

Fundamentos de Programación

PEC3 - 20182

Fecha límite de entrega: 18/03/2019

Estudiante

Apellidos: Giménez Gorrís

Nombre: Álvaro

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 **18 de marzo 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 UOCAirways quieren estudiar si la distribución de asientos de un vuelo diario concreto es rentable o no y solicitan nuestra ayuda para analizarlo.

El vuelo en cuestión tiene un total de 180 asientos distribuidos de la siguiente manera:

1. 16 asientos de primera clase (4 asientos por fila) con un *pitch* (distancia entre dos puntos idénticos de dos asientos en fila consecutivas) de 135 cm. y una longitud total para la primera clase de 6 metros. Es decir, que sobra un espacio de 60 cm, entre la primera clase y la clase *business*.
2. 20 asientos de clase *business* (4 asientos por fila) con un *pitch* de 100 cm. y una longitud total de 5,5 metros, es decir que hay un espacio de 50 cm de separación entre la clase *business* y la turista.
3. 144 asientos de clase turista (6 asientos por fila) con un *pitch* de 74 cm.

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

```
NUM_SEATS_FIRST: integer = 16;  
NUM_SEATS_BUSINESS: integer = 20;  
NUM_SEATS_ECONOMY: integer = 144;  
PITCH_FIRST: integer = 135;  
PITCH_BUSINESS: integer = 100;  
PITCH_ECONOMY: integer = 74;  
LENGTH_FIRST: integer = 600;  
LENGTH_BUSINESS: integer = 550;
```

endconst

algorithmsimulator

var

{Declaration of variables}

...

endvar

...

endalgorithm

completadlo siguiendo los siguientes pasos:

1. [10%] Declare un vector de enteros de tres posiciones donde guardar la media de billetes vendidos durante el último año en cada clase. En la primera posición del vector va el número que corresponde a la primera clase, en la segunda posición del vector va la de clase *business* y en la última posición del vector va la de la clase turista.
2. [10%] Declare otras variables y constantes que pueden ser necesarias para resolver el problema.
3. [10%] Lea del canal estándar de entrada tres enteros que corresponden a las tres medias de billetes vendidos durante el último año (primera, *business* y turista). Para cada lectura se debe indicar que información se pide.
4. [10%] Compruebe que los datos entrados son correctos, es decir, que no son ni valores negativos ni superen el número máximo de asientos de cada clase. Si los datos son incorrectos el algoritmo debe dar un error y no continuar.
5. [60%] En caso contrario, es decir, si los datos son correctos, haga lo siguiente:
 - a. [40%] Si el % de la media de billetes vendidos de primera clase es inferior al 25% de los asientos disponibles en esta clase y el % de la media de billetes vendidos en *business* es inferior al 50% de los asientos disponibles en clase *business*, se debe mostrar por el canal estándar de salida un mensaje que diga que se recomienda convertir todo el avión a una única clase, la turista, y calcule y muestre por canal estándar de salida, cuántos asientos nuevos de clase turista se añadirían si se suprimen los asientos de primera clase y de *business*.
 - b. [40%] En caso contrario, si el % de la media de billetes vendidos de primera es menor del 25% pero el % de la clase *business* es superior o igual al 50%, se debe mostrar un mensaje por el canal estándar de salida que recomiende sacar la primera clase y reconvertir el espacio en clase *business*, y calcule y muestre por el mismo canal el número de asientos de clase *business* que se añadirán.
 - c. [20%] En caso contrario saque un mensaje por el canal estándar de salida indicando que es necesario hacer un estudio más detallado.

Apartado b [20%] Explica que se debería modificar para que este algoritmo fuese genérico. Es decir que no haga los cálculos para un avión concreto sino que lo pueda hacer para cualquier vuelo. No hace falta hacer el diseño, solo hay que explicar que se debe hacer.

