Câu 1: In ra n năm nhuận sau năm 2016

n = int(input())

leap\_years = []

current\_year = 2016

while current\_year % 4 != 0 or (current\_year % 100 == 0 and current\_year % 400 != 0):

current\_year -= 1

while len(leap\_years) < n:

current\_year += 4

if current\_year % 100 == 0 and current\_year % 400 != 0:

current\_year += 4

leap\_years.append(current\_year)

for year in leap\_years:

print(year, end=' ')

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Câu 2: Hệ pt

a1,b1,c1=map(float,input().split())

a2,b2,c2=map(float,input().split())

D=a1\*b2-a2\*b1

Dx=c1\*b2-c2\*b1

Dy=a1\*c2-a2\*c1

if (D!=0):

x=Dx/D

y=Dy/D

print("%.5f %.5f"%(x,y))

elif (Dx!=0): print("VN")

else: print ("VSN")

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Câu 3: Bảng cửu chương

a, b = map(int, input().split())

for n in range(a, b + 1):

for i in range(n, 10):

print("%dx%d=%d"%(n,i,n\*i)) //Bảng cửu chương dài thì thay n=1

Câu 4: Tìm ước số chung lớn nhất của 3 số

a, b, c = map(int, input().split())

if a > 0 and b > 0 and c > 0:

gcd\_ab = 1

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

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

gcd\_ab = i

gcd\_abc = 1

for i in range(1, min(gcd\_ab, c) + 1):

if gcd\_ab % i == 0 and c % i == 0:

gcd\_abc = i

print("%d"%(gcd\_abc))

else:

print("DL sai.")

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Câu 5: Cặp số thân thiết

n,m=map(int,input().split())

tonguocn=0

tonguocm=0

for i in range (1,n):

if(n%i==0):

tonguocn+=i

for j in range (1,m):

if(m%j==0):

tonguocm+=j

if(tonguocn==m and tonguocm==n):

print("true")

else:

print("false")

Câu 6: Pt bậc 2

import math

a,b,c=map(float,input().split())

if(a==0):

if(b==0):

if(c==0):

print("VSN")

else: print("VN")

else: print("%.3f"%(-c/b))

else:

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

if(delta<0): print("VN")

elif(delta==0): print("%.3f"%(-b/(2\*a)))

else:

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

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

if(x1<x2):

print("%.3f %.3f"%(x1,x2))

elif(x1>x2):

print("%.3f %.3f"%(x2,x1))

Câu 7: Đếm số lượng và tính tổng các số lẻ đoạn a b b a

a,b=map(int,input().split())

dem=0

tong=0

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

if i%2==1:

dem+=1

tong+=i

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

if i%2==1:

dem+=1

tong+=i

print("%d %d"%(dem,tong))

Câu 8: Tính tbc các số chẵn trong đoạn a b b a

a,b=map(int,input().split())

tbc=0

tong=0

dem=0

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

if(i%2==0):

dem+=1

tong+=i

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

if(i%2==0):

dem+=1

tong+=i

print("%.2f"%(tong/dem))

Câu 9: Vẽ tam giác ngôi sao (dễ)

n = int(input())

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

print(" " \* (n - i), end="")

print("\* " \* i)

Câu 10: Vẽ hình vuông rỗng

def draw\_square(h):

for i in range(1, abs(h) + 1):

for j in range(1, abs(h) + 1):

if i == 1 or j == 1 or i == abs(h) or j == abs(h):

print("\*", end="")

else:

print(" ", end="")

print()

x, y, z = map(int, input().split())

print(x)

draw\_square(x)

print(y)

draw\_square(y)

print(z)

draw\_square(z)

Câu 11: Vẽ tam giác ngôi sao (khó)

def draw\_triangle(height):

if height > 0:

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

spaces = height - i

stars = 2 \* i - 1

print(" " \* spaces + "\*" \* stars)

elif height < 0:

for i in range(abs(height), 0, -1):

spaces = abs(height) - i

stars = 2 \* i - 1

print(" " \* spaces + "\*" \* stars)

def main():

x, y, z = map(int, input().split())

print(x)

draw\_triangle(x)

print(y)

draw\_triangle(y)

print(z)

draw\_triangle(z)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 12: Chuyển hệ 10 sang 2

def doi\_10\_sang\_2(n):

if n <= 0:

return

else:

doi\_10\_sang\_2(n // 2)

print(n % 2, end=" ")

def in\_hang(a, b):

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

print("%d: "%i, end="")

doi\_10\_sang\_2(i)

print()

def main():

a, b = map(int, input().split())

if a > b:

a, b = b, a

in\_hang(a, b)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 13: Fibonanci

n = int(input())

t1, t2 = 1, 1

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

print(t1, end=" ")

next = t1 + t2

t1, t2 = t2, next

Câu 14: Tìm chữ số lớn nhất của số nguyên dương n

def timchuso(n):

if n<0:

return "Error"

str\_n=str(n)

so\_max=max(map(int,str\_n))

return so\_max

x,y,z=map(int,input().split())

print(timchuso(x), timchuso(y), timchuso(z))

Câu 15: Căn 2+ căn 2+…

import math

n=int(input())

if(n<0):

print("Error.")

