

Estructuras de Datos y Algoritmos
Grado en Ingeniería del Software. Grupos E y F
Examen Parcial, 24 de enero de 2018.

Normas de realización del examen

1. Debes programar soluciones para cada uno de los tres ejercicios, probarlas y subirlas al juez. El juez se utiliza exclusivamente para la entrega del ejercicio, y no realiza ninguna prueba adicional a los casos del ejemplo. Por lo tanto el veredicto del juez no es significativo y no debe tomarse como que el programa es correcto.
2. Para que los problemas estén presentados deben estar subidos al juez, aunque no estén correctos. La dirección es [http://exacrc/domjudge/team](http://exacrc.domjudge/team). Debe escribirse la dirección completa.
3. En el juez te identificarás con el nombre de usuario y contraseña que has recibido al comienzo del examen. El nombre de usuario y contraseña que has estado utilizando durante la evaluación continua **no** son válidos.
4. Escribe tu nombre y apellidos, el laboratorio y el puesto en el que estás y el usuario que vas a utilizar en el juez de examen en la primera línea de cada fichero que subas al juez.
5. Puedes descargar el fichero <http://exacrc/123.zip> que contiene material que puedes utilizar para la realización del examen (plantilla de código fuente y ficheros de texto con los casos de prueba de cada ejercicio del enunciado).
6. Para obtener la máxima puntuación, las soluciones deberán seguir los criterios exigidos a los ejercicios durante el curso (en cuanto a eficiencia, simplicidad, buena programación, comentarios etc.)
7. Tus soluciones serán evaluadas por el profesor independientemente del veredicto del juez automático. Para ello, el profesor tendrá en cuenta exclusivamente el último envío que hayas realizado del ejercicio.
8. La especificación y la prueba de corrección del primer problema puede realizarse en Dafny. Para ello se proporciona una plantilla en la carpeta de la documentación y un resumen de la sintaxis de Dafny. El programa en Dafny se traducirá a C++ y se pondrá entre comentarios todo el código realizado en Dafny antes de subir el programa al juez.
9. Los costes que se piden de los algoritmos se escribirán en un comentario antes de la cabecera de la función.

Ejercicio 1

Especifica, diseña e implementa un algoritmo que dado un vector de números enteros encuentre la secuencia más larga de números consecutivos crecientes. Si existen dos secuencias de igual longitud elegirá la que se encuentre mas a la izquierda del vector. La secuencia quedará definida por su comienzo en el vector y su longitud. Para ello:

(1 punto) Define un predicado *consecutivos*, que dada una secuencia de valores, un comienzo y un final indique si los valores de la secuencia entre el comienzo y el final, ambos incluidos, son consecutivos.

Define un predicado *secuenciaMax* que dada una secuencia y una longitud indique si la longitud dada coincide con la longitud de la secuencia más larga de elementos consecutivos.

(1,5 puntos) Diseña e implementa un algoritmo que resuelva el problema.

(1 punto) Escribe un invariante y una función cota que permitan demostrar que la secuencia encontrada tiene todos sus elementos consecutivos.

(0,5 puntos) Justifica el coste del algoritmo

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de 2 líneas. En la primera se indica el número de elementos del vector. En la siguiente se indican los elementos del vector. El final de los casos se marca con el valor cero.

El vector tiene al menos un elemento.

Salida

Para cada caso de prueba se escribe en una línea el comienzo del intervalo seguido de su longitud.

Entrada de ejemplo

```
3
5 3 4
5
5 6 2 5 7
9
-3 -2 -1 6 3 4 5 2 3
10
5 3 4 1 2 3 -2 4 6 7
4
5 3 5 7
0
```

Salida de ejemplo

```
1 2
0 2
0 3
3 3
0 1
```

Ejercicio 2

(1,5 puntos) Implementa un programa recursivo que dado un vector de números enteros positivos y consecutivos, del que se han eliminado todos los elementos impares menos uno encuentre cuál es el número impar que queda en el vector. La implementación realizada debe ser eficiente.

(0.5 puntos) Indica el coste de la solución implementada. Justifícalo escribiendo la ecuación de recurrencia del problema y desplegándola.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de 2 líneas. En la primera se indica el número de elementos del vector una vez eliminados los elementos impares. En la siguiente se indican los elementos del vector. El final de los casos se marca con el valor cero.

El vector tiene al menos un elemento. Se garantiza que siempre existe un elemento impar en el vector.

Salida

Para cada caso de prueba se escribe en una línea el elemento impar del vector.

Entrada de ejemplo

```
4
6 8 9 10
4
1 2 4 6
4
4 5 6 8
4
4 6 8 9
5
1 2 4 6 8
5
2 3 4 6 8
1
3
2
5 6
2
6 7
0
```

Salida de ejemplo

```
9
1
5
9
1
3
3
5
7
```

Ejercicio 3

(4 puntos) Se está preparando el rodaje de una nueva película. El guión ya está listo y ha llegado el momento de seleccionar a los actores que interpretarán cada uno de los papeles. Se ha convocado a todos los actores disponibles a un *casting* y el director está asignando una puntuación a cada uno de ellos para cada papel. Por otra parte el productor ha elaborado una lista con lo que cobraría cada actor por participar en la película.

a) (3 puntos.) El director ha encargado que se obtenga el reparto de la película, de forma que se maximice la afinidad en su conjunto de los actores con los personajes, siempre dentro de un determinado presupuesto. La afinidad de cada actor con cada personaje se mide con la puntuación asignada en el casting y lo que se busca es maximizar la suma de las puntuaciones. Debe tenerse en cuenta también, que para simplificar la grabación, un actor sólo puede interpretar un papel y que no se admitirá un reparto con actores cuya puntuación en el papel asignado sea menor que una cierta constante.

b) (1 punto.) Mejora la implementación anterior cortando las llamadas recursivas que no producirán una puntuación mejor que la encontrada.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso de prueba consta de $n+2$ líneas. En la primera se indica el número de papeles de la obra, $n > 0$, seguido del número de actores, $k \geq n$, del presupuesto total con el que se cuenta, $p > 0$, y de la puntuación mínima requerida para poder actuar, $m > 0$. En la segunda, lo que cobra cada actor por participar en la película. En las n siguientes se indica para cada papel la puntuación otorgada por el director a cada uno de los actores. Si un actor no puede interpretar un papel su puntuación es cero. La entrada termina con una línea con cuatro ceros.

La cantidad que cobra cada actor es un número entero positivo. La puntuación de cada actor con cada papel es un entero mayor o igual que cero.

Salida

Para cada caso de prueba se escribe en primer lugar la puntuación total obtenida con el reparto elegido seguido del presupuesto necesario para este reparto. A continuación tantas líneas como papeles tenga la película. En cada línea se indica un papel y el actor seleccionado para interpretarlo, ambos datos separados por un carácter blanco. Los casos de prueba se escriben uno a continuación del otro.

Si no existe un reparto posible, se escribirá una única línea con puntuación cero y presupuesto cero.

Entrada de ejemplo

```
2 3 200 4
100 100 100
10 5 0
0 5 10
3 5 100 40
50 20 40 30 40
100 0 70 50 60
0 50 40 30 60
0 60 50 70 40
0 0 0 0
```

Salida de ejemplo

```
20 200
0 0
1 2
220 100
0 0
1 1
2 3
```