Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Problemas propuestos en la II Olimpiada Informática de Santa Cruz de Tenerife

Enero, 2024

Problemas propuestos en la II Olimpiada Informática de Santa Cruz de Tenerife

Volumen 1

Editores

Rafael Herrero-Álvarez, Universidad de La Laguna.
Coromoto León, Universidad de La Laguna.

Fecha de edición

29 de enero de 2024

2024. Universidad de La Laguna. Escuela Superior de Ingeniería y Tecnología. Departamento de Ingeniería Informática y de Sistemas.

Preámbulo

Este documento recoge los enunciados de los problemas propuestos en la *II Olimpiada Informática de Santa Cruz de Tenerife* (OITF), un concurso individual de programación algorítmica para estudiantes de educación secundaria, bachillerato o grado medio, en el que puede participar cualquier centro educativo de esta provincia Canaria.

El objetivo del evento es seleccionar a la persona que junto con el ganador de la *Olimpiada Informática de Las Palmas* constituirá la delegación de la comunidad autónoma de las Islas Canarias en la *Olimpiada Informática Española* (OIE).

En esta edición se propusieron un total de cinto ejercicios que debían resolverse en dos horas haciendo uso de un lenguaje de programación de alto nivel. Los lenguajes permitidos fueron `C++` y `Python`. Las soluciones en Python pueden consultarse en el repositorio de GitHub de la OITF[1].

[1] https://github.com/oliminforcanarias/Soluciones-OITF/tree/master

Organización

Escuela Superior de Ingeniería y Tecnología de la Universidad de La Laguna

Responsables

Coromoto León
Rafael Herrero-Álvarez

Colaboradores

Leopoldo Acosta Sánchez
Casiano Rodríguez León
Gara Miranda Valladares
Eduardo Manuel Segredo González
Carmen Elvira Ramos Domínguez
Luz Marina Moreno de Antonio
Iván Castilla Rodríguez

Centro de Cálculo
Aula Cultural de Pensamiento Computacional
Fundación Canaria General de la Universidad de La Laguna

Índice

Ejercicio 1. Control en el banco

1.1. Enunciado

Thalía quiere abrirse una cuenta bancaria, pero antes de hacerlo, necesita calcular los dígitos de control de su nueva cuenta bancaria para asegurarse de que la información proporcionada es correcta. El banco le ha dado las siguientes instrucciones para calcular los dígitos de control:

1. Para el primer dígito de control:
    - La primera cifra del **banco** se multiplica por 4.
    - La segunda cifra del **banco** se multiplica por 8.
    - La tercera cifra del **banco** se multiplica por 5.
    - La cuarta cifra del **banco** se multiplica por 10.
    - La primera cifra de la **entidad** se multiplica por 9.
    - La segunda cifra de la **entidad** se multiplica por 7.
    - La tercera cifra de la **entidad** se multiplica por 3.
    - La cuarta cifra de la **entidad** se multiplica por 6.
    - Se suman todos los resultados obtenidos.
    - Se divide entre 11 y se toma el resto de la división.
    - A 11 se le resta el resto de la división obtenido. Este resultado es el primer dígito de control, con la salvedad de que si es 10, el dígito es 1 y si es 11, el dígito es 0.
2. Para el segundo dígito de control:
    - La primera cifra de la cuenta se multiplica por 1.
    - La segunda cifra de la cuenta se multiplica por 2.
    - La tercera cifra de la cuenta se multiplica por 4.
    - La cuarta cifra de la cuenta se multiplica por 8.
    - La quinta cifra de la cuenta se multiplica por 5.
    - La sexta cifra de la cuenta se multiplica por 10.
    - La séptima cifra de la cuenta se multiplica por 9.
    - La octava cifra de la cuenta se multiplica por 7.
    - La novena cifra de la cuenta se multiplica por 3.
    - La décima cifra de la cuenta se multiplica por 6.
    - Se suman todos los resultados obtenidos.
    - Se divide entre 11 y se toma el resto de la división.

- A 11 se le resta el resto de la división obtenido. Este resultado es el segundo dígito de control, con la salvedad de que si es 10, el dígito es 1 y si es 11, el dígito es 0.

Thalía tiene los dígitos de su banco, entidad y cuenta, y necesita un programa para calcular los dígitos de control siguiendo estas instrucciones, y mostrar la numeración completa de la cuenta.

1.2. Entrada

La entrada se realiza por medio de la entrada estándar, teniendo en cuenta que se trata de 3 conjuntos de números enteros, distribuidos de forma que el primer conjunto (4 dígitos) representa el banco, el segundo conjunto (4 dígitos) representa la entidad en la que se abre la cuenta bancaria y el tercer conjunto (10 dígitos) representa el número de cuenta.