else:

s=0

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

s=math.sqrt(2+s)

print("%.2f"%s)

Câu 16: Kiểm tra số hoàn hảo

n=int(input())

tong=0

for i in range (1,n):

if(n%i==0):

tong+=i

if(tong==n):

print("%d la so hoan hao"%n)

else:

print("%d khong la so hoan hao"%n)

Câu 17: Tính tbc chẵn lẻ trong a b x y

a,b=map(int,input().split())

x,y=map(int,input().split())

dem1=0

dem2=0

tbc1=0

tbc2=0

tong1=0

tong2=0

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

if i%2==0:

dem1+=1

tong1+=i

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

if i%2==0:

dem1+=1

tong1+=i

if(dem1==0):

print("NO")

else: print("%.2f"%(tong1/dem1))

for j in range (x,y+1):

if j%2==1:

dem2+=1

tong2+=j

for j in range (y,x+1):

if j%2==1:

dem2+=1

tong2+=j

if(dem2==0):

print("NO")

else: print("%.2f"%(tong2/dem2))

Câu 18: 3 điểm thẳng hàng

x1, y1 = map(float, input().split())

x2, y2 = map(float, input().split())

x3, y3 = map(float, input().split())

x4, y4 = map(float, input().split())

x5, y5 = map(float, input().split())

x6, y6 = map(float, input().split())

result1 = "YES" if (y2 - y1) \* (x3 - x1) == (y3 - y1) \* (x2 - x1) else "NO"

result2 = "YES" if (y5 - y4) \* (x6 - x4) == (y6 - y4) \* (x5 - x4) else "NO"

print(result1, result2)

Câu 19: In ra so ngto trong doan ab ba

a, b = map(int, input().split())

if a > b:

a, b = b, a

found\_prime = False

for n in range(a, b + 1):

dem = 0

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

if n % i == 0:

dem += 1

if dem == 2:

found\_prime = True

print(n, end=' ')

if not found\_prime:

print("Khong co")

Câu 20: Số chính phương

a = int(input())

if a < 0:

print("False")

else:

square\_root = int(a\*\*0.5)

if square\_root\*\*2 == a:

print("True")

else:

print("False")

Câu 21: Câu đối xứng

def is\_palindrome(sentence):

cleaned\_sentence = ''.join(c.lower() for c in sentence if c.isalnum())

return cleaned\_sentence == cleaned\_sentence[::-1]

s1 = input()

result\_s1 = is\_palindrome(s1)

s2 = input()

print("true" if result\_s1 else "false")

result\_s2 = is\_palindrome(s2)

print("true" if result\_s2 else "false")

Câu 22: Tìm số lớn nhất trong dãy chia hết cho b

a = list(map(int, input().split()))

b = int(input())

a.sort(reverse=True)

max\_multiple = 0

for num in a:

if num % b == 0:

max\_multiple = num

break

print("%d"%(max\_multiple))

Câu 23: In ra hình hcn có thể tích max

n = int(input())

max\_thetich = 0

max\_box = []

for \_ in range(n):

canh = list(map(float, input().split()))

thetich = canh[0] \* canh[1] \* canh[2]

if thetich > max\_thetich:

max\_thetich = thetich

max\_box = canh

print("%.1f %.1f %.1f %.1f" % (max\_box[0], max\_box[1], max\_box[2], max\_thetich))

Câu 24: Sxep ma trận

def sort\_alternating(matrix, order):

for i in range(len(matrix)):

if i % 2 == 1:

matrix[i].sort(reverse=order)

return matrix

m, n = map(int, input().split())

matrix = [list(map(int, input().split())) for \_ in range(m)]

ascending\_order = int(input())

sorted\_matrix = sort\_alternating(matrix, ascending\_order)

for row in sorted\_matrix:

print(\*row)

Câu 25: In ra số phần tử lớn hơn tbc trong mảng

n = int(input())

a = list(map(int, input().split()))

sum\_a = sum(a)

avg = sum\_a / n

count = 0

for i in range(n):

if a[i] >= avg:

count += 1

print(a[i], end=' ')

print("\nSo phan tu thoa man:", count)

Câu 26: In ra vị trí của phần tử lớn nhất trong ma trận. Nếu có nhiều phần tử lớn nhất thì in ra phần tử có chỉ số hàng, chỉ số cột nhỏ nhất.

