assignment 1

- Karn Arora
- J078
- Data Science
- 3rd Year

day 0

Double-click (or enter) to edit

```
input_string = input()
```

```
     Hello,World.
```

```
print(input_string)
```

```
     Hello,World.
```

day 1:Data Types

```
#initialising the 3 variables(integer,double and string)
i=19
d=3.0
s="data science"
```

```
#Read 3 lines of input from stdin
i_new = int(input())
d_new = float(input())
s_new = input()
```

```
     3
     2.0
     machine learning
```

```
#Use the '+'  operator to perform the following operations:
#Print the sum of i plus your int variable on a new line.
#Print the sum of d plus your double variable to a scale of one decimal place on a new lin
#Concatenate  with the string you read as input and print the result on a new line.
print(f'(i) {i+i_new}')
print(f'(ii) {round(d+d_new,1)}')
print(f'(iii) {s + s_new}')
```

```
(i) 22
(ii) 5.0
(iii) data sciencemachine learning
```

day 2: Operators

Given the meal price (base cost of a meal), tip percent (the percentage of the meal price being added as tip), and tax percent (thepercentage of the meal price being added as tax) for a meal, find and print the meal's total cost. Round the result to the nearest integer

```
meal_cost = 749.99
tip_percentage = 15
tax_percentage = 20
total_cost = (meal_cost*tip_percentage/100)+(tax_percentage*meal_cost/100)+meal_cost
print('total cost of the meal was:',round(int(total_cost)))
```

```
total cost of the meal was: 1012
```

Day 3:Intro to Conditional Statements

```
import sys


n = int(input().strip())

# if number is odd
if n%2==1:
    answer = "Weird"
#if number is even and greater than 20
elif n>20:
    answer = "Not Weird"
#if number is even and in the inclusive range of 6 to 20
elif n>=6:
    answer = "Weird"
#if number is ven and in the inclusive range of 2 to 5
else:
    answer = "Not Weird"

print(answer)
```

```
23
Weird
```

Day 4:Class vs. Instance

Write a Person class with an instance variable, age, and a constructor that takes an integer, initialAge, as a parameter.

The constructor must assign initialAge to Age after confirming the argument passed as initialAge is not negative;

if a negative argument is passed as initialAge, theconstructor should set age to 0 and print Age is not valid, setting age to 0..

In addition, you must write the following instance methods:yearPasses() should increase the instance variable by 1.

amIOld() should perform the following conditional actions:If , print You are young..If and , print You are a teenager..Otherwise, print You are old.

```
class Person:
    def __init__(self,initialAge):

        if(initialAge > 0):
            self.age = initialAge
        else:
            print("Age is not valid, setting age to 0.")
            self.age = 0

    def amIOld(self):

        if self.age >= 18:
            print("You are old.")
        elif self.age >= 13:
            print("You are a teenager.")
        else: # age < 13
            print("You are young.")

    def yearPasses(self):

        self.age += 1

for f in range(3):
  initial_age=int(input('Enter Age: '))
  person=Person(initial_age)
  person.amIOld()
  person.yearPasses()
  person.amIOld()
```

```
Enter Age: 17
You are a teenager.
You are old.
Enter Age: -78
Age is not valid, setting age to 0.
You are young.
You are young.
Enter Age: 69
```

```
You are old.
You are old
```

Day 5:

Given an integer, n, print its first 10 multiples.

```
n = int(input())

for i in range(1,11):
  print(f'{n} x {i} = {n*i}')
```

```
33
33 x 1 = 33
33 x 2 = 66
33 x 3 = 99
33 x 4 = 132
33 x 5 = 165
33 x 6 = 198
33 x 7 = 231
33 x 8 = 264
33 x 9 = 297
33 x 10 = 330
```

day 6:

