Relative Speed

1 mar

Problem statement

You are given a function,

```
int RelativeSpeed(int direction, int distance1, int time1, int distance2, int time2);
```

The function accepts integers 'direction', 'distance1', 'time1', 'distance2' & 'time2' as its argument. 'distance1' & 'distance2' are the distances travelled and 'time1' & 'time2' are the time taken by 1st & 2nd person respectively. Implement the function to find 'speed1' & 'speed2' of 1st & 2nd person respectively and return the relative speed as per given conditions:

- if direction = 0, relative speed = speed1 + speed2
- if direction = 1, relative speed = speed1 - speed2

Speed = Distance / Time

Assumption:

- speed1 >= speed2
- distance1, time1, distance2, time2 are greater than zero.

Note:

- direction is either 0 or 1.
- Computed value lies within integer range.
- Consider division as integer division.

Example:

Input:

direction: 0
distance1: 30
time1: 5
distance2: 20
time2: 10

Output:

8

Explanation:

speed1 = 30 / 5 = 6
speed2 = 20 / 10 = 2
Since direction = 0, relative speed = (6 + 2) = 8.

Sample Input

```
direction: 1
distance1: 50
time1: 2
distance2: 60
time2: 3
```

Sample Output

```
5
```

Instructions :

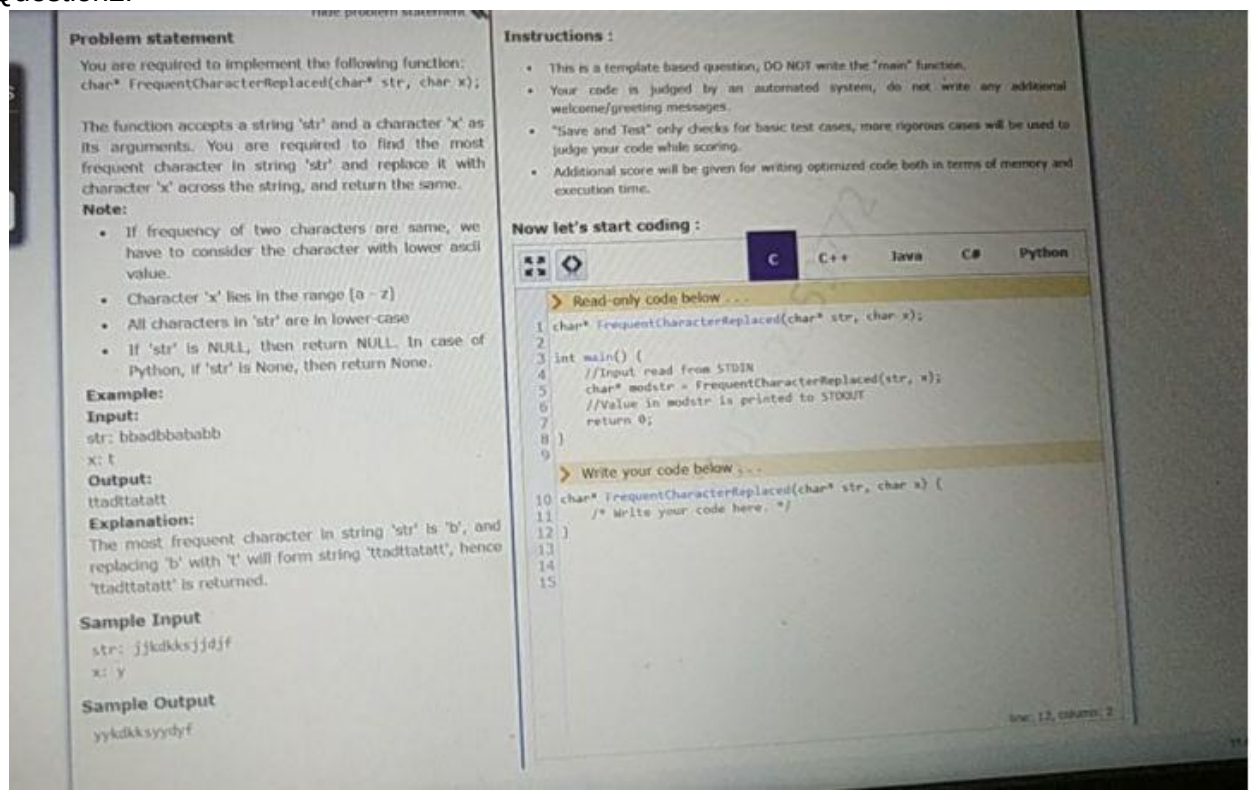
Solution:

```
def RelativeSpeed(direction, distance1, time1, distance2, time2):
#Solved by Placement Solutions 2020-2021
  relativespeed = 0
  direction = int(input())
  distance1 = int(input())
  time1 = int(input())
  distance2 = int(input())
  time2 = int(input())
  if (direction==0) :
    result = (distance1/time1)+(distance2/time2)
  elif direction==1:
    result = (distance1/time1)-(distance2/time2)
  return result

result = RelativeSpeed(direction, distance1, time1, distance2, time2)
print(result)
```

```
1
50
2
60
3
5.0
```

