check\_divisible\_by\_5(binary\_numbers))

**Bài 24:** Xác định một class với generator có thể lặp lại các số nằm trong khoảng 0 và n, và chia hết cho 9.

def putNumbers(n):

i = 0

while i<n:

j=i

i=i+1

if j%9==0:

yield j

for i in putNumbers (100):

print (i)

**Bài 25: Viết chương trình nhập vào 1 dãy số nguyên (n: nhập vào). Viết chương trình KT số hoàn hảo trong dãy vừa nhập.**

n = int(input("Nhập số nguyên: "))

tong = 0

for i in range(1, n):

if (n % i == 0):

tong += i

if (tong == n):

print(n, "là số hoàn hảo")

else:

print(n, "không phải là số hoàn hảo")

**Bài 26: Xác định một class với generator có thể lặp lại các số nằm trong khoảng 0 và n, và chia hết cho 7.**

**Gợi ý:**

**Sử dụng yield.**

def NUMBERs (n):

i = 0

while i < n:

j = i

i = i + 1

if j% 7 == 0:

yield j

for i in NUMBERs (100):

print (i)

**Bài 27a: Định nghĩa một class có tên là Shape và class con là Square. Square có hàm init để lấy đối số là chiều dài. Cả 2 class đều có hàm area để in diện tích của hình, diện tích mặc định của Shape là 0.**

class Shape(object):

def \_\_init\_\_(self):

pass

def area(self):

return 0

class Square(Shape):

def \_\_init\_\_(self, l):

Shape.\_\_init\_\_(self)

self.length = l

def area(self):

return self.length\*self.length

aSquare= Square(3)

print (aSquare.area())

**Câu 27b: Dưới dạng class có 2 kiểu dữ liệu là tên và lớp. Từ nguồn dữ liệu ban đầu là ‘An’ và ‘12’ hãy nhập dữ liệu và điền thông tin của chúng.**

**class AB:**

**def \_\_init\_\_ (self, aten,alop):**

**self.ten = aten**

**self.lop = alop**

**AB\_A = AB('An','12')**

**print('Ten cua A la: ', AB\_A.ten)**

**print('Lop: ', AB\_A.lop)**

**Bài 28: Nhập tên các con vật rồi sắp xếp chúng theo Alphabet chữ cái đầu của mỗi từ.**

Ten = input("Nhập tên các con vật, phân tách bởi dấu phẩy: ")

Ten\_list = Ten.split(",")

Ten\_list.sort(key=lambda x: x.split()[0])

print("Các con vật theo thứ tự Alphabet chữ cái đầu tiên của mỗi từ là: ")

for name in Ten\_list:

print(name)

**Bài 29:** **Lập list để tạo ra dãy fibonacci.**

**VD: Nhập số: 5 🡪 KQ: [1;1;2;3;5]**

def fibonacci():

num = int(input("How many numbers that generates?: "))

i = 1

if num < 0:

print("Please enter a non-negative number.")

exit(0)

elif num == 0:

fib = []

elif num == 1:

fib = [1]

elif num == 2:

fib = [1,1]

else:

fib = [1,1]

while i < (num - 1):

fib.append(fib[i] + fib[i-1])

i += 1

return fib

print(fibonacci())

input()

**Bài 30.1:** **(Chương trình chạy trong máy bán hàng) Một dạng list kép gồm có món đồ cần mua và giá như sau:**

|  |  |
| --- | --- |
| **Chuối** | **5.000** |
| **Xoài** | **20.000** |
| **Táo** | **15.000** |
| **Đu đủ** | **25.000** |
| **Ổi** | **15.000** |

**Từ đó người dùng nhập món đồ từ món hàng trên và tính tổng món hàng đã nhập.**

price\_list = [

["01", "Chuối", 5000, "cái"],

["02", "Xoài", 20000, "kg"],

["03", "Táo", 15000, "kg"],

["04", "Đu đủ", 25000, "cái"],

["05", "Ổi", 15000, "kg"]

]

def find\_price(item\_name):

for item in price\_list:

if item[1] == item\_name:

return item[2]

return None

total\_price = 0

while True:

item\_name = input("Nhập tên món đồ (nhập 'x' để kết thúc): ")

if item\_name == 'x':

break

price = find\_price(item\_name)

if price is None:

print("Món đồ bạn nhập không có trong danh sách.")

else:

print("Giá của" ,item\_name, "là" ,price, "đồng.")

total\_price += price

print("Tổng giá trị của các món hàng đã nhập là" ,total\_price, "đồng.")

**Bài 30.2:** **Một dạng list chạy d.sách mảng 2 chiều gồm có món đồ cần mua và giá như sau:**

|  |  |  |
| --- | --- | --- |
| **Chuối** | **5.000** | **2023-05-01** |
| **Xoài** | **20.000** | **2023-05-06** |
| **Táo** | **15.000** | **2023-06-03** |
| **Đu đủ** | **25.000** | **2023-08-04** |
| **Ổi** | **15.000** | **2023-09-12** |

**Từ đó người dùng nhập món đồ cần mua và số lượng mua món hàng đó, từ món hàng trên hãy tính tổng số lượng món đã mua và tính tổng tất cả món hàng đã nhập dựa trên số lượng.**

items = [

["Chuối", 5000, "2023-05-01"],

["Xoài", 20000, "2023-05-06"],

["Táo", 15000, "2023-06-03"],

["Đu đủ", 25000, "2023-08-04"],

["Ổi", 15000, "2023-09-12"]

]

total\_quantity = 0

total\_price = 0

while True:

item\_name = input("Nhập tên món hàng (nhập 'x' để kết thúc): ")

if item\_name == 'x':

break

quantity = int(input("Nhập số lượng cần mua: "))

for item in items:

if item[0] == item\_name:

print("Ngày nhập: ",item[2], ",Giá tiền: ",item[1], "đồng.")

total\_quantity += quantity

total\_price += quantity \* item[1]

print("Tổng số lượng của các món hàng đã nhập là: ",total\_quantity)

print("Tổng giá trị của các món hàng đã nhập là: " ,total\_price, "đồng.")

**Bài 31:** **Cho ba số a, b, c là các số tự nhiên, hãy dùng vòng lặp để tìm các ước chung của nó.**

**Yêu cầu:**

**1. Tính tổng các ước chung của cả 3 số.**

**2. Xem thử tổng đó có chia hết cho 2 không?**

a = int(input("Nhập số tự nhiên thứ nhất: "))

b = int(input("Nhập số tự nhiên thứ hai: "))

c = int(input("Nhập số tự nhiên thứ ba: "))

UC = 1

for i in range(1, int(min(a, b, c))+1):

if a % i == 0 and b % i == 0 and c % i == 0:

UC = i

Sum = sum(set([i for i in range(1, int(max(a, b, c))+1) if a % i == 0 and b % i == 0 and c % i == 0]))

if Sum % 2 == 0:

print("Tổng các ước của " ,a,"",b, "và" ,c, "là" ,Sum, "và chia hết cho 2.")

else:

print("Tổng các ước của " ,a,"",b, "và" ,c, "là" ,Sum, "và không chia hết cho 2.")

**Bài 32:** Viết chương trình và in giá trị theo công thức cho trước: q = √([(2 \* c \* d)/h])

(trong đó: q bằng căn bậc hai của [(2 nhân c nhân d) chia h]. Với giá trị cố định của c là 30, h là 20. d là dãy giá trị tùy biến, được nhập vào từ giao diện người dùng, các giá trị của d được phân cách bằng dấu phẩy.)

Giả sử chuỗi giá trị của d nhập vào là 100,150,180 thì đầu ra sẽ là 17,21,23.

import math

c = 30

h = 20

value = []

items=[x for x in input("Nhập giá trị của d: ").split(',')]

for d in items:

value.append(str(int(round(math.sqrt(2\*c\*float(d)/h)))))

print (','.join(value))

**DẠNG 7: DẠNG MỞ FILE**

**Bài 33:** Dựa vào cấu trúc sau đây:

data/

├── user

│ ├── dir1

│ ├── dir2

│ │ └── name.txt

│ ├── pass.txt

│ ├── direct.txt

│ └── user.txt

├── client

└── move.py

Hãy viết 1 chương trình mở 1 file bài lồng vào nhau, từ đó chạy chương trình để mở vào đúng file bài đã cho.

import pathlib

with open('data/user/dir2/name.txt', 'r') as f:

name = f.read()

print(f"Đọc tên file: {name}")

with open(f"data/user/{name}", 'r') as f2:

content = f2.read()

print(f"Nội dung của file {name}:")

print(content)

