

EXAMEN FINAL

Cátedra Programación II

10/03/2020

1. Presentación del ejercicio

En el marco del distanciamiento preventivo y obligatorio dado ante la pandemia de COVID-19, un instituto de investigación debe iniciar algunas actividades presenciales con ciertas restricciones. Las personas que pertenecen al instituto tienen una ALTA o BAJA prioridad para asistir, dependiendo de sus actividades. Y por otra parte, algunos asistentes al instituto comparten oficina.

Los protocolos relacionados a la pandemia establecen que:

- Sólo podrán asistir al instituto aquellas personas, que dadas sus actividades, tengan prioridad ALTA.
- Cada oficina puede estar ocupada por a lo sumo una persona.

Ante esto, se necesita desarrollar software que permita determinar qué personas podrán asistir al centro de investigación y qué oficinas ocuparán, contemplando que algunas personas pueden ser reubicadas.

Se pide entonces realizar un programa en C y un programa en Python.

- El programa en C tomará de un archivo de entrada, cierta información sobre la estructura del edificio (cantidad de plantas/pisos, máximo y mínimo número de oficinas por planta y cantidad de escritorios por oficina). Con dicha información creará un escenario de cómo está ocupado habitualmente el instituto por las personas que concurren. Esta información la escribirá en un archivo.
- El programa en Python tomará el escenario generado por el programa anterior y determinará qué personas asistirán al centro y dónde estarán ubicadas. Esta información la escribirá en un archivo de salida.

2. Programa en C

El programa debe recibir un archivo de entrada que tendrá tres líneas con la información de la estructura del edificio:

- La cantidad de plantas del edificio. Como mínimo 1 (que sería solo PB) y como máximo 3 (PB, 1ro y 2do piso).
- La cantidad mínima y la cantidad máxima de oficinas que puede haber por planta, separados ambos valores por un espacio. La cantidad mínima no puede ser menor a 2 y la cantidad máxima no puede ser mayor a 10.
- La cantidad de escritorios que tiene cada oficina. Esta cantidad es al menos 1 y a lo sumo 3.

Por ejemplo, el archivo de entrada puede tener las siguientes tres líneas:

```
3
2 4
3
```

La primera línea indica que el edificio tiene 3 plantas. La segunda línea indica que en cada planta hay entre 2 y 4 oficinas. La tercer línea indica que cada oficina tiene 3 escritorios.

El programa deberá hacer lo siguiente:

1. Deberá corroborar que los valores de entrada cumplen con las restricciones descriptas.
2. Deberá representar internamente una matriz que tenga una columna y tantas filas como plantas tenga el edificio. En esta matriz deberá asignar a cada planta una cantidad **aleatoria** de oficinas que deberán respetar el rango establecido por los valores de entrada. Por ejemplo:

0	2	En la planta 0 hay 2 oficinas, en la planta 1 hay 4 oficinas y en la planta 2 hay 2 oficinas.
1	4	
2	2	

3. A partir del punto anterior y del dato de entrada que establece cuántos escritorios hay por oficina, el programa deberá construir una matriz de 4 columnas y de tantas filas como puestos de trabajo/escritorios haya en el edificio. En el ejemplo anterior, como hay 3 escritorios por oficina, habrá 24 puestos de trabajo; por tanto 24 filas. Un ejemplo de la matriz que se debe generar es el siguiente:

	PISO	OFICINA	PERSONA/NADIE	PRIORIDAD/NO APLICA
0	0	0	1	1
1	0	0	2	0
2	0	0	0	-1
3	0	1	0	-1
4	0	1	3	1
5	0	1	4	1
6	1	0	0	-1
7	1	0	0	-1
8	1	0	5	0
9	1	1	0	-1
10	1	1	6	1
11	1	1	0	-1
12	1	2	7	1
13	1	2	0	-1
14	1	2	8	1
15	1	3	0	-1
16	1	3	0	-1
17	1	3	0	-1
18	2	0	9	0
19	2	0	0	-1
20	2	0	0	-1
21	2	1	10	0
22	2	1	11	0
23	2	1	0	-1

Cada fila corresponderá a un escritorio e indicará la ubicación del mismo (planta y oficina) y si está ocupado o está libre; y en el caso de que esté ocupado indicará qué prioridad tiene la persona.

Los valores de la tercer y cuarta columna, correspondientes a si hay un **persona o no** y la **prioridad** de la misma, deben **generarse en forma aleatoria**. Así, para un escritorio determinado (una fila) el programa decide aleatoriamente si está o no ocupado.

- Si está ocupado, asigna en la tercer columna un entero positivo que será un identificador de persona. Luego decide aleatoriamente si la persona tiene prioridad 0 (BAJA) o prioridad 1 (ALTA) y completa con estos valores la cuarta columna.
- Si no está ocupado, asigna a la tercer columna el valor 0 (NADIE) y a la cuarta columna el valor -1 (no aplica).

4. Finalmente, a partir de la matriz anterior, el programa deberá escribir en un archivo el escenario generado. Cada identificador N de la tercer columna, será utilizado para escribir 'PersonaN', si $N > 0$; y 'NADIE', si $N=0$. Por ejemplo, el archivo de salida correspondiente a la matriz anterior tendría las siguientes líneas.

```
0 0 Persona1 1
0 0 Persona2 0
0 0 NADIE -1
0 1 NADIE -1
0 1 Persona3 1
0 1 Persona4 1
1 0 NADIE -1
1 0 NADIE -1
1 0 Persona5 0
1 1 NADIE -1
1 1 Persona6 1
1 1 NADIE -1
1 2 Persona7 1
1 2 NADIE -1
1 2 Persona8 1
1 3 NADIE -1
1 3 NADIE -1
1 3 NADIE -1
2 0 Persona9 0
2 0 NADIE -1
2 0 NADIE -1
2 1 Persona10 0
2 1 Persona11 0
2 1 NADIE -1
```

3. Programa en Python

El programa deberá tomar como entrada, el archivo generado por el programa hecho en C, el cual describirá cuál es la ocupación habitual del instituto y la prioridad que tienen las personas para asistir al mismo. A partir de esto, deberá determinar quiénes asistirán a edificio y qué oficina ocuparán, llevando adelante los pasos y criterios descritos más abajo. Recordar que en una oficina puede haber a lo sumo una persona.

Así, el programa deberá:

1. Descartar a las personas que tienen BAJA prioridad (esto podría resultar en que algunas oficinas queden vacías).
2. Reubicar a las personas de ALTA prioridad, si corresponde, de acuerdo al siguiente criterio.
 - a) Si en una oficina hay sólo una persona. No habrá modificaciones y la persona se mantendrá en esa oficina.
 - b) Si hay oficinas en conflicto, en las que hay más de una persona, el programa deberá reubicar alguna de ellas en una oficina vacía.
 - c) Si hay oficinas en conflicto pero no hay más oficinas vacías. El programa dejará sólo una persona en la oficina, eliminando las restantes (que también tendrán ALTA prioridad). La forma de elegir qué persona se queda, es arbitraria y decidida por el programador.

Tomando como entrada el archivo generado por C, presentado más arriba; una posible salida de este programa sería:

Personas de ALTA prioridad que asistirán al Instituto

Planta - Oficina - Persona/VACIA

```
0 0 Persona1
0 1 Persona4
1 0 Persona3
1 1 Persona6
1 2 Persona8
1 3 Persona7
2 0 VACIA
2 1 VACIA
```

4. Características del Código a Entregar

Se pide que escriba un programa que cumpla con los siguientes requisitos:

- los nombres de funciones y variables deber ser significativos (estar relacionados con su uso);
- no se pueden usar variables globales, definidas fuera de funciones;
- se debe comentar, adecuadamente, el código.