A continuación, expongo el algoritmo que da respuesta a todo el ejercicio 1, que queda de la siguiente manera:

const

NUM_SEATS_FIRST: **integer** = 16;
NUM_SEATS_BUSINESS: **integer** = 20;
NUM_SEATS_ECONOMY: **integer** = 144;
PITCH_FIRST: **integer** = 135;
PITCH_BUSINESS: **integer** = 100;
PITCH_ECONOMY: **integer** = 74;
LENGTH_FIRST: **integer** = 600;
LENGTH_BUSINESS: **integer** = 550;

SEATS_PER_LINE_BUSINESS : **integer** = 4; {asientos por fila en clase
business}
SEATS_PER_LINE_ECONOMY : **integer** = 6; {asientos por fila en clase turista}

SEATS_WITHOUT_FIRST_AND_BUSINESS : **integer**; {asientos si es todo el avión
turista}
SEATS_WITHOUT_FIRST : **integer**; {asientos si la 1ª clase es
business}

end const

algorithm simulator

var

{1-Declaración de vector de enteros de tres variables}

halfSold : **vector[3]** of **real**; {vector que guarda el valor de las medias}

{2-Declaración de otras variables}

halfSoldFirst : **real**; {media de vendidos en 1ª clase}
halfSoldBusiness : **real**; {media de vendidos en clase business}
halfSoldEconomy : **real**; {media de vendidos en clase turista}

end var

{3-Valores de entrada media}

writeString("Enter the half of tickets sold of the first class: ");
halfSold[1] := readReal();
writeString("Enter the half of tickets sold of the business class: ");
halfSold[2] := readReal();
writeString("Enter the half of tickets sold of the economy class: ");
halfSold[3] := readReal();

halfSoldFirst := (halfSold[1] * 100) **div** NUM_SEATS_FIRST;
halfSoldBusiness := (halfSold[2] * 100) **div** NUM_SEATS_BUSINESS;
halfSoldEconomy := (halfSold[3] * 100) **div** NUM_SEATS_ECONOMY;

```

SEATS_WITHOUT_FIRST_AND_BUSINESS := ((LENGTH_FIRST + LENGTH_BUSINESS)


PITCH_ECONOMY)) * SEATS_PER_LINE_ECONOMY;
SEATS_WITHOUT_FIRST := (LENGTH_FIRST 

PITCH_BUSINESS) *
SEATS_PER_LINE_BUSINESS;


```

{4-Comprobación de datos}

```

if (halfSold[1] ≥ 0) and (halfSold[1] ≤ NUM_SEATS_FIRST) and (halfSold[2] ≥ 0) and
(halfSold[2] ≤ NUM_SEATS_BUSINESS) and halfSold[3] ≥ 0) and (halfSold[3] ≤
NUM_SEATS_ECONOMY) then

    if (halfSoldFirst < 25) then
        if (halfSoldBusiness < 50) then {5a-Si first es menor que 25 y business
        menor que 50}
            writeString("It is recommended that you convert all the plane to
            economy class");
            writeString("New seats if you convert all the plane to
            economy class: ");
            readInteger(seatsWithoutFirstAndBusiness);

        else {5b-Si first es menor que 25 y business mayor que 50}
            writeString("It is recommended that you convert the first class to
            business class");
            writeString("New seats if you convert the first class to business
            class: ");
            readInteger(seatsWithoutFist);

        end if

    else {5c-Si first es mayor que 25}
        writeString("You need a more detailed study.");

    end if

else
    writeString("Data error");
    if ((halfSold[1] < 0) or (halfSold[1] > NUM_SEATS_FIRST)) then
        writeString("For the first class, you need a number between 0 and 16");
    if ((halfSold[2] < 0) or (halfSold[2] > NUM_SEATS_BUSINESS)) then
        writeString("For the business class, you need a number between 0 and
        20");
    if ((halfSold[3] > 0) or (halfSold[3] > NUM_SEATS_ECONOMY)) then
        writeString("For the economy class, you need a number between 0 and
        144");

    end if

end algorithm

```

Apartado B

Para que el algoritmo fuera genérico se necesitaría convertir todas las constantes en variables y leer por el canal de entrada el valor de cada variable de la siguiente forma:

1º - El algoritmo debería preguntar si el avión que se quiere analizar dispone de las tres clases con las que cuenta nuestro avión actual, mostrando por el canal estándar de salida las respectivas preguntas de la siguiente forma:

- ¿Cuenta el avión xxxxx con primera clase?
- ¿Cuenta el avión xxxxx con clase business?
- ¿Cuenta el avión xxxxx con clase turista?

