Xbox Live Points

|  |  |  |  |
| --- | --- | --- | --- |
| Puntos |  | Límite de memoria | 32MB |
| Límite de tiempo (caso) | 1s | Límite de tiempo (total) | 60s |

**Descripción**

Con la salida del nuevo Xbox One, la compañía está llevando a cabo una campaña para promocionar el nuevo producto. Para ello quiere obsequiar puntos para Xbox Live de una manera muy peculiar. Primero, se van a elegir al azar NN cajas muy bonitas, que en su interior contienen un número anotado en un papelito, ese número indica la cantidad de puntos que te van a regalar si eliges esa caja. Para hacer las cosas más interesantes, te dicen que ese número puede ser tanto positivo, negativo o cero. Pero descuida, te aseguran que si eliges bien, nunca te llevarás puntos en contra.

Después, las NN cajas van a ser colocadas en una fila y podrás tomar la cantidad de cajas que quieras siempre y cuando sean cajas consecutivas.

Luego, te darán la oportunidad de saber de antemano el valor que tienen las cajas para que, un segundo después, tomes las cajas que más te gusten (o que más puntos te den) respetando la restricción. Cada número estará entre -100 y 100.

Como buen olímpico, quieres maximizar el número de puntos y para ello vas a escribir un programa que te diga cuál es ese máximo número de puntos que puedes conseguir en esta gran oportunidad.

Dados NN números, decir cuál es el máximo de puntos que puedes escoger cumpliendo con las condiciones.

**Entrada**

Línea 1: Un número NN que representa el número de cajas. Línea 2: N números, donde el ii-ésimo número representa los puntos en la caja ii.

**Salida**

Línea 1: Un número que diga cuál es la suma máxima de puntos que puedes obtener.

**Ejemplo**

| **Entrada** | **Salida** | **Descripción** |
| --- | --- | --- |
| 10  31 -41 59 26 -53 58 97 -93 -23 84 | 187 | La suma máxima se puede formar tomando de la caja 2 a la caja 6, contando a partir de 0. |

**Límites**

* Para un 10% de los casos: 1≤N≤1001≤N≤100
* Para un 30% de los casos independiente al anterior: 1≤N≤100001≤N≤10000
* Para el resto de los casos: 1≤N≤10000001≤N≤1000000

*Fuente: El Yorch*

Problema subido por: [neoinyawhe](https://omegaup.com/profile/neoinyawhe/)

| **ID** | **Status** | **Porcentaje** | **Penalty** | **Lenguaje** | **Memoria** | **Tiempo** | **Detalles** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| [Nuevo envío](https://omegaup.com/arena/problem/Xbox-Live-Points#problems/new-run) | | | | | | | | |
| 48a07684 | Error de compilación | — | — | cpp | 0.00 | 0.00 |  |  |
| 49d51ab5 | Respuesta correcta | 100.00% | 0 | cpp | 6.89 | 1.03 |  |  |
| **Envíos** | | | | | | | |

<https://omegaup.com/arena/problem/Xbox-Live-Points#problems>

#include<iostream>

#include <stdio.h>

using namespace std;

int maxSubArraySum(int a[], int size)

{

int max\_so\_far = a[0];

int curr\_max = a[0];

for (int i = 1; i < size; i++)

{

curr\_max = max(a[i], curr\_max+a[i]);

max\_so\_far = max(max\_so\_far, curr\_max);

}

return max\_so\_far;

}

/\* Driver program to test maxSubArraySum \*/

int main()

{

int N;

scanf("%d", &N);

int arr[N];

for(int i =0; i < N; i++) {

scanf("%d", &arr[i]);

}

printf("%d\n", maxSubArraySum(arr, N));

system("pause");

return 0;

}