1.3. Salida

La salida tiene que constar de 4 conjuntos de números, siendo el primero el del banco (4 dígitos), el siguiente conjunto el de la entidad (4 dígitos), el tercer conjunto los dígitos de control (2 dígitos) y el cuarto conjunto el número de cuenta (10 dígitos).

En caso de que no aparezca la cantidad de dígitos correctos, tiene que salir el texto "IBAN incorrecto".

1.4. Entrada de ejemplo

```
1234 5678 9012345678
1234 2365 6958213620
1324 45 1234567890
1478 8596 1234567890
```

1.5. Salida de ejemplo

```
1234 5678 02 9012345678
1234 2365 36 6958213620
IBAN incorrecto
1478 8596 96 1234567890
```

Datos por la entrada estándar

Thalía sabe como espera el servidor recibir los datos y como sacarlos por la salida estándar.

En el caso de Python, una forma en la que puedes leer línea a línea es la siguiente:

```
import sys

for line in sys.stdin:
```

También, en el caso de Python, puedes leer todas las líneas de la siguiente manera:

```
import sys

lines = sys.stdin.readlines()
```

Para sacar datos por la salida estándar en Python:

```
print()
```

En el caso de C++, para leer desde la entrada estándar línea a línea puedes hacerlo de la siguiente manera:

```
#include <iostream>

int main() {
    while (std::getline(std::cin, line)) {
    }
}
```

También, en el caso de C++, puedes leer todas las líneas de la siguiente manera, teniendo en cuenta que cada vez que ejecutes *cin* se leerá la línea completa.

```
#include <iostream>

int main(){
    std::cin >> ...;
}
```

Para sacar datos por la salida estándar en C++:

```
std::cout << ...;
```

Ejercicio 2. Letras combinadas

2.1. Enunciado

Jorge, un apasionado de las letras, tiene un interesante desafío frente a él: quiere escribir todas las posibles combinaciones de letras usando el alfabeto desde la 'a' hasta la 'z', formando secuencias progresivas hasta llegar a una palabra completa. Por ejemplo, si la palabra objetivo es "hola", tiene que comenzar listando cada letra del alfabeto por separado, y luego empezar a añadir letras progresivamente hasta formar la palabra completa. La secuencia comenzaría de la siguiente manera:

```
a
b
c
d
e
f
g
h
ha
hb
hc
hd
he
hf
hg
hh
hi
hj
hk
hl
hm
hn
ho
hoa...
```

Y así sucesivamente, hasta que finalmente llegue a formar la palabra "hola". Este método le permitirá explorar todas las combinaciones posibles de letras del alfabeto, desde una sola letra hasta llegar a la formación completa de la palabra deseada. Jorge planea utilizar este método para investigar y disfrutar de la belleza y la complejidad de las combinaciones de letras en el lenguaje.

2.2. Entrada

Se espera, por la entrada estándar, una palabra formada únicamente por los siguientes 26 caracteres: a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z.

2.3. Salida

La salida tendrá que mostrar todas las posibles combinaciones de caracteres, comenzando por la letra 'a', construyendo progresivamente la palabra de entrada.

2.4. Entrada de ejemplo

```
aire
```

2.5. Salida de ejemplo

```
a
aa
ab
ac
ad
ae
af
ag
ah
ai
aia
aib
aic
aid
aie
aif
aig
aih
aii
aij
aik
ail
aim
ain
aio
aip
aiq
air
aira
```

```
airb
airc
aird
aire
```

RECUERDA: el sistema espera recibir los datos por la entrada estándar.

Ejercicio 3. Límite en la llamada

3.1. Enunciado

Jéssica quiere calcular la cantidad de minutos que puede hablar por teléfono dentro de su plan de tarifa prepago, ya que su proveedor de servicios telefónicos tiene una tarifa que incluye dos componentes: un coste por el establecimiento de la llamada y un coste por cada minuto de llamada. Además, esta tarifa varía dependiendo de si la llamada se realiza a un teléfono fijo o a un teléfono móvil.

La tarifa se compone de los siguientes elementos:

- Coste de establecimiento de llamada a un teléfono fijo.
- Coste de establecimiento de llamada a un teléfono móvil.
- Coste por minuto de llamada a un teléfono fijo.
- Coste por minuto de llamada a un teléfono móvil.

Jéssica quiere determinar cuántos minutos de llamada puede permitirse dentro de su plan, dependiendo del saldo disponible.

3.2. Entrada

Los datos se proporcionarán por la entrada estándar, siendo 6 conjuntos:

1. Coste de establecimiento de llamada a un teléfono fijo (número decimal).
2. Coste de establecimiento de llamada a un teléfono móvil (número decimal).
3. Coste por minuto de llamada a un teléfono fijo (número decimal).
4. Coste por minuto de llamada a un teléfono móvil (número decimal).
5. Saldo disponible en la tarjeta prepago (número decimal).
6. Palabra que indica si se llama a fijo o a móvil ("fijo" | "movil").

3.3. Salida

La salida tendrá que ser un único número entero que represente la cantidad de minutos que se puede hablar por teléfono. Este número deberá truncarse en **TODOS** los casos.

En caso de que no hubiese saldo disponible, imprimir "Saldo insuficiente".

3.4. Entrada de ejemplo

```
0.23 0.15 0.44 0.09 11.22 fijo
0.23 0.15 0.44 0.09 11.22 movil
0.23 0.15 0.44 0.09 0.22 fijo
```

3.5. Salida de ejemplo

```
24
123
Saldo insuficiente
```

RECUERDA: el sistema espera recibir los datos por la entrada estándar.

Ejercicio 4. Viaje barato

4.1. Enunciado

Sofía está planeando un viaje por carretera y quiere calcular cuánto dinero puede ahorrar en combustible utilizando diferentes tipos de vehículos. Ella tiene información sobre varios vehículos, incluyendo su consumo de combustible (kilómetros por litro) y el coste por litro del combustible que lleva ese coche, ya que cada vehículo tiene un motor diferente.

Sofía quiere desarrollar un programa que le permita determinar qué vehículo le ahorrará más dinero en combustible durante su viaje y cuánta es la diferencia con el segundo más económico.

4.2. Entrada

Los datos se proporcionan por la entrada estándar, tratándose de 3 conjuntos numéricos:

1. Distancia en kilómetros (número decimal).
2. Cantidad de vehículos (número entero).
3. Vehículos. Cada uno de ellos aparecerá en una línea con los siguientes datos:
   a) Identificador único del vehículo (número entero).
   b) Cantidad de kilómetros que puede hacer por litro (número decimal).
   c) Precio del combustible que utiliza (número decimal).

Los datos de entrada nunca podrán ser 0 o inferior.

4.3. Salida

El formato de salida esperado consta de dos números, el primero es un entero con el identificador del vehículo más económico, y el segundo un número decimal (4 dígitos en la parte decimal) con la diferencia de ahorro respecto al segundo más económico.

4.4. Entrada de ejemplo

```
200.45
6
1 20.6 1.5
2 18 1.8
3 17.4 1.66
4 22.03 1.54
5 12 0.98
```

```
6 19 1.22

23.67
0

432.98
2
1 16.3 1.23
2 -8.9 2

432.98
3
1 16.3 1.23
2 8.9 2
3 5.4 2.1
```

4.5. Salida de ejemplo

```
6 1.1414

ERROR

ERROR

1 64.6262
```

RECUERDA: el sistema espera recibir los datos por la entrada estándar.

Ejercicio 5. Lanzamiento de aceitunas

5.1. Enunciado

En la localidad de Cieza (Murcia) abundan las aceitunas, tanto, que organizan anualmente el famoso concurso de lanzamiento de hueso de aceituna. Gabriel, un joven de la localidad y muy avispado con las matemáticas, se ha dado cuenta que todos los huesos rebotan siempre igual: cuando el hueso ha rebotado r veces, se para.

Cuando los participantes lanzan la aceituna, esta recorre k metros antes de la primera vez que rebota, luego la mitad $(k/2)$, luego un tercio $(k/3)$, luego un cuarto $(k/4)$, y así hasta que se para cuando ha rebotado r veces. Gabriel quiere saber cuántos metros tiene que lanzarlo inicialmente el participante para que alcance el objetivo final m.

Es decir: $k + \frac{k}{2} + \frac{k}{3} + ... + \frac{k}{r} = m$

Si no se puede realizar la operación porque no hay número entero posible, el programa debe devolver la palabra **ERROR**.

5.2. Entrada

Los datos se proporcionan por la entrada estándar, tratándose de 2 números enteros:

1. m: metros que recorre la aceituna al final.
2. r: veces que rebota la aceituna.

5.3. Salida

El formato de salida esperado consta de un número entero con la cantidad de metros **mínima** que debe lanzarse inicialmente la aceituna k.

5.4. Entrada de ejemplo

```
14 15
14 2
14 1
14 3
```

5.5. Salida de ejemplo

```
6
ERROR
14
8
```

RECUERDA: te será más fácil llegar a la solución si tienes en cuenta que se trabaja únicamente con enteros.

Versión adaptada del problema Lanzamiento de pelota.