**DẠNG 8: Giải thuật và GAME**

**Bài 34: Một bạn cần tạo một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Em hãy viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.**

**Có 6 tiêu chí kiểm tra mật khẩu bao gồm:**

**1. Ít nhất 1 chữ cái nằm trong [a-z]  
2. Ít nhất 1 số nằm trong [0-9]  
3. Ít nhất 1 kí tự nằm trong [A-Z]  
4. Ít nhất 1 ký tự nằm trong [$ # @]  
5. Độ dài mật khẩu tối thiểu: 6  
6. Độ dài mật khẩu tối đa: 30**

**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.**

**Giải**

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))

**Bài 35: (Bài toán thực tế) Bạn hãy viết 1 chương trình gồm có các nội dung sau:**

**1. Nhập dữ liệu vào 1 list thông qua def như sau: Món hàng cần nhập, số lượng tối đa nhập vào, giá tiền.**

**2. Từ list nhập vào, người sử dụng hãy nhập vào các thông tin đã cho như trên theo dạng tổng số lượng cần mua.**

**3. Viết vào chương trình máy để tính tổng số tiền mua các món hàng.**

**\* Lưu ý: Số lượng phải được tính tăng lên thì giá tiền cũng phải tăng lên (không cần hiện tổng số lượng đã mua). Nếu các giá trị từ list nhập vào không trùng khớp với các giá trị đã cho thì sử dụng def để mở rộng hàm.**

resource\_list = []

gias\_list = []

maxs\_list = []

n = int(input("Nhập số tài nguyên: "))

for i in range(n):

resource = input("Nhập tài nguyên: ")

gias = int(input("Nhập giá tiền: "))

maxs = int(input("Nhập số lượng tối đa trong kho: "))

resource\_list.append(resource)

gias\_list.append(gias)

maxs\_list.append(maxs)

print("Danh sách tài nguyên:")

for i in range(len(resource\_list)):

print(f"{i+1}. {resource\_list[i]} ({gias\_list[i]} đồng/{resource\_list[i]}), {maxs\_list[i]} {resource\_list[i]}")

total = 0

while True:

index = int(input("Nhập số của tài nguyên cần lấy (nhập 0 để kết thúc): "))

if index == 0:

break

quantity = int(input("Nhập số lượng cần lấy: "))

if quantity > maxs\_list[index-1]:

print(f"Số lượng tài nguyên còn lại không đủ ({maxs\_list[index-1]} {resource\_list[index-1]}).")

else:

total += quantity \* gias\_list[index-1]

maxs\_list[index-1] -= quantity

print(f"{quantity} {resource\_list[index-1]} đã được lấy.")

print(f"Tổng giá trị các món hàng được lấy ra là {total} đồng.")

**Bài 36: Sử dụng import random để tạo game kéo búa bao.**

import random

def game():

choices = ['kéo', 'búa', 'bao']

player = input("Bạn hãy lựa chọn kéo, búa hoặc bao: ").upper()

computer = random.choice(choices).upper()

print("Bạn đã lựa chọn: ", player)

print("Máy đã lựa chọn: ", computer)

if player == computer:

return "Hòa!"

if (player == "kéo" and computer == "bao") or (player == "búa" and computer == "kéo") or (player == "bao" and computer == "búa"):

return "Bạn đã thắng!"

else:

return "Bạn đã thua!"

print(game())

**Bài 36: Tạo game Flappy bird**

import pygame

import sys

def draw\_floor():

screen.blit(floor\_surface,(floor\_x\_pos,900))

screen.blit(floor\_surface,(floor\_x\_pos + 576,900))

def create\_pipe():

random\_pipe\_pos = random.choice(pipe\_height)

bottom\_pipe = pipe\_surface.get\_rect(midtop = (700,random\_pipe\_pos))

top\_pipe = pipe\_surface.get\_rect(midbottom = (700,random\_pipe\_pos - 300))

return bottom\_pipe,top\_pipe

def move\_pipes(pipes):

for pipe in pipes:

pipe.centerx -= 5

return pipes

def draw\_pipes(pipes):

for pipe in pipes:

if pipe.bottom >= 1024:

screen.blit(pipe\_surface,pipe)

else:

flip\_pipe = pygame.transform.flip(pipe\_surface,False,True)

screen.blit(flip\_pipe,pipe)

def check\_collision(pipes):

for pipe in pipes:

