

# Algorithm Design Term Project

---

Instructed by Asst. Prof. Dr. Thitipong Tanprasert

# Problem

Grand Theft Array V | Problem 2000

---

# Problem

---

**Difficulty** : 137

**Solved by** : Navamongkol Tongta 6213387

**Problem Author** : Ilya Kuchumov. (Prepared by Kirill Devyatkin)

**Problem Source** : Ural Regional School Programming Contest 2013

# Question

---

A long anticipated game called *Grand Theft Array V* is about to appear in shops! What, haven't you heard of it? Then we must tell you all about it!

The gameplay in *GTA V* takes place on a one-dimensional array of integers. The game has two players, each player has his own specified starting position. Players move in turns. During each turn a player takes a number written in his current cell, then writes a zero in it and moves to the left or right adjacent cell. Naturally, the player cannot move beyond the boundaries of the array. At some moment of time two players can be located in the same cell. A player's score is the sum of all numbers he earns during the game. The game ends when zeroes are written in all cells of the array.

Now please calculate the maximum number of points the first and the second player can get (the first player moves first, naturally), if they play optimally well, that is, if they try to maximize their score and if there are multiple variants of maximizing one's own score, they try to minimize the opponent's score.

# Input / Output

---

Input	Output
10 1 2 3 4 5 6 7 8 9 0 4 8	21 24

**Solution**

---

# Solution

---

The first player **move first** naturally means first player can will move more than second player.

There are **3 cases** for starting points :

- starting point of  $X > Y$
- starting point of  $X < Y$
- starting point of  $X == Y$



$$X > Y$$

---

Score	1	2	3	4	5	6	7	8	9
Position			X				Y		

X = sum[]  
Y = sum[]

Score	1	2	3	4	5	6	7	8	9
Position			X				Y		

$\text{Len}[3,4,5,6,7] \% 2 \neq 0$

The size of range X to Y is odd number



$$\underline{X > Y}$$

X move first and get score digit more than Y

Score	1	2	3	4	5	6	7	8	9
Position			X				Y		

X = sum[3,4,5]  
Y = sum[6,7]

Score	1	2	3	4	5	6	7	8	9
Position			X				Y		

X = sum[1,2,3,4,5]  
Y = sum[6,7,8,9]

$$X < Y$$

---

Score	1	2	3	4	5	6	7	8	9
Position			Y				X		

X = sum[]  
Y = sum[]

Score	1	2	3	4	5	6	7	8	9
Position			Y				X		

$\text{Len}[3,4,5,6,7] \% 2 \neq 0$

The size of range X to Y is odd number

X < Y

X move first and get score digit more than Y

Score	1	2	3	4	5	6	7	8	9
Position			Y				X		

X = sum[5,6,7]  
Y = sum[3,4]

Score	1	2	3	4	5	6	7	8	9
Position			Y				X		

X = sum[5,6,7,8,9]  
Y = sum[1,2,3,4]

$$\underline{X = Y}$$

Score	1	2	3	4	5	6	7	8	9
Position					X,Y				

Front = sum[]  
Back = sum[]

Score	1	2	3	4	5	6	7	8	9
Position					X,Y				

Front = sum[1,2,3,4]  
Back = sum[6,7,8,9]

$$X = Y$$

---

$$\text{Front} = \text{sum}[1, 2, 3, 4] = 10$$

$$\text{Back} = \text{sum}[6, 7, 8, 9] = 30$$

Because X is move first

$$X = \max(\text{Front}, \text{Back}) + \text{Last}$$

$$Y = \min(\text{Front}, \text{Back})$$

Score	1	2	3	4	5	6	7	8	9
Position					X, Y				

$$X = 30 + 5 = 35$$

$$Y = 10$$

Code

---



# Code

---

```
4  n = int(input())
5  array = list(map(int, input().split()))
6  start = list(map(int, input().split()))
7
```

N is size of array.  
Array is the list of number.  
Start is the starting point of X and Y

```
11 x = start[0]-1
12 y = start[1]-1
13
14 temp_x = 0
15 temp_y = 0
```

X and Y is the starting point as a index.  
Temp\_x and Temp\_y is the total score.

# Code - logic of 3 cases

---

```
25  √ if(x>y):
26
27  √     if len(array[y:x+1]) % 2 == 0:
28         temp_x += sum(array[x+1:])+sum(array[int(((x+y)/2)+1]:x+1])
29         temp_y += sum(array[:y])+sum(array[y:int((x+y)/2)+1])
30  √     else:
31
32         temp_x += sum(array[x+1:])+sum(array[int(((x+y)/2)):x+1])
33         temp_y += sum(array[:y])+sum(array[y:int((x+y)/2)])
34
35  √ if (x<y):
36
37  √     if len(array[x:y+1]) % 2 == 0:
38         temp_x += sum(array[:x+1])+sum(array[x+1:int((x+y)/2)+1])
39
40  √     else:
41         temp_x += sum(array[:x])+sum(array[x:int((x+y)/2)+1])
42
43
44     temp_y += sum(array[y+1:])+sum(array[int(((x+y)/2)+1]:y+1])
45
46
```

```
29  if(x==y):
30     front = sum(array[:x])
31     back = sum(array[x+1:])
32     temp_x += max(front,back)+array[x]
33     temp_y += min(front, back)
```

# Approval

---

ID	Date	Author	Problem	Language	Judgement result	Test #	Execution time	Memory used
9461023	17:09:37 15 Sep 2021	<a href="#">navamongkol</a>	<a href="#">2000. Grand Theft Array V</a>	Python 3.8 x64	Accepted		0.109	13 068 KB

# References

---

- <https://acm.timus.ru/forum/thread.aspx?id=33740&upd=635996979450040804>

Thanks You

---