ANSHUL SAVLA

J053

Day - 0

In [1]:
```
var = input()
print('Hello World')
print(var)
```

```
Hello Anshul
Hello World
Hello Anshul
```

Day - 1

In [2]:
```
i = 4
d = 4.0
s = 'HackerRank '

ii = int(input())
dd = float(input())
ss = input()

print(i+ii)
print(d+dd)
print(s+ss)
```

```
4
2.3
anshul
8
6.3
HackerRank anshul
```

Day - 2

In [5]:
```
import numpy as np
cost = 100
tip_p= 15
tax_p = 8

tip = cost*0.15
tax = cost*0.08

print(np.round(tip+tax+cost,0))
```

```
123.0
```

Day - 3

In [6]:
```
n = int(input())

if n%2==1:
    ans = "Weird"

elif n>20:
    ans = "Not Weird"

elif n>=6:
    ans = "Weird"

else:
    ans = "Not Weird"

print(ans)
```

```
4
Not Weird
```

Day - 4

In [1]:
```
class Person:
    def __init__(self,initialAge):
        # Add some more code to run some checks on initialAge
        if(initialAge > 0):
            self.age = initialAge
        else:
            print("Age is not valid, setting age to 0.")
            self.age = 0

    def amIOld(self):
        # Do some computations in here and print out the correct stateme
nt to the console
        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):
        # Increment the age of the person in here
        self.age += 1

t = int(input())
for i in range(0,t):
    age = int(input())
    p = Person(age)
    p.amIOld()
    for j in range(0,3):
        p.yearPasses()
    p.amIOld()
    print("")
```

```
2
18
You are old.
You are old.

32
You are old.
You are old.
```

Day - 5

In [7]:
```
inp = int(input())
for i in range(10):
    print(inp ,'x', i ,'=', inp*i)
```

```
3
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
```

Day - 6

In [15]:
```
s = input()
for i in range(0 , len(s) , 2):
    print(s[i])
for j in range(1 , len(s) , 2):
    print(s[j])
```

```
anshul
a
s
u
n
h
l
```

Day - 7

In [20]:
```
a = [4 , 2 ,3 ,1]
for i in range(len(a)-1 , -1, -1):
    print(a[i])
```

```
1
3
2
4
```

Day - 8

In [4]:
```
dict = {}

for i in range(4):
    name = input()
    phone = input()
    dict[name] = []
    #dict[phone] = []
    dict[name].append(phone)
    #dict[phone].append(phone)

print(dict)
key_list = list(dict.keys())
val_list = list(dict.values())
position = key_list.index(input())
print(val_list[position])
```

```
anshul
1
arham
2
saumya
3
nishant
4
{'anshul': ['1'], 'arham': ['2'], 'saumya': ['3'], 'nishant': ['4']}
anshul
['1']
```

Day - 9

In [5]:
```
def factorial(n):
    if n<=1:
        return 1
    else:
        return n*factorial(n-1)

n = int(input())
print(factorial(n))
```

```
76
1885494701666050254987932260861146558230394535379329335672487982961844043
4955379231177299722240000000000000000000
```

Day - 10

In [7]:
```
def max(a,b):
    return a if a>b else b

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

max_num = 0
count = 0

while n:
    while n&1:
        count += 1
        n>>=1
    max_num = max(count, max_num)
    if not n&1:
        count = 0
        n>>=1

print(max_num)
```

```
15
4
```

Day - 11

In [15]:
```
import numpy as np
arr = np.array([[1,1,1,0,1,0] , [0,1,0,0,0,0] , [1,2,3,1,2,3] , [0,0,0,0
,1,0] , [1,1,1,1,1,1] , [1,2,3,0,2,3]])
max = 0
n = len(arr)

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

print(max)
```

```
10
```

Day - 12

In [ ]:

Day - 13

In [ ]:

Day - 14

In [4]:
```
class Difference:
    def __init__   (self, elements_):
        self.elements_ = elements_

    def maximum_difference(self):
        return max(self.elements_) - min(self.elements_)

diff = Difference([1 , 11 , 34 , 23 , 16 , 13 ,1])
diff.maximum_difference()
```

Out[4]: 33

Day - 15

In [22]:
```
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
```

Day - 16

In [3]:
```
S = input()
try:
    r = int(S)
    print(r)
except ValueError:
    print("Bad String")
```

```
anshul
Bad String
```

Day - 17

In [6]:
```
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)
```

Day - 18

In [14]:
```
class Solution:
    def __init__ (self):
        self.stack = []
        self.queue = []

    def push_char(self, ch):
        self.stack.append(ch)

    def enqueue_char(self, ch):
        self.queue.append(ch)

    def pop_char(self):
        try:
            x = self.stack[-1]
            self.stack = self.stack[:-1]
            return x
        except:
            return None

    def dequeue(self):
        try:
            x = self.queue[0]
            self.queue = self.queue[1:]
            return x
        except:
            return None

sol = Solution()
S = 'bannab'
for c in S:
    sol.push_char(c)
    sol.enqueue_char(c)

flag = True
while True:
    from_stack = sol.pop_char()
    from_queue = sol.dequeue()

    if from_stack == None:
        break

    if from_stack != from_queue:
        flag = False
        break

if flag:
    print("Palindrome")
else:
    print("Not Palindrome")
```

```
Palindrome
```

Day - 19

In [6]:
```
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
```

Day - 20

In [8]:
```
n = int(input().strip())
a = list(map(int, input().strip().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]) )
```

```
6
3 7 2 8 4 9
Array is sorted in 4 swaps.
First Element: 2
Last Element: 9
```

Day - 21

Cant be done in python

Day - 22

In [34]:
```
def getHeight(self,root):
        if root is None or (root.left is None and root.right is None):
            return 0
        else:
            return max(self.getHeight(root.left),self.getHeight(root.rig
ht))+1
```

Day - 23

In [23]:
```
def levelOrder(self,root):
        output = ""
        queue = [root]
        while queue:
            current = queue.pop(0)
            output += str(current.data) + " "
            if current.left:
                queue.append(current.left)
            if current.right:
                queue.append(current.right)
        print(output[:-1])
```

Day - 24

In [24]:
```
def removeDuplicates(self,head):
    #Write your code here
    current = head
    while (current.next):
        if (current.data == current.next.data):
            current.next = current.next.next
        else:
            current = current.next

    return head
```

Day - 25

In [26]:
```
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))
```

```
4
1
Not prime
2
Prime
3
Prime
4
Not prime
```

Day - 26

In [29]:
```

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 )
```

```
4 5 6
4 5 6
0
```

Day - 27

In [30]:
```
def minimum_index(seq):
    if len(seq) == 0:
        raise ValueError("Cannot get the minimum value index from an emp
ty sequence")
    min_idx = 0
    for i in range(1, len(seq)):
        if seq[i] < seq[min_idx]:
            min_idx = i
    return min_idx

class TestDataEmptyArray(object):

    @staticmethod
    def get_array():
        return []

class TestDataUniqueValues(object):

    @staticmethod
    def get_array():
        return [7, 4, 3, 8, 14]

    @staticmethod
    def get_expected_result():
        return 2

class TestDataExactlyTwoDifferentMinimums(object):

    @staticmethod
    def get_array():
        return [7, 4, 3, 8, 3, 14]

    @staticmethod
    def get_expected_result():
        return 2
```

Day - 28

In [32]:
```
import sys
import re


N = int(input().strip())
names = []
for a0 in range(N):
    firstName,emailID = input().strip().split(' ')
    firstName,emailID = [str(firstName),str(emailID)]
    match = re.search(r'[\w\.-]+@gmail.com', emailID)

    if match:
        names.append(firstName)
names.sort()
for name in names:
    print( name )
```

```
2
anshul a@gmsil.com
piyu p@yahoo.com
```

Day - 29

In [33]:
```
import sys


t = int(input().strip())
for a0 in range(t):
    n, k = input().strip().split(' ')
    n, k = [int(n), int(k)]
    print(k-1 if ((k-1) | k) <= n else k-2)
```

```
3
2 3
1
6 9
7
2 11
9
```

In [ ]: