

1) (DEBE RESOLVERSE EN PYTHON)**Construyendo la ciudad...**

La foto del desarrollo edilicio de una cuadra determinada de la ciudad puede “dibujarse” con una estructura de lista de listas de la siguiente manera:

```
[[0,0,1,0,0],
```

```
[0,1,1,1,0],
```

```
[0,1,1,1,1],
```

```
[1,1,1,1,1]]
```

La misma refleja una cuadra corta donde hay 5 construcciones de diversas alturas: 1 piso, 3 pisos, 4 pisos, 3 pisos y 2 pisos respectivamente. El lote 3 de la cuadra tiene el edificio más alto. Los lotes se numeran de izquierda a derecha.

Se solicita generar un programa que nos permita resolver las siguientes funcionalidades:

1. Ingreso manual del desarrollo edilicio de una cuadra determinada validando que la estructura sea correcta. El largo de la cuadra viene dado por el primer nivel ingresado y luego el resto de niveles deben respetar dicho largo. El ingreso debe hacerse con 1 string por nivel. En el ejemplo dado anteriormente los strings ingresados deberían ser: 00100, 01110, 01111, 111111.
2. Ingreso desde un archivo llamado estructura.txt (con las mismas especificaciones del punto 1) el cual debe validarse que sea correctamente formado.
3. Suponiendo que se puede construir el 100% de la altura en todos los edificios, indicar el % de aprovechamiento del espacio aéreo de la cuadra y a su vez Indicar el promedio de los pisos construidos por lote en la cuadra. En el ejemplo anterior sería 80% de aprovechamiento y 2.6 pisos por lote.
4. Emitir el reporte de desarrollo de la cuadra ordenado por cantidad de pisos descendente e indicando los lotes donde hay desarrollos de esa altura. En el ejemplo dado anteriormente el reporte debería indicar:
4 pisos: Lote 3
3 pisos: Lote 2, Lote 4
2 pisos: Lote 5
1 piso: Lote 1
5. Dado que las cuadras tienen mucho margen llegando a sus esquinas se está estudiando una reglamentación para permitir agregar una construcción en cada uno de los extremos de la cuadra. Siendo así y preparándose para dicha situación, el programa debe permitir agregar

un edificio (ingresado como string) en el momento que el usuario desee y agregarlo a la foto de la cuadra ingresado anteriormente. El usuario debe indicar si agregarlo sobre el margen izquierdo o derecho de la foto.

6. Guardado de la estructura con la cual se está trabajando a un archivo llamado estructura_output.txt

Aclaración 1: Los puntos del 1 al 6 deberán resolverse con funciones.

Aclaración 2: Es obligatorio usar try-except por lo menos en un lugar y justificar su uso

Aclaración 3: Debe existir un menú para poder llamar a las opciones a gusto del usuario

2) (DEBE RESOLVERSE EN C)

Dada la siguiente firma de función:

bool ksum(int[] vector, int n, int k); // Evalúa si existe, o no, dos índices distintos del vector cuyos valores sumados sean igual a k

Se pide completar dicha función y utilizarla en un programa para imprimir por pantalla si existe, o no, índices distintos del vector que cumplan dicha suma. El vector debe solicitarse al usuario.

Ejemplos:

1er ejemplo:

VECTOR = [1,2,3,4,5,7,8,9], K = 5

En este caso pueden ser más de dos i,j que cumplan que $VECTOR[i] + VECTOR[j] = K$.

$VECTOR[0] + VECTOR[3] = 1 + 4 = 5$

$VECTOR[1] + VECTOR[2] = 2 + 3 = 5$

Pero basta con la primera ocurrencia para finalizar con la evaluación.

2do ejemplo:

VECTOR = [1,4,6,2,6,8,5], K = 9

En este caso...

$VECTOR[6] + VECTOR[1] = 5 + 4 = 9$