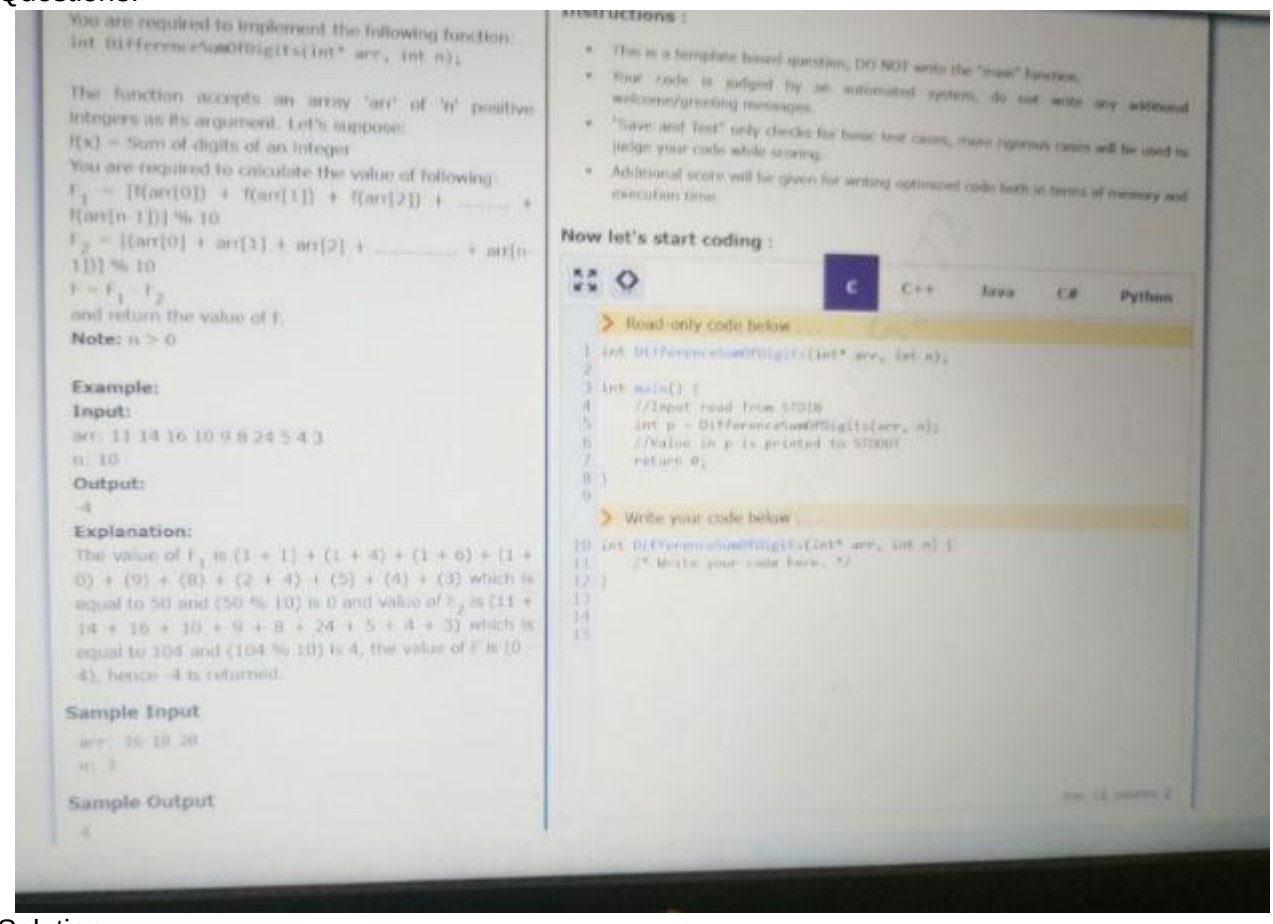
Question2:



```python
def GetChar(myDict, k_max):
    chList = []
    for ch, val in myDict.items():
        if(val == k_max):
            chList.append(ch)
    if(len(chList) == 1):
        return chList.pop()
    else:
        aVal = [ord(ch) for ch in chList]
        k = aVal.index(min(aVal))
        return chList[k]
def FrequentCharacterReplaced(myStr, new_ch):
    myDict = {}
    chList = set(myStr)
    for letter in chList:
        myDict[letter] = myStr.count(letter)
    k_max = max(myDict.values())
    ch = GetChar(myDict, k_max)
    return myStr.replace(ch, new_ch)
myStr, ch = input().split()
print(FrequentCharacterReplaced(myStr, ch))
```

```
jjkdkksjjdjf y
yykdkksyydyf
```

Question3:



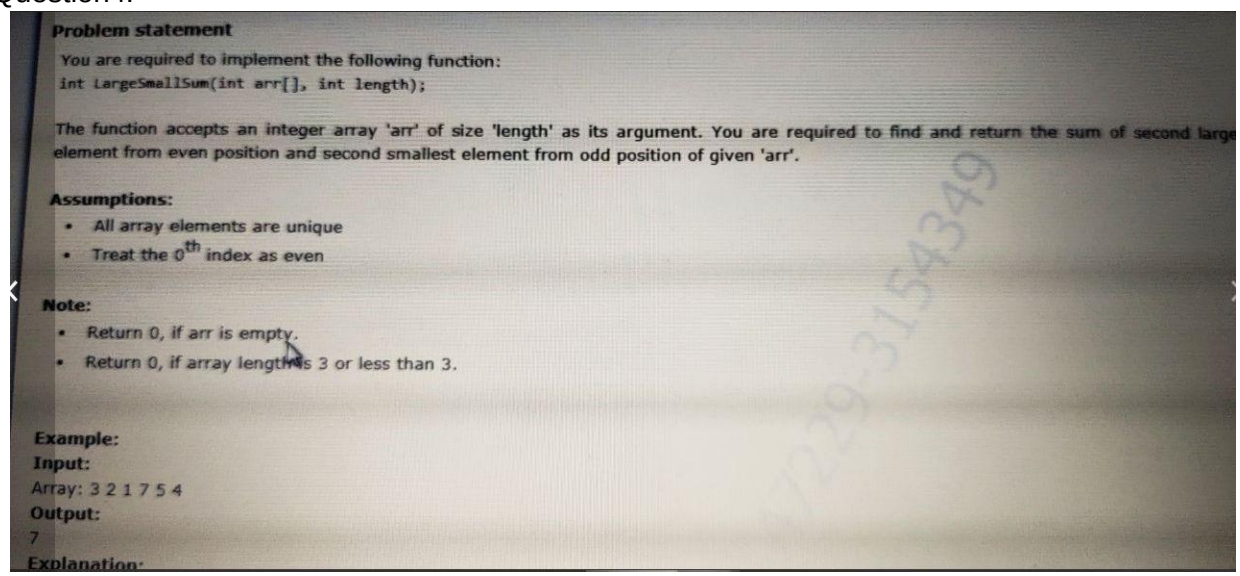
Solution:

```python
def GetSumOfDigit(num):
    d_sum = 0;
    while(num!=0):
        d_sum = d_sum + (num%10)
        num = num // 10
    return d_sum
def DifferenceSumOfDigits(arr, n):
    f1 = []
    f2 = []
    for i in range(n):
        f1.append(arr[i])
        f2.append(GetSumOfDigit(arr[i]))
    return ((sum(f2)%10) - (sum(f1)%10))

n = int(input("Enter size: "))
arr = list(map(int, input("Enter elements: ").split()))
print(DifferenceSumOfDigits(arr, n))
```

```
Enter size: 10
Enter elements: 11 14 16 10 9 8 24 5 4 3
-4
```

Question4:



Problem statement

You are required to implement the following function:

```
int LargeSmallSum(int arr[], int length);
```

The function accepts an integer array 'arr' of size 'length' as its argument. You are required to find and return the sum of second large element from even position and second smallest element from odd position of given 'arr'.

Assumptions:

- All array elements are unique
- Treat the 0^{th} index as even

Note:

- Return 0, if arr is empty.
- Return 0, if array length is 3 or less than 3.

Example:

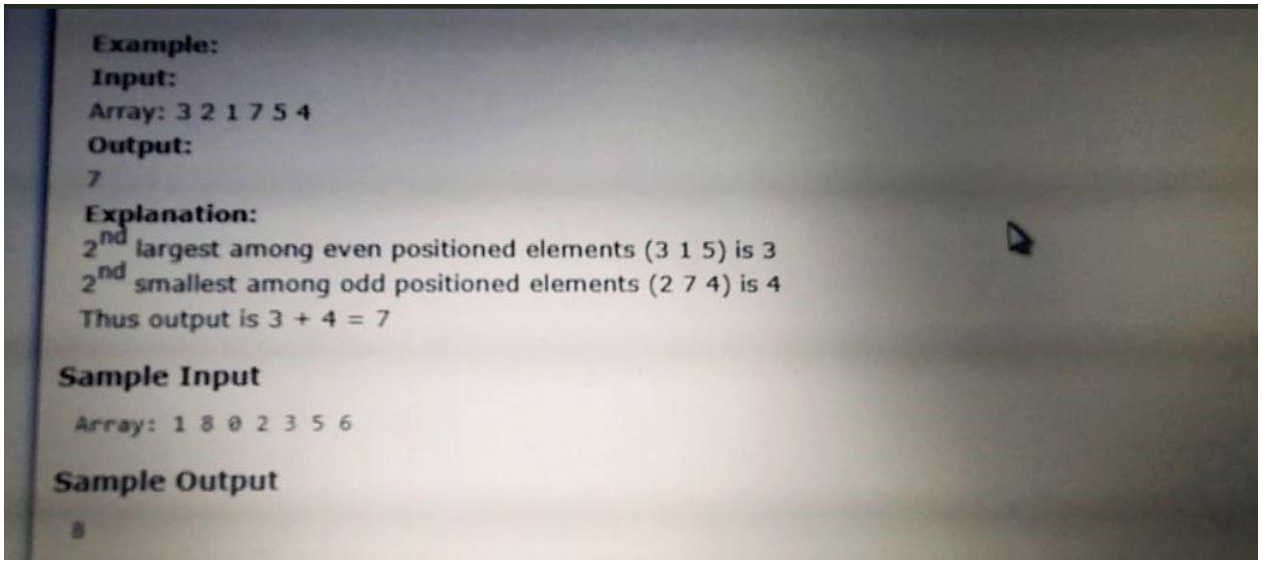
Input:

Array: 3 2 1 7 5 4

Output:

7

Explanation:



Solution:

```
def LargeSmallSum(arr, size):
    o = []
    e = []
    for i in range(size):
        if(i%2==0):
            e.append(arr[i])
        else:
            o.append(arr[i])
    e.sort()
    o.sort()
    return (o[1] + e[len(e)-2])

arr = list(map(int, input("Enter elements: ").split()))
if(len(arr)>3):
    print(LargeSmallSum(arr, len(arr)))
else:
    print(0)
```

```
Enter elements: 1 8 0 2 3 5 6
8
```