Given a string, s, of length N that is indexed from 0 to N-1, print its even-indexed and odd-indexed characters as 2 space-separated strings on a single line

```
import sys

def printEvenIndexChar(s):
    l = len(s)
    output = ""
    for i in range(0,l,2):
        output += s[i]
    return output

def printOddIndexChar(s):
    l = len(s)
    output = ""
    for i in range(1,l,2):
        output += s[i]
    return output

t = int(input())
for a0 in range(0,t):
    s = input()
    print(printEvenIndexChar(s)+ " "  + printOddIndexChar(s))
```

```
2
sci-fi
sif c-i
```

```
happy man
hpymn ap a
```

Day 7:Arrays

```
A=[0,1,2,3,4,5,6,7,8,9]
for i in A[::-1]:
  print(i,end=" ")
```

```
9 8 7 6 5 4 3 2 1 0
```

Day 8:

```
phonebook={}
for i in range(int(input('>>>'))):
  inp=input().split()
  phonebook[inp[0]]=inp[1]
q=input('Query: ')
while q !="":
  try:
    print(f'{q}={phonebook[q]}')
  except:
      print('Not Found!')
      q = input('Query:')
      break
```

```
>>>1
jeff 2233
Query: jose 3535
Not Found!
Query:jeff 2233
```

Day 9:

```
def rec_fact(n):
  if n == 0 or n == 1:
    return 1
  else:
      return n*rec_fact(n-1)
print(rec_fact(int(input('>>>'))))
```

```
>>>6
720
```

Day:10

```
n = int(input('>>>'))
binary = ''
while n > 0:
  binary = str(n % 2) + binary
n //= 2
print(max([len(i) for i in binary.split('0')]))
```

DAY 11

```
import sys


arr = []
for arr_i in range(6):
   arr_temp = list(map(int,input().strip().split(' ')))
   arr.append(arr_temp)
max = 0

for i in range(0,4):
    for j in range(0,4):
        sum = 0
        sum= arr[i][j]+arr[i][j+1]+arr[i][j+2]+arr[i+1][j+1]+arr[i+2][j]+arr[i+2][j+1]+arr
        if i==0 and j==0:
            max = sum
        if sum > max:
            max =sum

print(max)
```

Day 12

```
class Student:
  def __init__(self,  first_name,  last_name,  id_number,  scores):
     self.first_name = first_name
     self.last_name = last_name
     self.id_number = id_number
     self.scores = scores
     def calculate(self):
        d = {
            'T':40,
            'D':55,
            'P':70,
            'A':80,
            'E':90,
            'O':100,
        }

        avg = sum(self.scores) /len(self.scores)
```

```
        for grade, score_threshold in d.items():
          if avg < score_threshold:
            return grade
            karn = Student('Karn', 'Arora', 70091017003, [100, 99, 89, 91])
            print(f'Name: {karn.first_name} {karn.last_name}')
            print(f'ID: {karn.id_number}')
            print(f'Grade: {karn.calculate()}')
```

Day 13

```
class Book:
  def _init_(self, title, author):
     self.title =  title
     self.author = author
class MyBook(Book):
  def _init_(self, title, author, price):
     Book._init_(self, title, author)
     self.price = price
  def display(self):
       print(f'Title: {self.title}\nAuthor: {self.author}\nPrice: {self.price}')
my_book1 = Book('Futre','Dave',1499)
my_book1.display()
```

Day 14

```
class Difference:
   def __init__(self,elements_):
     self.elements_ = elements_
   def maximum_difference(self):\
    return max(self.elements_)- min(self.elements_)

diff = Difference([1,2,3,4,5,6,7,8,9])
diff.maximum_difference()
```

```
8
```

Day 15

```
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None
class Solution:
    def display(self,head):
        current = head
        while current:
```

```
            print(current.data,end=' ')
            current = current.next


    def insert(self,head,data):
       if head is None:
          head = Node(data)
       elif head.next is None:
          head.next = Node(data)
       else:
        self.insert(head.next, data)
       return head


mylist= Solution()
T=int(input())
head=None
for i in range(T):
    data=int(input())
    head=mylist.insert(head,data)
mylist.display(head);
```

```
 2
 3
 4
 3 4
```

Day 16

```
import sys


S = input().strip()
try:
    r = int(S)
    print(r)
except ValueError:
    print("Bad String")
```

```
 jaka4
 Bad String
```