def nhap\_ma\_tran(n, m):

ma\_tran = []

for i in range(n):

dong = list(map(float, input().split()))

ma\_tran.append(dong)

return ma\_tran

def tim\_phan\_tu\_lon\_nhat(ma\_tran, n, m):

max\_val = float('-inf')

hang\_max, cot\_max = -1, -1

for i in range(n):

for j in range(m):

if ma\_tran[i][j] > max\_val:

max\_val = ma\_tran[i][j]

hang\_max, cot\_max = i, j

return hang\_max, cot\_max, max\_val

def main():

n, m = map(int, input().split())

ma\_tran = nhap\_ma\_tran(n, m)

hang\_max, cot\_max, max\_val = tim\_phan\_tu\_lon\_nhat(ma\_tran, n, m)

print("Phan tu lon nhat co chi so hang %d, chi so cot %d, gia tri %.2f"%(hang\_max,cot\_max,max\_val))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 27: Xoá 1 kí tự trong chuỗi

def xoa\_ky\_tu(chuoi, vi\_tri):

return chuoi[:vi\_tri] + chuoi[vi\_tri+1:]

def main():

chuoi\_X = input()

vi\_tri\_X = int(input())

chuoi\_Y = input()

vi\_tri\_Y = int(input())

ket\_qua\_X = xoa\_ky\_tu(chuoi\_X, vi\_tri\_X)

ket\_qua\_Y = xoa\_ky\_tu(chuoi\_Y, vi\_tri\_Y)

print(ket\_qua\_X)

print(ket\_qua\_Y)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 28: Phần tử đặc biệt trong một ma trận

n, m = map(int, input().split())

matrix = [list(map(int, input().split())) for \_ in range(n)]

special\_elements = set(matrix[0])

for i in range(1, n):

special\_elements.intersection\_update(set(matrix[i]))

if special\_elements:

print(" ".join(map(str, sorted(special\_elements))))

else:

print("Khong co phan tu dac biet")

Câu 29: Chuẩn hoá tên

def bien\_doi(s):

s = list(s.lower().strip())

s[0] = s[0].upper()

i = 1

while i < len(s):

if s[i] == ' ' and i + 1 < len(s):

s[i + 1] = s[i + 1].upper()

i += 1

return ''.join(s)

def main():

S = input()

S\_transformed = bien\_doi(S)

print(S\_transformed)

if \_\_name\_\_ == "\_\_main\_\_":

main()thê

Câu 30: Nhân 2 ma trận

n1, m1 = map(int, input().split())

a1 = [list(map(int, input().split())) for \_ in range(n1)]

n2, m2 = map(int, input().split())

a2 = [list(map(int, input().split())) for \_ in range(n2)]

if m1 != n2:

print("Du lieu vao sai")

else:

nhan = [[0 for \_ in range(m2)] for \_ in range(n1)]

for i in range(n1):

for j in range(m2):

tich = 0

for k in range(m1):

tich += a1[i][k] \* a2[k][j]

nhan[i][j] = tich

print("Ma tran tich")

for i in range(n1):

for j in range(m2):

print(nhan[i][j], end=" ")

print()

Câu 31: Sxep từng hàng của ma trận tăng dần

n, m = map(int, input().split())

a = [list(map(int, input().split())) for \_ in range(n)]

for i in range(n):

for j in range(m - 1):

for k in range(j + 1, m):

if a[i][j] > a[i][k]:

a[i][j], a[i][k] = a[i][k], a[i][j]

print("%d %d"%(n,m))

for i in range(n):

for j in range(m):

print(a[i][j], end=" ")

print()

Câu 32: Xâu con dài nhất

s = input()

n = len(s)

start, max\_len = 0, 1

for i in range(1, n):

left, right = i - 1, i + 1

while left >= 0 and right < n and s[left] == s[right]:

if right - left + 1 > max\_len:

start = left

max\_len = right - left + 1

left -= 1

right += 1

# Kiểm tra xâu đối xứng với độ dài là chẵn

left, right = i - 1, i

while left >= 0 and right < n and s[left] == s[right]:

if right - left + 1 > max\_len:

start = left

max\_len = right - left + 1

left -= 1

right += 1

result = s[start:start + max\_len]

print(result)

Câu 33: Sxep daỹ số cả âm cả dương

vao=input().split()

a=[int(num)for num in vao]

num1=[]

num2=[]

for i in range (len(a)):

if a[i]<0:

num1.append(a[i])

else:

num2.append(a[i])

num1.sort(reverse=True)

num2.sort()

a=num1+num2

for i in range (len(a)):

print(a[i],end=" ")

Câu 34: x op y

def is\_integer(s):

try:

int(s)

return True

except ValueError:

return False

def calculate\_expression(expression):

operators = "+-\*/%"

x = None

op = None

y = None

for i in range(len(expression)):

if expression[i] in operators:

x = expression[:i]

op = expression[i]

y = expression[i+1:]

break

if is\_integer(x) and is\_integer(y):

x = int(x)

y = int(y)

if op == '+':

result = x + y

elif op == '-':

result = x - y

elif op == '\*':

result = x \* y

elif op == '/':

if y == 0:

return "error"

result = x / y

elif op == '%':

if y == 0:

return "error"

result = x % y

return '{:.2f}'.format(result)

else:

return "error"

expression = input()

result = calculate\_expression(expression)

print(result)

Câu 35: In ra các số nguyên tố trong a b b a, có in ngược

def is\_prime(n):

if n <= 1:

return False

if n <= 3:

return True

if n % 2 == 0 or n % 3 == 0:

return False

i = 5

while i \* i <= n:

if n % i == 0 or n % (i + 2) == 0:

return False

i += 6

return True

a, b = map(int, input().split())

prime\_numbers = []

if a <= b:

for num in range(a, b + 1):

if is\_prime(num):

prime\_numbers.append(str(num))

else:

for num in range(a ,b-1,-1):

if is\_prime(num):

prime\_numbers.append(str(num))

if not prime\_numbers:

print("Khong co")

else:

print(" ".join(prime\_numbers))

Câu 36: Tính tiền trả sách cho thư viện

from datetime import date

def calculate\_late\_fee(return\_date, due\_date):

return\_day, return\_month, return\_year = map(int, return\_date.split())

due\_day, due\_month, due\_year = map(int, due\_date.split())

return\_date\_obj = date(return\_year, return\_month, return\_day)

due\_date\_obj = date(due\_year, due\_month, due\_day)

if return\_date\_obj <= due\_date\_obj:

return 0

days\_late = (return\_date\_obj - due\_date\_obj).days

months\_late = return\_date\_obj.month - due\_date\_obj.month + 12 \* (return\_date\_obj.year - due\_date\_obj.year)

years\_late = return\_date\_obj.year - due\_date\_obj.year

if years\_late > 0:

return 1000000

elif months\_late > 0:

return 100000 \* months\_late

else:

return 5000 \* days\_late

return\_date = input().strip()

due\_date = input().strip()

late\_fee = calculate\_late\_fee(return\_date, due\_date)

print(late\_fee)

Câu 37: Caesar

chuoimatma=str(input())

k=int(input())

chuoi=str(input())

n=int(input())

if(n==0):

for c in chuoi:

a=chuoimatma.find(c)

if(a<(len(chuoimatma)-k-1)):

c=chuoimatma[a+k]

else:

c=chuoimatma[-len(chuoimatma)+a+k]

print(c,end='')

else:

for c in chuoi:

a=chuoimatma.find(c)

if(a>(-len(chuoimatma)+k+1)):

c=chuoimatma[a-k]

else:

c=chuoimatma[+len(chuoimatma)-a-k]

print(c,end='')

Câu 38: Viết hàm nhận đầu vào là một mảng các số thực có n phần tử và trả về vị trí của số dương đầu tiên tìm được

def tim\_so\_duong\_dau\_tien(arr):

for i in range(len(arr)):

if arr[i] > 0:

return i, arr[i]

return -1, None

N = int(input())

X = list(map(float, input().split()))

M = int(input())

Y = list(map(float, input().split()))

vi\_tri\_X, gia\_tri\_X = tim\_so\_duong\_dau\_tien(X)

vi\_tri\_Y, gia\_tri\_Y = tim\_so\_duong\_dau\_tien(Y)

if vi\_tri\_X != -1:

print("%d %.3f"%(vi\_tri\_X,gia\_tri\_X))

else:

print("Khong co so duong trong mang.")

if vi\_tri\_Y != -1:

print("%d %.3f"%(vi\_tri\_Y,gia\_tri\_Y))

else:

print("Khong co so duong trong mang.")

Câu 39: Tìm số lẻ nhỏ nhất trong mảng

def tim\_so\_le\_nho\_nhat(arr):

vi\_tri = -1

gia\_tri = float('inf')

for i, num in enumerate(arr):

if num % 2 != 0 and num < gia\_tri:

vi\_tri = i

gia\_tri = num

return vi\_tri, gia\_tri

N = int(input())

mang = list(map(int, input().split()))

vi\_tri, gia\_tri = tim\_so\_le\_nho\_nhat(mang)

if vi\_tri != -1:

print("So le nho nhat co vi tri %d gia tri %d"%(vi\_tri,gia\_tri))

else:

print("Khong co so le trong mang")

Câu 40: Số mạnh mẽ

import math

def isStrongNumber(n):

# Kiểm tra nếu n <= 1 thì trả về False

if n <= 1:

return "false"

maxFactor = int(math.sqrt(n))

for i in range(2, maxFactor+1):

if n % i == 0:

if isPrime(i):

if n % (i\*i) == 0:

return "true"

return "false"

def isPrime(x):

if x < 2:

return "false"

for i in range(2, int(math.sqrt(x))+1):

if x % i == 0:

return "false"

return "true"

n = int(input())

print(isStrongNumber(n))

Câu 41: Ma trận đối xứng

def check(matrix, size):

for i in range(size):

for j in range(size):

if matrix[i][j] != matrix[j][i]:

return False

return True

def main():

n = int(input())

matrix\_a = [list(map(int, input().split())) for \_ in range(n)]

m = int(input())

matrix\_b = [list(map(int, input().split())) for \_ in range(m)]

print("Ma tran 1:", end=" ")

if check(matrix\_a, n):

print("Co doi xung")

else:

print("Khong doi xung")

print("Ma tran 2:", end=" ")

if check(matrix\_b, m):

print("Co doi xung")

else:

print("Khong doi xung")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 42: Tổng 2 ma trận

n1, m1 = map(int, input().split())

a = [list(map(int, input().split())) for \_ in range(n1)]

n2, m2 = map(int, input().split())

b = [list(map(int, input().split())) for \_ in range(n2)]

if n1 != n2 or m1 != m2:

print("Du lieu vao sai")

else:

c = [[0 for \_ in range(m1)] for \_ in range(n1)]

for i in range(n1):

for j in range(m1):

c[i][j] = a[i][j] + b[i][j]

print("Ma tran tong")

for i in range(n1):

for j in range(m1):

print(c[i][j], end=" ")

print()

