Universitat Oberta de Catalunya uoc.edu

Fundamentos de Programación
PEC3 - 20191

Fecha límite de entrega: 14/10/2019

Estudiante
Apellidos: DELGADO FLORES
Nombre: PABLO JOSE

Objetivos

- Aplicar correctamente la estructura de control alternativa
- Aprender a utilizar el tipo de datos vector para representar estructuras de datos sencillas.

Formato y fecha de entrega

La PEC se debe entregar antes del día **14 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 de la pregunta 1 y apartado b de la pregunta 2
- El workspace de Codelite que contenga los ficheros .c solicitados en la pregunta 2.

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

Enunciado

En la UOCBookings aparecen tres hoteles de la misma categoría y en la misma ciudad. Cada hotel tiene un precio por habitación doble y también ofrece descuentos si se trata de grupos numerosos. En concreto, los tres tienen un descuento para grupos de más de 10 personas.

Nos pide que completemos el siguiente algoritmo que lee, del canal estándar de entrada, los precios y los descuentos de cada hotel y un entero para el número de personas que quieren hacer la reserva y escribe en el canal estándar de salida el nombre del hotel más económico dado el número de personas y el coste total por día.

Nota: Una persona sola no tiene precio especial. El coste es el de una habitación doble.

Nos proporcionan el siguiente algoritmo que hay que completar.

Ejercicio 1: Diseño en lenguaje algorítmico (50%)

Apartado a: [80%]

Partiendo del siguiente esqueleto de algoritmo, que ya tiene las constantes declaradas,

```
const

    NAME_HOTEL1: string  = “SUN”;
    NAME_HOTEL2: string  = “MOON”;
    NAME_HOTEL3: string  = “EARTH”;
    NUM_HOTELS: integer = 3;
    MAX_DISCOUNT: integer = 50;
    HOTEL1: integer = 1;
    HOTEL2: integer = 2;
    HOTEL3: integer = 3;

end const

algorithm UOCBookings

var
    prices: vector[NUM_HOTELS] of integer;
    ...
end var

end algorithm
```

completadlo siguiendo los siguientes pasos:

1. [5%] Declarar un vector de enteros de tres posiciones donde guardar los descuentos respectivos de cada hotel para grupos de más de 10 personas.

2. [5%] Declarar otras constantes y variables que puedan ser necesarias para resolver el ejercicio.
3. [5%] Leer del canal estándar de entrada tres enteros, que corresponden a los precios por habitación doble de cada uno de los hoteles y guardarlos en el vector correspondiente.
4. [5%] Leer del canal estándar de entrada tres enteros, que corresponden a los descuentos de cada hotel y guardarlos en el vector correspondiente.
5. [15%] Comprobar que los datos entrados son correctos, es decir, que cada descuento entrado es mayor o igual a 0% y menor o igual a 50%. No se permiten descuentos superiores al 50%. Si los datos son incorrectos el algoritmo debe mostrar un error y parar la ejecución.
6. [5%] Leer del canal estándar de entrada el número de personas para el que se quiere calcular el precio por día.
7. [60%] Si los datos son correctos, hacer lo siguiente:
    a. [80%] Calcular el coste por día de cada hotel en función del número de personas en el grupo y de los descuentos correspondientes.
    b. [20%] Mostrar por el canal estándar de salida el nombre del hotel más económico y el coste total por día. Si hay dos hoteles que tienen el mismo precio final por día, se coge el primero entrado.

SOLUCION

```
const
      NAME_HOTEL1: string = "SUN";
      NAME_HOTEL2: string = "MOON";
      NAME_HOTEL3: string = "EARTH";
      NUM_HOTELS: integer = 3;
      MAX_DISCOUNT: integer = 50;
      HOTEL1: integer = 1;
      HOTEL2: integer = 2;
      HOTEL3: integer = 3;
end const

algorithm UOCBookings
    var
        prices: vector[NUM_HOTELS] of integer;
        discount: vector[NUM_HOTELS] of integer;
        guests: integer;
        priceWithDiscount: vector[NUM_HOTELS] of real;
        totalPricePerDay: vector[NUM_HOTELS] of real;
    end var
    writeString("Introduce el precio por habitación doble del primer
hotel: ");
    writeString(NAME_HOTEL1);
    prices[1]:=readInteger();
```

```
    writeString(“Introduce el precio por habitación doble del segundo
hotel: ”);
    writeString(NAME_HOTEL2);
    prices[2]:=readInteger();
    writeString(“Introduce el precio por habitación doble del tercer
hotel: ”);
    writeString(NAME_HOTEL3);
    prices[3]:=readInteger();
    writeString(“Introduce el descuento del primer hotel: ”);
    discount[1]:=readInteger();
    writeString(“Introduce el descuento del segundo hotel: ”);
    discount[2]:=readInteger();
    writeString(“Introduce el descuento del tercer hotel: ”);
    discount[3]:=readInteger();
    if ((discount[1] ≥ 0 and discount[1] ≤ MAX_DISCOUNT) and (discount[2]
≥ 0 and discount[2] ≤ MAX_DISCOUNT) and (discount[3] ≥ 0 and discount[3] ≤
MAX_DISCOUNT)) then
        writeString(“Introduce el numero de personas que iran al hotel:“);
        guests:=readInteger();
        if (guests > 10) then
            priceWithDiscount[1]:=(((integerToReal)100 - discount[1]) div
100) * prices[1];
            priceWithDiscount[2]:=(((integerToReal)100 - discount[2]) div
100) * prices[2];
            priceWithDiscount[3]:=(((integerToReal)100 - discount[3]) div
100) * prices[3];
            totalPricePerDay[1]:=((guests div 2) + (guests mod 2)) *
priceWithDiscount[1];
            totalPricePerDay[2]:=((guests div 2) + (guests mod 2)) *
priceWithDiscount[2];
            totalPricePerDay[3]:=((guests div 2) + (guests mod 2)) *
priceWithDiscount[3];
            if ((totalPricePerDay[1] < totalPricePerDay[2] and
totalPricePerDay[1] < totalPricePerDay[3]) or (totalPricePerDay[1] =
totalPricePerDay[2]) or (totalPricePerDay[1] = totalPricePerDay[3])) then
                writeString(“El hotel mas barato es el “);
                writeInteger(HOTEL1);
                writeString(“,”);
                writeString(NAME_HOTEL1);
                writeString(“ con un precio de ”);
                writeReal(totalPricePerDay[1]);
            else
            if ((totalPricePerDay[2] < totalPricePerDay[1] and
totalPricePerDay[2] < totalPricePerDay[3]) or (totalPricePerDay[2] =
totalPricePerDay[3])) then
                writeString(“El hotel mas barato es el “);
                writeInteger(HOTEL2);
```

```
                writeString(“,”);
                writeString(NAME_HOTEL2);
                writeString(“ con un precio de ”);
                writeReal(totalPricePerDay[2]);
            else
                writeString(“El hotel mas barato es el “);
                writeInteger(HOTEL3);
                writeString(“,”);
                writeString(NAME_HOTEL3);
                writeString(“ con un precio de ”);
                writeReal(totalPricePerDay[3]);
            end if
            end if
        end if
    else
        writeString(“================================================”);
        writeString(“ERROR: El descuento introducido es erroneo, debe ser
una cantidad entre 0 y 50, ambos inclusive”);
        writeString(“================================================”);
    end if
end algorithm
```

Apartado b [20%] Razona qué pasaría si en lugar de tres hoteles tuviésemos 30. Con las herramientas que tenemos hasta ahora, ¿sería factible la solución? ¿Sería complicado?

SOLUCIÓN

Creo que sí y no, ya que, aunque cambiar el número de hoteles es tan sencillo como alterar el valor de la constante **NUM_HOTELS**, habría que solicitar los valores para todos esos treinta. Cuando veamos la estructura *for* o *while* todo será mucho más fácil dado que, con alterar el número final, se nos irá pidiendo la información de cada hotel hasta llegar a los 30.

Ejercicio 2: Programación en C (50%)

Apartado a: [80%] Implementar en C el algoritmo del ejercicio 1.

Apartado b: [20%] Como en las anteriores PEC, se solicita que indiquéis juegos de prueba per probar el algoritmo. Concretamente, rellenad la siguiente tabla con cuatro juegos de prueba explicando en cada caso por qué lo habéis escogido (qué parte del código queréis probar.)

SOLUCIÓN

Para llevar a cabo los siguientes juegos de prueba, he numerado cada pasada de la siguiente forma:

1. Introducir un descuento del 60, con la intención de que el programa salga de la ejecución mostrando el mensaje de error.
2. Introducir un valor de 8 para los huéspedes, para que el programa termine correctamente sin mostrar error o algún mensaje más, dado que solo se contempla hacer el cálculo de habitaciones cuando el grupo es mayor de 10.
3. Introducir valores aleatorios en precios y descuentos para comprobar que el cálculo se hace correctamente. Estos valores son:
    a. Precio Hotel 1: 30
    b. Precio Hotel 2: 35
    c. Precio Hotel 3: 25
    d. Descuento Hotel 1: 10
    e. Descuento Hotel 2: 15
    f. Descuento Hotel 3: 20
    g. Personas: 11
4. Introducir valores en precios y descuentos para que el hotel dos y tres tengan los mismos valores para un grupo de 14 personas para comprobar que ante dos hoteles con los mismos datos, salga siempre el introducido primero. Estos valores son:
    a. Precio Hotel 1: 50
    b. Precio Hotel 2: 35
    c. Precio Hotel 3: 35
    d. Descuento Hotel 1: 10
    e. Descuento Hotel 2: 25
    f. Descuento Hotel 3: 25
    g. Personas: 14

INTENTO	Input data	Output data
1	Discount: 60	ERROR
2	Guests: 11	NADA, se ejecuta hasta el final sin errores y sin mostrar más info
3	Datos indicados	El hotel más barato es el 3, HEARTH con un precio de 120.00
4	Datos indicados	El hotel más barato es el 2, MOON con un precio de 183.75

Criterios de corrección:

En el ejercicio 1:

- Que se sigue la notación algorítmica utilizada en la asignatura. Ved el documento Nomenclator en la xWiki.
- Que se siguen las instrucciones dadas y el algoritmo responda al problema planteado.
- Que se aplica correctamente la estructura de control alternativa.
- Que se razona correctamente la respuesta del apartado b de la primera pregunta.

En el ejercicio 2:

- Que el programa se adecua a las indicaciones dadas.
- Que el programa compila y funciona de acuerdo con lo que se pide.
- Que se respetan los criterios de estilo de programación C. Ved la Guía de estilo de programación en C que tenéis en la xWiki.