

#Q. largest subarray

Read only region start

```
class UserMaincode(object):  
    @claassmethod  
    def largestSubarray(cls, input1, input2):  
        '''  
        input1 : int  
        input2 : int[]  
  
        Expected return type : int  
        '''  
  
        # Read only region end  
        # Write code here  
        Hash={0:-1}  
        total_sum = 0  
        for i in range(input1):  
            if input2[i]==0:  
                input2[i]=-1  
        pre_sum=0  
        res=0  
        for i in range(input1):  
            pre_sum+=input2[i]  
            if pre_sum in Hash:  
                res = max(res,i-Hash[pre_sum])  
            else:  
                Hash[pre_sum]=i  
        return res
```

#Q. Student report

Read only region start

```
class UserMaincode(object):  
    @claassmethod  
    def SortStudentMarks(cls, input1, input2, input3):  
        '''  
        input1 : int  
        input2 : int  
        input3 : int[-]  
  
        Expected return type : int[]  
        '''  
  
        # Read only region end  
        # Write code here  
        avg_list = []  
        list = []
```

```

for i in range(input2):
    sum1 = 0
    for j in input3:
        sum1 += j[i]
    avg_list.append(sum//input1)
minimum = min(avg_list)
index1 = avg_list.index(minimum)

for i in range(input1):
    del input3[i][index1]
    a = input3[i]
    sum1 = sum(a)
    list.append(sum1)
return list

```

#Q. Student report

Read only region start

```

class UserMaincode(object):
    @classmethod
    def SortStudentMarks(cls, input1, input2, input3):
        '''
        input1 : int
        input2 : int
        input3 : int[-]

        Expected return type : int[]
        '''

```

Read only region end

Write code here

```

average = []
for i in range(input2):
    plus = 0
    for y in range(input1):
        plus += input3[y][i]
    average.append(plus)

value = 1000000000
for i in average:
    if value > i:
        value = i
index_min = average.index(value)
scores = []
for i in input3:
    i.pop(index_min)

for i in input3:

```

```
scores.append(sum(i))
```

```
return scores
```

```
#Q.2 on ={'+', '-', '/', '*'}
```

```
class UserMainCode(object):
```

```
    @classmethod
```

```
    def letter(cls, input1):
```

```
        '''
```

```
        input1:string
```

```
        Expected return type: int
```

```
        '''
```

```
        s=input1
```

```
        n=len(s)
```

```
        stack=[]
```

```
        a={'+', '-', '/', '*'}
```

```
        for i in range(n):
```

```
            if s[i].isdigit():
```

```
                stack.append(int(s[i]))
```

```
            elif s[i]=='+':
```

```
                op1=stack.pop()
```

```
                op2=stack.pop()
```

```
                stack.append(int(op2)+int(op1))
```

```
            elif s[i]=='-':
```

```
                op1=stack.pop()
```

```
                op2=stack.pop()
```

```
                stack.append(int(op2)-int(op1))
```

```
            elif s[i]=='/':
```

```
                op1=stack.pop()
```

```
                op2=stack.pop()
```

```
                stack.append(int(op2)/int(op1))
```

```
            elif s[i]=='*':
```

```
                op1=stack.pop()
```

```
                op2=stack.pop()
```

```
                stack.append(int(op2)*int(op1))
```

```
        return stack.pop()
```

```
#Q.4 sum of letters
```

```
class UserMaincode(object):
```

```
    @claassmethod
```

```
    def letter(cls, input1):
```

```
        '''
```

```
        input1 : string
```

```
        Expected return type : int
```

```
        '''
```

```
# Read only region end
# Write code here
scores = {}
scores['A'] = 0
scores['B'] = 1
for j in range(ord('A')+2, ord('Z')+1):
    scores[chr(j)] = scores[chr(j-1)] + scores[chr(j-2)]
total = 0
for j in input1:
    total += scores[j]
return total
```