Câu 43: Mảng tăng dần

def main():

N = int(input())

arr = list(map(int, input().split()))

arr.sort()

print(N)

print(\*arr)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 44: Vẽ theo bit

def print\_bitmap(matrix):

for row in matrix:

for value in row:

if value == 0:

print(' ', end=' ')

else:

print('\*', end=' ')

print()

def main():

n, m = map(int, input().split())

matrix = []

for \_ in range(n):

row = list(map(int, input().split()))

matrix.append(row)

print\_bitmap(matrix)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 45: Kiểm tra dãy tăng, không tăng

def daytang(arr):

for i in range(len(arr) - 1):

if arr[i] >= arr[i + 1]:

return False

return True

def main():

n = int(input())

a = list(map(float, input().split()))

m = int(input())

b = list(map(float, input().split()))

dayA = daytang(a)

dayB = daytang(b)

if dayA:

print("TANG")

else:

print("KHONG\_TANG")

if dayB:

print("TANG")

else:

print("KHONG\_TANG")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 46: Điểm nằm trong ngoài hình tròn

import math

class HinhTron:

def \_\_init\_\_(self, ma, tamx, tamy, bk):

self.ma = ma

self.tamx = tamx

self.tamy = tamy

self.bk = bk

def kiem\_tra\_diem(self, x, y):

khoang\_cach = math.sqrt((x - self.tamx)\*\*2 + (y - self.tamy)\*\*2)

if khoang\_cach < self.bk:

return "trong"

elif khoang\_cach == self.bk:

return "tren"

else:

return "ngoai"

def main():

n = int(input())

danh\_sach\_hinh\_tron = []

for \_ in range(n):

thong\_tin\_ht = input().split()

ma = thong\_tin\_ht[0]

tamx = int(thong\_tin\_ht[1])

tamy = int(thong\_tin\_ht[2])

bk = float(thong\_tin\_ht[3])

hinh\_tron = HinhTron(ma, tamx, tamy, bk)

danh\_sach\_hinh\_tron.append(hinh\_tron)

xa, ya, xb, yb = map(int, input().split())

for hinh\_tron in danh\_sach\_hinh\_tron:

ket\_qua\_A = hinh\_tron.kiem\_tra\_diem(xa, ya)

ket\_qua\_B = hinh\_tron.kiem\_tra\_diem(xb, yb)

print("Diem ({}, {}) nam {} hinh tron tam ({}, {}) ban kinh {:.3f}".format(

xa, ya, ket\_qua\_A, hinh\_tron.tamx, hinh\_tron.tamy, hinh\_tron.bk

))

print("Diem ({}, {}) nam {} hinh tron tam ({}, {}) ban kinh {:.3f}".format(

xb, yb, ket\_qua\_B, hinh\_tron.tamx, hinh\_tron.tamy, hinh\_tron.bk

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 47: Công ty HP. Khách hàng phải trả tiền nhiều nhất

class KhachHang:

def \_\_init\_\_(self, ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky):

self.ten = ten

self.ma\_kh = ma\_kh

self.chi\_so\_dau\_ky = chi\_so\_dau\_ky

self.chi\_so\_cuoi\_ky = chi\_so\_cuoi\_ky

def luong\_dien\_tieu\_thu(self):

return self.chi\_so\_cuoi\_ky - self.chi\_so\_dau\_ky

def tinh\_tien\_phai\_tra(self):

luong\_dien = self.luong\_dien\_tieu\_thu()

if luong\_dien <= 100:

return luong\_dien \* 1000

elif luong\_dien <= 200:

return 100 \* 1000 + (luong\_dien - 100) \* 1500

else:

return 100 \* 1000 + 100 \* 1500 + (luong\_dien - 200) \* 2000

def main():

n = int(input())

danh\_sach\_khach\_hang = []

for \_ in range(n):

thong\_tin\_kh = input().split()

ten = thong\_tin\_kh[0]

ma\_kh = int(thong\_tin\_kh[1])

chi\_so\_dau\_ky = int(thong\_tin\_kh[2])

chi\_so\_cuoi\_ky = int(thong\_tin\_kh[3])

khach\_hang = KhachHang(ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky)

danh\_sach\_khach\_hang.append(khach\_hang)

vi\_tri\_max\_tien = danh\_sach\_khach\_hang.index(max(danh\_sach\_khach\_hang, key=lambda x: x.tinh\_tien\_phai\_tra()))

print("Khach hang phai tra tien nhieu nhat: {}".format(vi\_tri\_max\_tien ))

khach\_hang\_max\_tien = danh\_sach\_khach\_hang[vi\_tri\_max\_tien]

print("{} {} {} {} {} {}".format(

khach\_hang\_max\_tien.ma\_kh,

khach\_hang\_max\_tien.ten,

khach\_hang\_max\_tien.chi\_so\_dau\_ky,

khach\_hang\_max\_tien.chi\_so\_cuoi\_ky,

khach\_hang\_max\_tien.luong\_dien\_tieu\_thu(),

khach\_hang\_max\_tien.tinh\_tien\_phai\_tra()

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 48: Nhân viên có lương thấp nhất

class NhanVien:

def \_\_init\_\_(self, ma\_nv, ten, he\_so\_luong, phu\_cap):

self.ma\_nv = ma\_nv

self.ten = ten

self.he\_so\_luong = he\_so\_luong

self.phu\_cap = phu\_cap

def tinh\_luong\_thang(self):

return self.he\_so\_luong \* 2000000 + self.phu\_cap

def tim\_nhan\_vien\_luong\_thap\_nhat(danh\_sach\_nv):

if not danh\_sach\_nv:

return None

luong\_thap\_nhat = danh\_sach\_nv[0].tinh\_luong\_thang()

nhan\_vien\_luong\_thap\_nhat = danh\_sach\_nv[0]

for nv in danh\_sach\_nv[1:]:

luong\_thang = nv.tinh\_luong\_thang()

if luong\_thang < luong\_thap\_nhat:

luong\_thap\_nhat = luong\_thang

nhan\_vien\_luong\_thap\_nhat = nv

return nhan\_vien\_luong\_thap\_nhat

def main():

n = int(input())

danh\_sach\_nv = []

for \_ in range(n):

thong\_tin\_nv = input().split()

ma\_nv = int(thong\_tin\_nv[0])

ten = thong\_tin\_nv[1]

he\_so\_luong = float(thong\_tin\_nv[2])

phu\_cap = int(thong\_tin\_nv[3])

nv = NhanVien(ma\_nv, ten, he\_so\_luong, phu\_cap)

danh\_sach\_nv.append(nv)

nhan\_vien\_luong\_thap\_nhat = tim\_nhan\_vien\_luong\_thap\_nhat(danh\_sach\_nv)

print("Nhan vien co luong thap nhat")

print("{} {} {:.2f} {} {:.2f}".format(nhan\_vien\_luong\_thap\_nhat.ma\_nv,

nhan\_vien\_luong\_thap\_nhat.ten,

nhan\_vien\_luong\_thap\_nhat.he\_so\_luong,

nhan\_vien\_luong\_thap\_nhat.phu\_cap,

nhan\_vien\_luong\_thap\_nhat.tinh\_luong\_thang()))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 49: Công ty HP 3 loại khách hàng (Doanh nghiệp)

class KhachHang:

def \_\_init\_\_(self, ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky):

self.ten = ten

self.ma\_kh = ma\_kh

self.chi\_so\_dau\_ky = chi\_so\_dau\_ky

self.chi\_so\_cuoi\_ky = chi\_so\_cuoi\_ky

def luong\_dien\_tieu\_thu(self):

return self.chi\_so\_cuoi\_ky - self.chi\_so\_dau\_ky

def tinh\_tien\_phai\_tra(self):

don\_gia\_thuong = 1000

if self.ma\_kh[0] == 'D':

don\_gia\_thuong \*= 1.05

elif self.ma\_kh[0] == 'K':

don\_gia\_thuong \*= 1.07

return int(self.luong\_dien\_tieu\_thu() \* don\_gia\_thuong)

def main():

n = int(input())

danh\_sach\_khach\_hang = []

for \_ in range(n):

thong\_tin\_kh = input().split()

ten = thong\_tin\_kh[0]

ma\_kh = thong\_tin\_kh[1]

chi\_so\_dau\_ky = int(thong\_tin\_kh[2])

chi\_so\_cuoi\_ky = int(thong\_tin\_kh[3])

khach\_hang = KhachHang(ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky)

danh\_sach\_khach\_hang.append(khach\_hang)

khach\_hang\_doanh\_nghiep = [kh for kh in danh\_sach\_khach\_hang if kh.ma\_kh[0] == 'D']

if not khach\_hang\_doanh\_nghiep:

print("Khong co khach hang la doanh nghiep.")

else:

khach\_hang\_doanh\_nghiep.sort(key=lambda x: x.tinh\_tien\_phai\_tra(), reverse=True)

khach\_hang\_max\_tien = khach\_hang\_doanh\_nghiep[0]

print("{} {} {} {} {} {}".format(

khach\_hang\_max\_tien.ma\_kh,

khach\_hang\_max\_tien.ten,

khach\_hang\_max\_tien.chi\_so\_dau\_ky,

khach\_hang\_max\_tien.chi\_so\_cuoi\_ky,

khach\_hang\_max\_tien.luong\_dien\_tieu\_thu(),

khach\_hang\_max\_tien.tinh\_tien\_phai\_tra()

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 50: Giảm dần hình tròn

class HinhTron:

def \_\_init\_\_(self, ma, tamx, tamy, bk):

self.ma = ma

self.tamx = tamx

self.tamy = tamy

self.bk = bk

def dientich(self):

return 3.14159 \* self.bk \* self.bk

def main():

n = int(input())

danh\_sach\_hinh\_tron = []

for \_ in range(n):

thong\_tin\_ht = input().split()

ma = int(thong\_tin\_ht[0])

tamx = int(thong\_tin\_ht[1])

tamy = int(thong\_tin\_ht[2])

bk = float(thong\_tin\_ht[3])

hinh\_tron = HinhTron(ma, tamx, tamy, bk)

danh\_sach\_hinh\_tron.append(hinh\_tron)

danh\_sach\_hinh\_tron.sort(key=lambda x: x.dientich(), reverse=True)

print("Danh sach hinh tron da sap xep: {}".format(n))

for hinh\_tron in danh\_sach\_hinh\_tron:

print("{} {} {} {:.3f}".format(

hinh\_tron.ma,

hinh\_tron.tamx,

hinh\_tron.tamy,

hinh\_tron.bk

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 51: Hình tròn xa gốc toạ độ nhất

class HinhTron:

def \_\_init\_\_(self, ma, tamx, tamy, bk):

self.ma = ma

self.tamx = tamx

self.tamy = tamy

self.bk = bk

def khoang\_cach\_den\_goc(self):

return (self.tamx\*\*2 + self.tamy\*\*2)\*\*0.5

def main():

n = int(input())

danh\_sach\_hinh\_tron = []

for \_ in range(n):

thong\_tin\_ht = input().split()

ma = int(thong\_tin\_ht[0])

tamx = int(thong\_tin\_ht[1])

tamy = int(thong\_tin\_ht[2])

bk = float(thong\_tin\_ht[3])

hinh\_tron = HinhTron(ma, tamx, tamy, bk)

danh\_sach\_hinh\_tron.append(hinh\_tron)

hinh\_tron\_xa\_goc = max(danh\_sach\_hinh\_tron, key=lambda x: x.khoang\_cach\_den\_goc())

print("%d %d %d %.3f"%(hinh\_tron\_xa\_goc.ma,hinh\_tron\_xa\_goc.tamx,hinh\_tron\_xa\_goc.tamy,hinh\_tron\_xa\_goc.bk))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 52: Công ty HP 3 loại khách hàng (Doanh nghiệp)

class KhachHang:

def \_\_init\_\_(self, ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky):

self.ten = ten

self.ma\_kh = ma\_kh

self.chi\_so\_dau\_ky = chi\_so\_dau\_ky

self.chi\_so\_cuoi\_ky = chi\_so\_cuoi\_ky

def luong\_dien\_tieu\_thu(self):

return self.chi\_so\_cuoi\_ky - self.chi\_so\_dau\_ky

def tinh\_tien\_phai\_tra(self):

don\_gia\_thuong = 1000

if self.ma\_kh[0] == 'D':

don\_gia\_thuong \*= 1.05

elif self.ma\_kh[0] == 'K':

don\_gia\_thuong \*= 1.07

return self.luong\_dien\_tieu\_thu() \* don\_gia\_thuong

def main():

n = int(input())

danh\_sach\_khach\_hang = []

for \_ in range(n):

thong\_tin\_kh = input().split()

ten = thong\_tin\_kh[0]

ma\_kh = thong\_tin\_kh[1]

chi\_so\_dau\_ky = int(thong\_tin\_kh[2])

chi\_so\_cuoi\_ky = int(thong\_tin\_kh[3])

khach\_hang = KhachHang(ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky)

danh\_sach\_khach\_hang.append(khach\_hang)

khach\_hang\_ho\_kinh\_doanh = [kh for kh in danh\_sach\_khach\_hang if kh.ma\_kh[0] == 'K']

if not khach\_hang\_ho\_kinh\_doanh:

print("Khong co khach hang la ho kinh doanh.")

else:

khach\_hang\_ho\_kinh\_doanh.sort(key=lambda x: x.tinh\_tien\_phai\_tra(), reverse=True)

khach\_hang\_max\_tien = khach\_hang\_ho\_kinh\_doanh[0]

print("{} {} {} {} {} {}".format(

khach\_hang\_max\_tien.ten,

khach\_hang\_max\_tien.ma\_kh,

khach\_hang\_max\_tien.chi\_so\_dau\_ky,

khach\_hang\_max\_tien.chi\_so\_cuoi\_ky,

khach\_hang\_max\_tien.luong\_dien\_tieu\_thu(),

khach\_hang\_max\_tien.tinh\_tien\_phai\_tra()

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 53: Nhân Viên giảm dần lương

class NhanVien:

def \_\_init\_\_(self, ten, ma, hsl, pc):

self.ten = ten

self.ma = ma

self.hsl = hsl

self.pc = pc

self.luong = hsl \* 2000000 + pc

def main():

n = int(input())

danh\_sach\_nhan\_vien = []

for \_ in range(n):

thong\_tin\_nv = input().split()

ten = thong\_tin\_nv[0]

ma = int(thong\_tin\_nv[1])

hsl = float(thong\_tin\_nv[2])

pc = int(thong\_tin\_nv[3])

nhan\_vien = NhanVien(ten, ma, hsl, pc)

danh\_sach\_nhan\_vien.append(nhan\_vien)

danh\_sach\_nhan\_vien.sort(key=lambda x: x.luong, reverse=True)

print("Danh sach nhan vien da sap xep: {}".format(n))

for nv in danh\_sach\_nhan\_vien:

