

GEPROC-ULA

Grupo de Entrenamiento de Programación Competitiva ULA

Universidad de Los Andes



Problema Ejemplo:

1) Maximum triplet sum in array

Given an array of size **N**, the task is to find maximum triplet sum(MTS) in the array. A MTS is calculated by the sum of 3 values **V** in the array.

$$3 \leq N \leq 100$$

$$-100000 \leq V \leq 100000$$

Examples:

Input : arr[] = 1, 2, 3, 0, -1, 8, 10

Output : 21

$$10 + 8 + 3 = 21$$

Input : arr[] = 9, 8, 20, 3, 4, -1, 0

Output : 37

$$20 + 9 + 8 = 37$$

Solución

Inconvenientes en la solución.

- ¿Cómo leer correctamente la entrada.?
- ¿Qué tipos de datos utilizar para representar la información?
- ¿Cuál es la salida?
- ¿Cómo mostrar la salida correctamente?
- ¿Cómo validar rápidamente la solución?

```
1  int main(int argc, char* argv[]){
2      int N;
3      int val;
4      scanf("%d",&N); //leemos N (Sin validar)
5      int array[N];
6
7      for(int i=0; i < N;i++)
8      {
9          scanf("%d",&val); //Leemos los valores V (Sin validar)
10         array[i] = val;
11     }
12
13     std::sort(array,array+N); //Ordenamos el arreglo (STL)
14
15     //Sumamos para hallar el MTS
16     int sum = 0;
17     for(int i = N-1; i >= N-3;i--)
18         sum += array[i];
19
20     //Mostramos la salida
21     printf("%d\n",sum);
22     return 0;
23 }
```

Testing

¿Cómo probar eficientemente la solución?

- Generar bancos de pruebas.
- Por cada ejemplo del problema, generar un archivo en el cual se registre la entrada. Ejm: **test.txt**.
- Para probar utilizamos **./ejecutable < test.txt**
- Valida la salida con la del ejemplo.
- Si pasa todas las pruebas, es muy probable que este bien.
- No obstante, siempre es bueno generar tus casos de prueba.

test.txt

7

1

2

3

0

-1

8

10

./ejecutable < test.txt

21

Actividad

- Realice una solución alternativa del problema anterior.
- Diseñe los tests correspondientes.
- La solución es ¿mejor o peor?. ¿Qué significa mejor o peor?