Day-6:

In [11]:

```
l = [1,2,3,4]
l
```

Out[11]:

```
[1, 2, 3, 4]
```

In [6]:

```
l.append([4,3,2,1])
```

In [9]:

```
l.pop(8)
```

Out[9]:

```
[4, 3, 2, 1]
```

In [10]:

```
l
```

Out[10]:

```
[1, 2, 3, 4, 4, 3, 2, 1]
```

Problem in mock test

Closest to Zero

-

In [45]:

```
n = int(input())
s = input()

ls = convertToIntList(s)
pl = []
for i in ls:
    pl.append(abs(i))
pl.sort()
if pl[-1] in ls:
    print(pl[-1])
else:
    print(-pl[-1])
```

```
4
-1 -1 0 3
3
```

In [43]:

```
def convertToIntList(s):
    ls =s.split()
    cl = []
    for i in ls:
        cl.append(int(i))
    return cl
```

In [13]:

```
l[-1::-1] # reverse sorting order
```

Out[13]:

```
[4, 3, 2, 1]
```

In [19]:

```
l.append([-2,4,-4,-6])
```

In [22]:

```
l
```

Out[22]:

```
[1, 2, 3, 4]
```

In [ ]:

```
l
```

Problem -3

you are given numbers a , b, c. Write a program to find the largest number which is less than or equal to c and leaves remainder b when divided by a or print -1

In [78]:

```
# Normal Logic
a = 3
b = 2
c = 9
for i in range(c,a-1,-1):
    if i%a == b:
        print(i)
        break
```

```
8
```

In [66]:

```
# Functions
def calc3best(a,b,c):
    for c in range(c,a-1,-1):
        if c%a == b:
            return c
    return -1
calc3best(3,2,9)
```

Out[66]:

```
8
```

In [143]:

```
# To Generate n prime numbers
def getPrimes(v):
    ls = []
    for i in range(1,v+1):
        if(checkPrime(i)):
            ls.append(i)

    return ls
getPrimes(10)
```

Out[143]:

```
[1, 2, 3, 5, 7]
```

In [ ]:

```

```

In [4]:

```
# Check Prime or not
def checkPrime(i):
    flag = 0
    if(i == 1 or i==2):
        return True
    else:
        for j in range(2,i):
            if(i%j == 0):
                flag = 1
    if flag == 1:
        return False
    else:
        return True
```

In [5]:

```
def getPrimes(n):
    ls = []
    i = 1

    while(n>0):
        if(checkPrime(i)):
            ls.append(i)
            n = n-1
            i = i+1
        else:
            i = i+1

    return ls
getPrimes(10)
```

Out[5]:

```
[1, 2, 3, 5, 7, 11, 13, 17, 19, 23]
```

In [132]:

```
# Generate n Fibonacci series
def getFibos(n):
    ls = []
    temp = 0
    a = 0
    b = 1
    for i in range(1,n+1):
        ls.append(a)
        temp = a+b
        a = b
        b = temp
    return ls
# getFibos(6)
```

Out[132]:

```
[0, 1, 1, 2, 3, 5]
```

In [ ]:

```
## TA's Tasks Vijay

n = int(input())
pls = getPrimes(n)
fils = getFibos(n)
data = generatePrimesAndFibos(pls,fils)
```

In [88]:

```
def check(l,r,k):
    count = 0
    for l in range(l,r+1):
        if l%k == 0:
            count+=1
    return count
a = check(1,10,2)
print(a)
```

```
5
```

In [98]:

```
def check(l,r,k):
    count = 0
    for l in range(l,r+1):
        if l%k == 0:
            count+=1
    return count
ls = input()
ls = ls.split()
l = int(ls[0])
r = int(ls[1])
k = int(ls[2])

a = check(l,r,k)
print(a)
```

```
1 10 1
10
```

In [99]:

```
# Finding Factorial number
def findFact(n):
    count = 1
    for i in range(1,n+1):
        count *=i
    return count
n=int(input())
print(findFact(n))
```

```
2
2
```

In [125]:

```
s = "aBcse"
ss =s.swapcase()
ss
```

Out[125]:

```
'AbCSE'
```

In [ ]:

```

```

In [126]:

```
def doToggle(s):
    s = s.swapcase()
    return s
s = input()
print(doToggle(s))
```

```
asdFgG
ASDfGg
```

In [ ]:

```
# Task 5: Generating prime numbers in a range

def checkPrime(i):
    flag = 0
    if(i == 1 or i==2):
        return True
    else:
        for j in range(2,i):
            if(i%j == 0):
                flag = 1
    if flag == 1:
        return False
    else:
        return True

def generatePrimes(n):
    for j in range(2,n+1):
        if(checkPrime(j)):
            print(j,end=" ")

n = int(input())
generatePrimes(n)
```

In [21]:

```
# Duration Problem:
def countMin(sh,sm,eh,em):
    total = 0
    if (sh+1)==eh or (eh-1)==sh:
        total = total + (60-sm)+em
    elif(sh==eh):
        total = em-sm
    else:
        i = sh+1
        j = eh-1
        for i in range(i,j+1):
            total += 60

    return total

def duration(sh,sm,eh,em):
    tot_min = countMin(sh,sm,eh,em)
    print(tot_min//60," ",tot_min%60)
s = input()
ls = input()
ls = ls.split()
l = int(ls[0])
r = int(ls[1])
k = int(ls[2])
duration(2,0,3,0)
```

```
1   0
```

In [47]:

```
def countMin(sh,sm,eh,em):
    total = 0
    if (sh+1)==eh or (eh-1)==sh:
        total = total + (60-sm)+em
    elif(sh==eh):
        total = em-sm
    else:
        total = (60-sm)+em
        i = sh+1
        j = eh
        for i in range(i,j):
            total += 60

    return total

def duration(sh,sm,eh,em):
    tot_min = countMin(sh,sm,eh,em)
    hours = tot_min//60
    mint = tot_min%60
    print(hours,"",mint)

n = int(input())
lst=[]
for i in range(1,n+1):
    s = input()
    lst.append(s)
def cal(lst):
    for i in range(0,len(lst)):
        temp = lst[i].split()
        sh = int(temp[0])
        sm = int(temp[1])
        eh = int(temp[2])
        em = int(temp[3])
        duration(sh,sm,eh,em)
cal(lst)
```

```
2
5 40 8 0
3 00 6 50
2  20
3  50
```

In [38]:

```
i=0
ad = lst[i].split()
a = int(ad[0])
a
```

Out[38]:

```
2
```

In [ ]:

```

```