print("{} {} {:.2f} {} {:.2f}".format(

nv.ma, nv.ten, nv.hsl, nv.pc, nv.luong

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 54: Công ty HP 3 loại khách hàng (Hộ Kinh Doanh)

class KhachHang:

def \_\_init\_\_(self, ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky):

self.ten = ten

self.ma\_kh = ma\_kh

self.chi\_so\_dau\_ky = chi\_so\_dau\_ky

self.chi\_so\_cuoi\_ky = chi\_so\_cuoi\_ky

def luong\_dien\_tieu\_thu(self):

return self.chi\_so\_cuoi\_ky - self.chi\_so\_dau\_ky

def tinh\_tien\_phai\_tra(self):

don\_gia\_thuong = 1000

if self.ma\_kh[0] == 'D':

don\_gia\_thuong \*= 1.05

elif self.ma\_kh[0] == 'K':

don\_gia\_thuong \*= 1.07

return int(self.luong\_dien\_tieu\_thu() \* don\_gia\_thuong)

def main():

n = int(input())

danh\_sach\_khach\_hang = []

for \_ in range(n):

thong\_tin\_kh = input().split()

ten = thong\_tin\_kh[0]

ma\_kh = thong\_tin\_kh[1]

chi\_so\_dau\_ky = int(thong\_tin\_kh[2])

chi\_so\_cuoi\_ky = int(thong\_tin\_kh[3])

khach\_hang = KhachHang(ten, ma\_kh, chi\_so\_dau\_ky, chi\_so\_cuoi\_ky)

danh\_sach\_khach\_hang.append(khach\_hang)

khach\_hang\_ho\_kinh\_doanh = [kh for kh in danh\_sach\_khach\_hang if kh.ma\_kh[0] == 'K']

if not khach\_hang\_ho\_kinh\_doanh:

print("Khong co khach hang la ho kinh doanh.")

else:

khach\_hang\_ho\_kinh\_doanh.sort(key=lambda x: x.tinh\_tien\_phai\_tra(), reverse=True)

khach\_hang\_max\_tien = khach\_hang\_ho\_kinh\_doanh[0]

print("{} {} {} {} {} {}".format(

khach\_hang\_max\_tien.ten,

khach\_hang\_max\_tien.ma\_kh,

khach\_hang\_max\_tien.chi\_so\_dau\_ky,

khach\_hang\_max\_tien.chi\_so\_cuoi\_ky,

khach\_hang\_max\_tien.luong\_dien\_tieu\_thu(),

khach\_hang\_max\_tien.tinh\_tien\_phai\_tra()

))

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 55: Số thứ tự thủ khoa

class ThiSinh:

def \_\_init\_\_(self, ten, sobaodanh, diem1, diem2, diem3):

self.ten = ten

self.sobaodanh = sobaodanh

self.diem1 = diem1

self.diem2 = diem2

self.diem3 = diem3

self.tongdiem = diem1 + diem2 + diem3

def \_\_str\_\_(self):

return "{} {} {:.2f} {:.2f} {:.2f} {:.2f}".format(

self.sobaodanh, self.ten, self.diem1, self.diem2, self.diem3, self.tongdiem

)

def main():

n = int(input())

danh\_sach\_thisinh = []

for \_ in range(n):

thong\_tin\_ts = input().split()

ten = thong\_tin\_ts[0]

sobaodanh = int(thong\_tin\_ts[1])

diem1 = float(thong\_tin\_ts[2])

diem2 = float(thong\_tin\_ts[3])

diem3 = float(thong\_tin\_ts[4])

thisinh = ThiSinh(ten, sobaodanh, diem1, diem2, diem3)

danh\_sach\_thisinh.append(thisinh)

thi\_sinh\_do = [ts for ts in danh\_sach\_thisinh if ts.tongdiem >= 15 and ts.diem1 >= 1 and ts.diem2 >= 1 and ts.diem3 >= 1]

if not thi\_sinh\_do:

print("Khong co thi sinh thi do")

else:

thu\_khoa = max(thi\_sinh\_do, key=lambda x: (x.tongdiem, x.sobaodanh))

so\_thu\_tu\_thu\_khoa = danh\_sach\_thisinh.index(thu\_khoa)

print("So thu tu cua thu khoa: {}".format(so\_thu\_tu\_thu\_khoa))

print(thu\_khoa)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Câu 56: Nhân viên lương lớn nhất

class NV:

def \_\_init\_\_(self, ten, ma, hsl, pc):

self.ten = ten

self.ma = ma

self.hsl = hsl

self.pc = pc

def Luong(self):

return self.hsl \* 2000000 + self.pc

n = int(input())

m = []

for i in range(n):

infor = input().split()

ten = infor[0]

ma = int(infor[1])

hsl = float(infor[2])

pc = int(infor[3])

t = NV(ten, ma, hsl, pc)

m.append(t)

nv\_luong\_max = max(m, key=lambda nv:nv.Luong())

print("Nhan vien co luong lon nhat")

print(nv\_luong\_max.ma, nv\_luong\_max.ten, format(nv\_luong\_max.hsl, ".2f"), nv\_luong\_max.pc, format(nv\_luong\_max.Luong(), ".2f"))

Câu 57: