Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Segundo Semestre de 2014

IIC 1103 - Introducción a la Programación

Midterm R ⋆ Tiempo: 180 minutos ⋆ Sin apuntes ⋆ Sin consultas

1. Escribe un programa en Python que encuentre el k-ésimo número *buena onda*. Un número buena onda es aquel que solo tiene como factores primos el 3, 5, y 7. Para esto:

   - Escribe una función que verifique si un número es primo.
   - Escribe una función que verifique si un número es buena onda.
   - Usando las funciones de los pasos anteriores, escribe un programa que permita a un usuario verificar si el número ingresado es buena onda.

2. Un alumno viaja todos los días de la universidad a la casa y viceversa pero se está dando cuenta que gasta mucha bencina y no entiende por qué, debe ayudarlo a encontrar un camino óptimo.

   El auto recorre la ciudad cuadra a cuadra en sólo cuatro direcciones posibles: 0, 1, 2 y 3; los cuales representan movimientos hacia arriba, abajo, derecha e izquierda, respectivamente.

   Escribe una función que reciba como parámetro un string que contiene valores 0,1,2 y 3 que representa el camino y retorne como resultado otro string que representa la optimización del camino.

   Por ejemplo:
   Si el string de entrada a la función es `02221213300031111` el/los camino/s optimizado/s son: `112`, `121` o `211`. La función debe retornar sólo uno de los posibles caminos, por ejemplo, `112`.

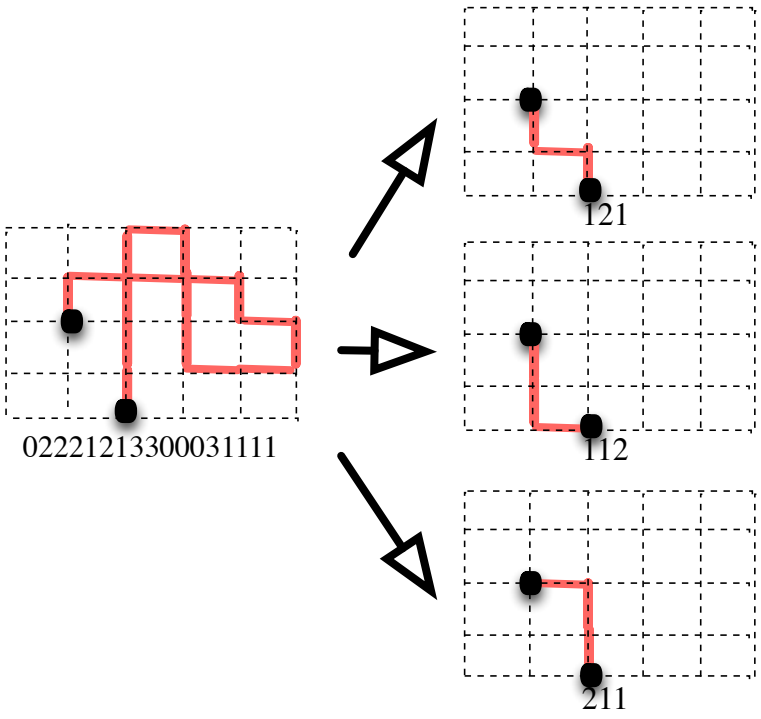


Figura 1: Ejemplo del camino.

Hint: Recuerda que ir una cuadra hacia arriba y luego hacia abajo dos cuadras representa un gasto de bencina innecesaria, seria mejor sólo ir hacia abajo una cuadra.

3. Crea un programa capaz de resolver el juego conocido como la *sopa de letras*. Este juego consiste en un tablero de $N \times M$ donde cada posición del tablero tiene una letra. El objetivo del juego es encontrar palabras dentro del tablero, ya sea en forma horizontal (de izquierda a derecha), vertical (de arriba a abajo) o diagonal (de esquina superior izquierda a esquina inferior derecha). Por ejemplo la Figura 2a) tiene una sopa de letras que contiene una palabra horizontal 2b), vertical 2c) y diagonal 2d).

C	A	S	A
E	R	G	H
I	O	A	M

(a)

C	**A**	**S**	**A**
E	R	G	H
I	O	A	M

(b)

C	**A**	S	A
E	**R**	G	H
I	**O**	A	M

(c)

C	A	S	A
E	**R**	G	H
I	O	**A**	M

(d)

Figura 2: Sopa de letras.

Su programa recibe 2 parámetros: La sopa de letras y la lista de palabras, ambos como strings. Para la sopa de letras, cada fila del tablero está separada por un `"\n"` (sin espacios), mientras que para la lista de palabras, cada una de ellas están separadas por un `"#"`. Por ejemplo, el tablero de la Figura 2 queda codificado como `T = ''CASA\nERGH\nIOAM''`, y su lista de palabras como `L = ''casa#aro#cra''`.

El programa debe ser capaz de detectar las palabras de L en T y mostrarlas en consola usando el siguiente formato: (*palabra*, *posición*, *dirección*), donde *palabra* es la palabra encontrada, *posición* es el índice donde comienza la palabra en el string T, y *dirección* puede ser: *"derecha"*, *"abajo"* o *"diagonal"*. Para el ejemplo, su programa debería imprimir:

```
(casa, 0, derecha)
(cra, 0, diagonal)
(aro, 1, abajo)
```

Notar que las letras de una sopa de letras siempre son en mayúsculas y no tienen tildes, pero la lista de palabras está en minúscula y sí pueden tener tildes.

4. Un club de libros, el último que va quedando, quiere manejar sus libros mediante un programa escrito en Python. Cada libro se representará por una lista de dos elementos:

```
[<título>, <número de copias>]
```

en que `<título>` es un string y `<número de copias>` es un entero. El club organiza los libros en listas por temas:

```
[<tema>, <libroA>, <libroB>, ...]
```

en que `<tema>` es un string y cada `<libroi>` es una lista de dos elementos correspondiente a un libro. Por supuesto, un mismo libro puede pertenecer a más de un tema, por lo que un `<libroi>` puede estar en más de una de estas listas por tema. El club quiere manejar la totalidad de las listas por tema a su vez en otra lista:

```
[[<tema1>, <libroA>, <libroB>, ...], [<tema2>, <libroC>, <libroD>, ...], ...]
```

Se te pide escribir las siguientes funciones en Python para el programa del club:

a) `consultar(libro, listaTotal)`, que devuelve `True` si el club tiene el `libro` (basta que lo encuentre en una de las listas por tema); `False`, en otro caso.

b) `prestar(libro, listaTotal)`, que devuelve `True` si es posible prestar el libro; `False`, en caso contrario. El club puede prestar un libro si el número de copias correspondiente es positivo, en cuyo caso, la función debe reducir en uno el número de copias del libro en cada una de las listas por tema en que aparezca.

c) `agregar(libro, listaTemas, numeroDeCopias, listaTotal)`, que agrega el nuevo `libro` a la `listaTotal` del club; para esto, es necesario agregar el `libro` a cada lista por tema que sea pertinente, con el número de copias correspondiente.