

ENUNCIADO DEL PROBLEMA

Hay N hoteles a lo largo de la hermosa costa adriática. Cada hotel tiene su valor en Euros.

Sroljo ha ganado M Euros en la lotería. Ahora quiere comprar una secuencia de hoteles consecutivos, de modo que la suma de los valores de estos hoteles consecutivos sea lo mayor posible, pero no mayor que M.

Debe calcular este mayor valor total posible.

Entrada

$(1 \leq N \leq 300\,000, 1 \leq M < 2^{31})$.

En la siguiente línea hay N números naturales menores que 10^6 , que representan los valores del hotel en el orden en que se encuentran a lo largo de la costa.

Salida

Imprima el número requerido (será mayor que 0 en todos los datos de la prueba).

URL DE GITLAB O GITHUB PROYECTO EN FORMA PRIVADA, USUARIO madarme@ufps.edu.co, con rol mantener

[Ejercicios Proyecto ADA / Ejercicio 1 - Hotels Along the Croatian Coast · GitLab](#)

MÉTODO 1

```
public int metodoCandido(int cantHoteles, int dinero, int []hoteles) {
    int suma, max = 0;

    for(int i = 0; i<cantHoteles; i++) {
        suma = hoteles[i];

        for(int j = i; j<cantHoteles; j++) {
            if (j≠i) {
                suma += hoteles[j];
            }
            if(suma ≤ dinero && suma > max) {
                max = suma;
            }
        }
    }
    return max;
}
```

Eficacia

El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas.

Instancia 1:

CANTIDAD DE HOTELES: 1
CANTIDAD DE DINERO: 1
PRECIO DE LOS HOTELES: 1

	RESULTADO MÉTODO CÁNDIDO: 1
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESULTADO MÉTODO CÁNDIDO: 63</p>
	<p>Instancia 3:</p> <p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO MÉTODO CÁNDIDO: 241995</p>
Eficiencia	<p>El método no es el más eficaz ya que analiza los elementos múltiples veces, haciendo que avance muy lentamente en la lista de precios de los hoteles y requiere de esto para conseguir el resultado correcto; esto se ve reflejado en la instancia N°3 donde hay una mayor cantidad de elementos y esta falencia toma mayor peso.</p>
	Instancia 1:

	<p>CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1</p> <p>RESULTADO MÉTODO CÁNDIDO: 1</p> <p>Tiempo del proceso:81 microsegundos</p>
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESULTADO MÉTODO CÁNDIDO: 63</p> <p>Tiempo del proceso:27 microsegundos</p>
	<p>Instancia 3:</p> <p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO MÉTODO CÁNDIDO: 241995</p>

	Tiempo del proceso: 2822 microsegundos
Correctitud	El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles
	<p>Instancia 1:</p> <p>CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1</p> <p>RESPUESTA ESPERADA: 1 RESULTADO MÉTODO CÁNDIDO: 1</p>
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESPUESTA ESPERADA: 63 RESULTADO MÉTODO CÁNDIDO: 63</p>
	Instancia 3:

	<p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO ESPERADO: 241995 RESULTADO MÉTODO CÁNDIDO: 241995</p>
Completitud	<p>El método es completo, ya que la lógica usada para hallar el grupo de hoteles consecutivos, aunque es lento asegura siempre un resultado correcto, si es que este existe</p>
	<p>Instancia 1:</p> <p>CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1</p> <p>RESULTADO ESPERADO:1 RESULTADO MÉTODO CÁNDIDO: 1</p>
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p>

	RESULTADO ESPERADO: 63 RESULTADO MÉTODO CÁNDIDO: 63
	Instancia 3: CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES: RESULTADO ESPERADO: 241995 RESULTADO MÉTODO CÁNDIDO: 241995

MÉTODO 2

```
public int metodoOptimo(int cantHoteles, int dinero, int []hoteles) {
    int izq = 0, der = 0, suma = 0, max = 0;

    while (der < cantHoteles) {
        if (suma + hoteles[der] ≤ dinero) {
            suma += hoteles[der];
            if(suma > max) {
                max = suma;
            }
            der++;
        } else {
            suma -= hoteles[izq];
            izq++;
        }
    }

    return max;
}
```

Eficacia

El método es eficaz, puesto a que tiene una solución para todas las entradas válidas ingresadas

Instancia 1:

CANTIDAD DE HOTELES: 1
CANTIDAD DE DINERO: 1
PRECIO DE LOS HOTELES: 1

	RESULTADO MÉTODO ÓPTIMO: 1
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESULTADO MÉTODO ÓPTIMO: 63</p>
	<p>Instancia 3:</p> <p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO MÉTODO ÓPTIMO: 241995</p>
Eficiencia	El método es eficiente ya que solo necesita de un recorrido para llegar a un resultado correcto, evitando las múltiples validaciones de los mismo valores
	<p>Instancia 1:</p> <p>CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1</p>

	<p>RESULTADO MÉTODO ÓPTIMO: 1 Tiempo del proceso: 25 microsegundo</p>
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESULTADO MÉTODO ÓPTIMO: 63 Tiempo del proceso: 18 microsegundos</p>
	<p>Instancia 3:</p> <p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO MÉTODO ÓPTIMO: 241995 Tiempo del proceso: 126 microsegundos</p>
Correctitud	El algoritmo es correcto, ya que da un resultado esperado para todas las entradas posibles
	Instancia 1:

	<p>CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1</p> <p>RESULTADO ESPERADO: 1 RESULTADO MÉTODO ÓPTIMO: 1</p>
	<p>Instancia 2:</p> <p>CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3</p> <p>RESULTADO ESPERADO: 63 RESULTADO MÉTODO ÓPTIMO: 63</p>
	<p>Instancia 3:</p> <p>CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES:</p> <p>RESULTADO ESPERADO: 241995 RESULTADO MÉTODO ÓPTIMO: 241995</p>
Compleitud	El método es completo, ya que la lógica usada para hallar el grupo de hoteles consecutivos asegura siempre

	un resultado correcto, si es que este existe.
	Instancia 1: CANTIDAD DE HOTELES: 1 CANTIDAD DE DINERO: 1 PRECIO DE LOS HOTELES: 1 RESULTADO MÉTODO ÓPTIMO: 1
	Instancia 2: CANTIDAD DE HOTELES: 8 CANTIDAD DE DINERO: 63 PRECIO DE LOS HOTELES: 8, 32, 1, 5, 4, 9, 12, 3 RESULTADO MÉTODO ÓPTIMO: 63
	Instancia 3: CANTIDAD DE HOTELES: 1.000 CANTIDAD DE DINERO: \$242.000 PRECIO DE LOS HOTELES: RESULTADO MÉTODO CÁNDIDO: 241995