

Fundamentos de Programación

PEC2 - 20191

Fecha límite de entrega: 07/10/2019

Estudiante

Apellidos:

Nombre:

Objetivos

- Saber diseñar expresiones en lenguaje algorítmico y en C
- Aprender a hacer operaciones simples con punteros

Formato y fecha de entrega

La PEC se debe entregar antes del día **07 de octubre de 2019 a las 23:59**. Para la entrega se deberá entregar un fichero en formato **ZIP**, que contenga:

- Este fichero con las respuestas en lenguaje algorítmico del ejercicio 1 y el apartado b del ejercicio 2.
- El workspace de Codelite que contenga el fichero .c pedido en el apartado a del ejercicio 2.

La entrega se hará en el apartado de entregas de EC del aula de teoría.

Enunciado

La compañía UOCBookings tiene el siguiente algoritmo a medio diseñar.

const

NUM_FLOORS: **integer** = 5;

NUM_ROOMS_FLOOR: **integer** = 20;

end const

algorithm UOCBookings

var

hasPool: **boolean**; {the hotel has swimming pool}

hasGym: **boolean**; {the hotel has gym}

closeToSubway: **boolean**; {the hotel is close to a subway station}

distanceFromCityCenter: **real**; {distance in Km.}

priceDouble: **real**;

keyNumber: **integer**;

code: **character**;

discount: **real**;

numGuests: **integer**;

end var

.....

end algorithm

Y nos piden que resolvamos los siguientes ejercicios.

Ejercicio 1: Diseño en lenguaje algorítmico (50%)

Completar el algoritmo de tal manera que:

- a. (10%) Lea por el canal estándar de entrada los siguientes datos correspondientes a un hotel y los almacene en estas variables ya declaradas.

- hasPool
- hasGym
- closeToSubway
- distanceFromCityCenter
- priceDouble
- keyNumber
- code
- discount
- numGuests

- b. [80%] Con los datos leídos y las constantes definidas, calcule las siguientes expresiones y asigne el resultado de cada una de ellas a una variable del tipo adecuado.

b1. [20%] Una expresión que devuelva *true* si el carácter *code* leído asignado a la variable *code* es alfabético (una letra entre la a y la z, ya sea en minúsculas o en mayúsculas) y *false* en caso contrario.

b2. [20%] El hotel tiene 5 plantas y 20 habitaciones en cada planta, valores representados por las constantes *NUM_FLOORS* y *NUM_ROOMS_FLOOR* respectivamente. Las plantas se numeran de la 0 a la 4 y las habitaciones de cada planta de la 1 a la 20. Para identificar el número de una habitación se utiliza un entero de tres dígitos. El dígito más significativo indica la planta (primer dígito) y los otros dos el número de la habitación dentro de la planta. Por ejemplo, la habitación 206 indica que es la habitación número 6 de la planta 2.

En este hotel, cuando les han hecho las llaves, se han equivocado y las han numerado del 1 al 100.

Se pide una expresión que, dado el número de una llave, calcule el número de habitación que le corresponde. Por ejemplo, si la llave es la 56, el número de la habitación es la 216, es decir la habitación 16 de la planta 2 (Tenemos 20 habitaciones de la planta 0, donde añadimos 20 habitaciones más de la planta 1, que en total hacen 40, y finalmente añadimos 16 habitaciones de la planta 2, y obtenemos el 56 de la llave).

b3. [20%] Expresión que devuelve *true* si:

- el hotel tiene piscina o gimnasio,
- y está a menos de 5 Km del centro o está cerca de una estación de metro,
- y el precio de la habitación doble es inferior o igual a 100 €.

y *false* en caso contrario

- b4. [20%] Expresión que calcule el coste del hotel por día para un grupo de *numGuests* sabiendo que le aplicarán el descuento *discount* (número real entre 0 y 50). Se ha de tener en cuenta que sólo hay habitaciones dobles, es decir, si hay una persona sola, ésta tiene que ocupar una habitación doble y pagar el precio de la habitación doble.

- c. [10%] Escriba por el canal estándar de salida el resultado de las 4 expresiones del apartado b.

Nota: Se pueden declarar las variables auxiliares necesarias para guardar el resultado de cada una de las expresiones solicitadas.

Ejercicio 2: Programación en C (50%)

Apartado a [70%] Codificación

1. Implementar en lenguaje C el algoritmo anterior.

Notas:

- Recordad que en C no existe el tipo booleano así que se debe declarar el booleano como un tipo enumerado tal como se indica en los materiales.

Apartado b [30%] Pruebas / Ejecución del algoritmo

Al igual que en la PEC1, se solicita que indiquéis los valores de salida esperados de cada expresión en función de los datos de entrada. Indicad dos juegos de prueba para cada expresión, rellenando las tablas siguientes:

Expresión 1

	Input Data	Output

Expresión 2

	Input Data	Output

Expresión 3

	InputData	output
--	-----------	--------

	hasPool	hasGym	distanceToCenter	closeToSubway	priceDouble	

Expresión 4

Case	Input Data		Output

Criterios de corrección:

En el ejercicio 1:

- Que se siga la notación algorítmica utilizada en la asignatura. Ved el documento *Nomenclator* en la xWiki.
- Que se sigan las instrucciones dadas y el algoritmo responda al problema planteado.
- Que las expresiones diseñadas sean formalmente correctas.

En el ejercicio 2:

- Que el programa compile.
- Que el programa funcione según lo solicitado.
- Que se respeten los criterios de estilo de programación C. Ved la *Guía de estilo de programación en C* que tenéis a la Wiki.
- Se valorará que se declaren en C los tipos enteros más eficientes en relación al uso de memoria según el tipo de dato a representar.
- Que los juegos de prueba dados prueben diferentes casuísticas del programa.