if bird\_rect.colliderect(pipe):

return False

if bird\_rect.top <= -100 or bird\_rect.bottom >= 900:

return False

return True

def rotate\_bird(bird):

new\_bird = pygame.transform.rotozoom(bird,-bird\_movement \* 3,1)

return new\_bird

def bird\_animation():

new\_bird = bird\_frames[bird\_index]

new\_bird\_rect = new\_bird.get\_rect(center = (100,bird\_rect.centery))

return new\_bird,new\_bird\_rect

def score\_display(game\_state):

if game\_state == 'main\_game':

score\_surface = game\_font.render(str(int(score)),True,(255,255,255))

score\_rect = score\_surface.get\_rect(center = (288,100))

screen.blit(score\_surface,score\_rect)

if game\_state == 'game\_over':

score\_surface = game\_font.render(f'Score: {int(score)}' ,True,(255,255,255))

score\_rect = score\_surface.get\_rect(center = (288,100))

screen.blit(score\_surface,score\_rect)

high\_score\_surface = game\_font.render(f'High score: {int(high\_score)}',True,(255,255,255))

high\_score\_rect = high\_score\_surface.get\_rect(center = (288,850))

screen.blit(high\_score\_surface,high\_score\_rect)

def update\_score(score, high\_score):

if score > high\_score:

high\_score = score

return high\_score

pygame.init()

screen = pygame.display.set\_mode((576,1024))

clock = pygame.time.Clock()

game\_font = pygame.font.Font('04B\_19.TTF',40)

# Game Variables

gravity = 0.25

bird\_movement = 0

game\_active = True

score = 0

high\_score = 0

bg\_surface = pygame.image.load('assets/background-day.png').convert()

bg\_surface = pygame.transform.scale2x(bg\_surface)

floor\_surface = pygame.image.load('assets/base.png').convert()

floor\_surface = pygame.transform.scale2x(floor\_surface)

floor\_x\_pos = 0

bird\_downflap = pygame.transform.scale2x(pygame.image.load('assets/bluebird-downflap.png').convert\_alpha())

bird\_midflap = pygame.transform.scale2x(pygame.image.load('assets/bluebird-midflap.png').convert\_alpha())

bird\_upflap = pygame.transform.scale2x(pygame.image.load('assets/bluebird-upflap.png').convert\_alpha())

bird\_frames = [bird\_downflap,bird\_midflap,bird\_upflap]

bird\_index = 0

bird\_surface = bird\_frames[bird\_index]

bird\_rect = bird\_surface.get\_rect(center = (100,512))

BIRDFLAP = pygame.USEREVENT + 1

pygame.time.set\_timer(BIRDFLAP,200)

pipe\_surface = pygame.image.load('assets/pipe-green.png')

pipe\_surface = pygame.transform.scale2x(pipe\_surface)

pipe\_list = []

SPAWNPIPE = pygame.USEREVENT

pygame.time.set\_timer(SPAWNPIPE,1200)

pipe\_height = [400,600,800]

game\_over\_surface = pygame.transform.scale2x(pygame.image.load('assets/message.png').convert\_alpha())

game\_over\_rect = game\_over\_surface.get\_rect(center = (288,512))

flap\_sound = pygame.mixer.Sound('sound/sfx\_wing.wav')

death\_sound = pygame.mixer.Sound('sound/sfx\_hit.wav')

score\_sound = pygame.mixer.Sound('sound/sfx\_point.wav')

score\_sound\_countdown = 100

SCOREEVENT = pygame.USEREVENT + 2

pygame.time.set\_timer(SCOREEVENT,100)

while True:

for event in pygame.event.get():

if event.type == pygame.QUIT:

pygame.quit()

sys.exit()

if event.type == pygame.KEYDOWN:

if event.key == pygame.K\_SPACE and game\_active:

bird\_movement = 0

bird\_movement -= 12

flap\_sound.play()

if event.key == pygame.K\_SPACE and game\_active == False:

game\_active = True

pipe\_list.clear()

bird\_rect.center = (100,512)

bird\_movement = 0

score = 0

if event.type == SPAWNPIPE:

pipe\_list.extend(create\_pipe())

if event.type == BIRDFLAP:

if bird\_index < 2:

bird\_index += 1

else:

bird\_index = 0

bird\_surface,bird\_rect = bird\_animation()

screen.blit(bg\_surface,(0,0))

if game\_active:

# Bird

bird\_movement += gravity

rotated\_bird = rotate\_bird(bird\_surface)

bird\_rect.centery += bird\_movement

screen.blit(rotated\_bird,bird\_rect)

game\_active = check\_collision(pipe\_list)

# Pipes

pipe\_list = move\_pipes(pipe\_list)

draw\_pipes(pipe\_list)

score += 0.01

score\_display('main\_game')

else:

screen.blit(game\_over\_surface,game\_over\_rect)

high\_score = update\_score(score,high\_score)

score\_display('game\_over')

# Floor

floor\_x\_pos -= 1

draw\_floor()

if floor\_x\_pos <= -576:

floor\_x\_pos = 0

pygame.display.update()

clock.tick(120)

**Bài 37: Code cây thông Noel đơn giản**

from turtle import \*

from random import randint

def create\_rectangle(turtle, color, x, y, width, height):

turtle.penup()

turtle.color(color)

turtle.fillcolor(color)

turtle.goto(x, y)

turtle.pendown()

turtle.begin\_fill()

turtle.forward(width)

turtle.left(90)

turtle.forward(height)

turtle.left(90)

turtle.forward(width)

turtle.left(90)

turtle.forward(height)

turtle.left(90)

# fill the above shape

turtle.end\_fill()

# Reset the orientation of the turtle

turtle.setheading(0)

def create\_circle(turtle, x, y, radius, color):

oogway.penup()

oogway.color(color)

oogway.fillcolor(color)

oogway.goto(x, y)

oogway.pendown()

oogway.begin\_fill()

oogway.circle(radius)

oogway.end\_fill()

BG\_COLOR = "#0080ff"

oogway = Turtle()

# set turtle speed

oogway.speed(2)

screen = oogway.getscreen()

# set background color

screen.bgcolor(BG\_COLOR)

# set tile of screen

screen.title("Merry Christmas")

# maximize the screen

screen.setup(width=1.0, height=1.0)

y = -100

# create tree trunk

create\_rectangle(oogway, "red", -15, y-60, 30, 60)

# create tree

width = 240

oogway.speed(10)

while width > 10:

width = width - 10

height = 10

x = 0 - width/2

create\_rectangle(oogway, "green", x, y, width, height)

y = y + height

# create a star a top of tree

oogway.speed(1)

oogway.penup()

oogway.color('yellow')

oogway.goto(-20, y+10)

oogway.begin\_fill()

oogway.pendown()

for i in range(5):

oogway.forward(40)

oogway.right(144)

oogway.end\_fill()

tree\_height = y + 40

# create moon in sky

# create a full circle

create\_circle(oogway, 230, 180, 60, "white")

# overlap with full circle of BG color to make a crescent shape

create\_circle(oogway, 220, 180, 60, BG\_COLOR)

# now add few stars in sky

oogway.speed(10)

number\_of\_stars = randint(20,30)

# print(number\_of\_stars)

for \_ in range(0,number\_of\_stars):

x\_star = randint(-(screen.window\_width()//2),screen.window\_width()//2)

y\_star = randint(tree\_height, screen.window\_height()//2)

size = randint(5,20)

oogway.penup()

oogway.color('white')

oogway.goto(x\_star, y\_star)

oogway.begin\_fill()

oogway.pendown()

for i in range(5):

oogway.forward(size)

oogway.right(144)

oogway.end\_fill()

# print greeting message

oogway.speed(1)

oogway.penup()

msg = "Merry Christmas"

oogway.goto(0, -200) # y is in minus because tree trunk was below x axis

oogway.color("white")

oogway.pendown()

oogway.write(msg, move=False, align="center", font=("Arial", 15, "bold"))

oogway.hideturtle()

screen.mainloop()

**Bài 38: Tìm d trong chương trình mã RSA, trong đó nười lập trình tự cho số và ra kết quả.**

**def extended\_gcd(a, b):**

**if a == 0:**

**return (b, 0, 1)**

**else:**

**g, x, y = extended\_gcd(b % a, a)**

**return (g, y - (b // a) \* x, x)**

**def modinv(a, m):**

**g, x, y = extended\_gcd(a, m)**

**if g != 1:**

**raise Exception('Modular inverse does not exist')**

**else:**

**return x % m**

**e = 13**

**phi\_n = 2436**

**d = modinv(e, phi\_n)**

**print("d =", d)**