Day 17

```
class Calculator(Exception):
    def power(self,n,p):
        if (n<0 or p<0):
            raise Calculator("n and p should be non-negative")
        else:
            return pow(n,p)
myCalculator=Calculator()
```

```
myCalculator=Calculator()
T=int(input())
for i in range(T):
    n,p = map(int, input().split())
    try:
        ans=myCalculator.power(n,p)
        print(ans)
    except Exception as e:
        print(e)
```

```
 1
 23 12
 21914624432020321
```

Day 19

```
class AdvancedArithmetic(object):
    def divisorSum(n):
        raise NotImplementedError

class Calculator(AdvancedArithmetic):
    def divisorSum(self, n):
        s = 0
        for i in range(1,n+1):
            if (n%i == 0):
                s+=i
        return s


n = int(input())
my_calculator = Calculator()
s = my_calculator.divisorSum(n)
print("I implemented: " + type(my_calculator).__bases__[0].__name__)
print(s)
```

```
 23
 I implemented: AdvancedArithmetic
 24
```

Day 20

```
import math
import os
import random
import re
import sys



if __name__ == '__main__':
    n = int(input().strip())
```

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

    numberOfSwaps = 0
for i in range(0,n):
    for j in range(0, n-1):
        if (a[j] > a[j + 1]):
            temp=a[j]
            a[j] = a[j+1]
            a[j+1] = temp
            numberOfSwaps += 1
    if (numberOfSwaps == 0):
        break
print( "Array is sorted in " + str(numberOfSwaps) + " swaps." )
print( "First Element: " + str(a[0]) )
print( "Last Element: " +  str(a[n-1]) )
```

```
 2
 1 2
 Array is sorted in 0 swaps.
 First Element: 1
 Last Element: 2
```

Day 24

```
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None
class Solution:
    def insert(self,head,data):
            p = Node(data)
            if head==None:
                head=p
            elif head.next==None:
                head.next=p
            else:
                start=head
                while(start.next!=None):
                    start=start.next
                start.next=p
            return head
    def display(self,head):
        current = head
        while current:
            print(current.data,end=' ')
            current = current.next

    def removeDuplicates(self,head):
        current = head
        while (current.next):
            if (current.data == current.next.data):
                 current.next = current.next.next
```

```
            else:
                current = current.next

        return head

mylist= Solution()
T=int(input())
head=None
for i in range(T):
    data=int(input())
    head=mylist.insert(head,data)
head=mylist.removeDuplicates(head)
mylist.display(head);
```

```
 1
 2
 2
```

Day 25

```
import math

def check_prime(num):
    if num is 1:
        return "Not prime"
    sq = int(math.sqrt(num))
    for x in range(2, sq+1):
        if num % x is 0:
            return "Not prime"
    return "Prime"


t = int(input())
for i in range(t):
    number = int(input())
    print(check_prime(number))
```

```
 3
 49
 Not prime
 7
 Prime
 5
 Prime
```

Day 26

```
da, ma, ya = input().split(' ')
da = int(da)
ma = int(ma)
ya = int(ya)
```

```
de, me, ye = input().split(' ')
de = int(de)
me = int(me)
ye = int(ye)
fine = 0
if(ye==ya):
    if(me < ma):
        fine = (ma - me) * 500
    elif((me == ma) and (de < da)):
        fine = (da - de) * 15
elif(ye < ya):
    fine = 10000

print( fine )
```

```
 23 7 1998
 15 7 1998
 120
```

Day 28

```
if __name__ == '__main__':
  N = int(input().strip())
  names=[]

  for N_itr in range(N):
    i=input().split()
    if '@gmail.com' in i[1]:
      names.append(i[0])

  for name in sorted(names):
     print(name)
```

```
 2
 karn karn@gmail.com
 papa papay@gmail.com
 karn
 papa
```

Day 29

```
def bitwiseAnd(N, K):
  max_possible = K | (K-1)
  if max_possible <= N:
    return K-1
  else:
    return K-2
```

✓ 0s completed at 1:42 AM