

## Ejercicio 1 (2 puntos)

Dado un vector de enteros  $v$  de tamaño mayor o igual que cero, se define un pico como un valor  $v[i]$  tal que  $v[i - 1] < v[i]$  y  $v[i] > v[i + 1]$ ; en particular, un pico no puede aparecer al comienzo o al final del vector. Se pide:

1. Define un predicado  $pico(v, n, i)$  que sea verdadero si el vector  $v[0..n - 1]$  tiene un pico en  $i$ .
2. Especifica formalmente una función  $menorPico$  que reciba un vector de enteros y devuelva *verdadero* y el valor del menor pico en un vector o *falso* si no hay ninguno.
3. Implementa la función  $menorPico$  mediante un algoritmo iterativo eficiente.
4. Escribe la precondition del bucle principal del algoritmo y un invariante que permita probar que es correcto.

### Entrada

La entrada consiste en una serie de casos de prueba. Cada caso de prueba consiste en dos líneas: la primera indica la longitud del vector y la segunda contiene sus elementos. La entrada termina cuando la longitud del vector es 0.

### Salida

Para cada caso de entrada se muestra en una línea el valor del vector en el menor pico o no hay si no hay picos en el vector.

### Entrada de ejemplo

```
5
1 4 8 3 2
5
1 4 3 7 2
6
1 2 3 4 5 6
1
6
8
-3 -5 -2 -6 -5 -4 -7 -1
0
```

### Salida de ejemplo

```
8
4
no hay
no hay
-4
```

## Ejercicio 2 (2 puntos)

Queremos encontrar la raíz de un polinomio de cuarto grado  $ax^4 + by^3 + cx^2 + dx + e$  en un cierto intervalo  $[ini, fin]$ . Se entiende por raíz de un polinomio un valor de  $x$  que hace cero el polinomio. Se garantiza que el polinomio es estrictamente creciente o estrictamente decreciente en el intervalo dado y, que siempre existe una única raíz en el intervalo. Se pide:

1. Implementar una función recursiva que resuelva el problema utilizando la técnica de divide y vencerás.
2. Escribir la recurrencia que nos permite calcular el coste de la función implementada e indicar su orden de complejidad.

### Entrada

La primera línea de la entrada indica el número  $n$  de polinomios con los que se va a trabajar. Cada caso consta de 7 valores, los 5 primeros son los coeficientes del polinomio  $(a, b, c, d, e)$  y los dos últimos el inicio y el final del intervalo en el que se debe buscar la raíz del polinomio  $[ini, fin]$ . Los coeficientes del polinomio son números enteros y los límites del intervalo números reales con dos decimales.

### Salida

Para cada caso de prueba se escribe en una línea el valor de la raíz del polinomio en el intervalo dado **con dos decimales**.

El problema opera con números reales por lo que puede haber errores de precisión en los resultados obtenidos. Si los resultados del programa no difieren en más de 0.01 de los resultados del juez, éste los dará por válidos.

### Entrada de ejemplo

```
6
0 0 0 4 2 -10.00 10.00
0 0 1 -5 0 1.50 10.00
0 0 0 7 1 -1.00 1.00
0 0 1 5 4 -3.50 5.20
1 0 -10 0 9 2.00 4.00
1 -4 -1 16 -12 -5.00 -1.00
```

### Salida de ejemplo

```
-0.50
5.00
-0.14
-1.00
3.00
-2.00
```

## Ejercicio 3 (3 puntos)

Para celebrar el cumpleaños de Rodrigo vamos a pasar el día en un parque acuático. Debemos elegir con cuidado las atracciones en las que montarán los niños para que disfruten al máximo del día pero sin mojarse demasiado para no caer enfermos. Para cada atracción conocemos lo que los niños disfrutarán en ella, el tiempo que permanecerán jugando y si es acuática o no, es decir, si moja completamente la ropa o solo salpica. Queremos maximizar lo que van a disfrutar, teniendo en cuenta que permaneceremos en el parque un tiempo limitado. Para que no se pongan enfermos, el número de atracciones en que los niños montan y se mojan completamente no debe superar al acabar el día el número de atracciones en que montan y se salpican. Como podemos suponer, no quieren repetir ninguna atracción.

Se pide diseñar un algoritmo de vuelta atrás que maximice la satisfacción obtenida:

1. Describe el árbol de exploración y la forma del vector solución que se emplean en la solución del problema.
2. Implementa el algoritmo.
3. Piensa también en una estimación del coste que se pueda utilizar: aunque no la implementes, explícala.

### Entrada

La entrada consta de una serie de casos de prueba. Cada caso comienza con una línea en la que se indica el número de atracciones del parque,  $n$ , y el tiempo que permaneceremos en él. En las  $n$  líneas siguientes se muestra el nombre de la atracción, el tiempo que dura, la satisfacción que proporciona a los niños y si es acuática o no. El final de los datos se marca con dos ceros. El número de atracciones del parque cumple:  $0 < n < 30$  y el tiempo que permaneceremos en el parque es un entero positivo. El nombre de la atracción es una cadena de caracteres sin blancos, el tiempo que dura cada atracción y la satisfacción que proporciona son números enteros positivos. Por último, un 1 si la atracción es completamente acuática y 0 si solo salpica.

### Salida

Para cada caso de prueba se escribe la satisfacción total que se puede conseguir, seguido de las atracciones seleccionadas. Si no se puede montar en ninguna atracción se escribirá NINGUNA.

### Entrada de ejemplo

```
3 10
Tobogan 6 100 0
Piscina 5 50 0
Rio 5 60 0
4 15
Playa 10 1 0
Tobogan 4 2 1
Turbulencias 8 5 1
Lago 2 1 0
5 5
Playa 2 4 0
Rio 2 4 0
Olas 4 8 1
Tobogan 3 8 1
Turbulencias 1 4 1
2 5
Tubo 3 10 1
Olas 2 20 1
0 0
```

**Salida de ejemplo**

110 Piscina Rio 6 Turbulencias Lago 12 Rio Tobogan NINGUNA
---