A continuación de cada entrada, debería leer por el canal estándar de entrada si es cierto o falso.

2º- El algoritmo debería de preguntar cuantos asientos hay en cada clase. Para ello debería de mostrar por el canal estándar de salida de la siguiente forma:

- ¿Cuántos asientos hay en primera clase?
- ¿Cuántos asientos hay en clase business?
- ¿Cuántos asientos hay en clase turista?

A continuación de cada pregunta, debería de leer por el canal estándar de entrada el valor de cada variable.

3º- El algoritmo debería de preguntar la distancia que hay entre asientos en cada clase. Para ello debería de mostrar por el canal estándar de salida de la siguiente forma:

- ¿Cuánta distancia hay entre asientos en primera clase?
- ¿Cuánta distancia hay entre asientos en clase business?
- ¿Cuánta distancia hay entre asientos en clase turista?

A continuación de cada pregunta, debería de leer por el canal estándar de entrada el valor de cada variable.

4º- El algoritmo debería de preguntar el espacio que ocupa cada clase. Para ello debería de mostrar por el canal estándar de salida de la siguiente forma:

- ¿Cuánto espacio ocupa la primera clase?
- ¿Cuánto espacio ocupa la clase business?
- ¿Cuánto espacio ocupa la clase turista?

A continuación de cada pregunta, debería de leer por el canal estándar de entrada el valor de cada variable.

5º- El algoritmo debería de preguntar cuantos asientos hay por fila en cada clase. Para ello debería de mostrar por el canal estándar de salida de la siguiente forma:

- ¿Cuántos asientos hay por fila en primera clase?
- ¿Cuántos asientos hay por fila en clase business?
- ¿Cuántos asientos hay por fila en clase turista?

A continuación de cada pregunta, debería de leer por el canal estándar de entrada el valor de cada variable.

Con la introducción de los valores de las variables descritas anteriormente, el algoritmo debería de funcionar en cualquier avión, ya que se adaptaría a todas las condiciones posibles y sería totalmente genérico.

Ejercicio 2: Programación en C (50%)

Implementar en C el algoritmo del ejercicio 1.

Apartado a: [70%] Implementar en C el algoritmo del ejercicio 1.

Apartado b: [30%] Como en las anteriores PACs, se solicita que indiquéis juegos de prueba per probar el algoritmo. Concretamente, rellenad la siguiente tabla

| Input data | | | Output data |
|------------|----------|----------|---|
| average1 | average2 | average3 | |
| 3 | 8 | 120 | New seats of economy class if you convert all the plane: 90. |
| 3 | 10 | 120 | New seats of business class if you convert the first class: 24. |
| 5 | 10 | 120 | You need a more detailed study. |
| 17 | 10 | 120 | Data error. |

Donde *average1*, *average2* y *average3* son los valores que hay que entrar en el programa

Prueba 1

```
Enter the half tickets sold at the first class:
3
Enter the half tickets sold at the business class:
8
Enter the half tickets sold at the economy class:
120
It is recommended that you convert all the plane to economy class.
New seats of economy class if you convert all the plane:
90
Press ENTER to continue...
█
```

Prueba 2

```
Enter the half tickets sold at the first class:
3
Enter the half tickets sold at the business class:
10
Enter the half tickets sold at the economy class:
120
It is recommended that you convert the first class to business class.
New seats of business class if you convert the first class:
24
Press ENTER to continue...
█
```

Prueba 3

```
Enter the half tickets sold at the first class:
5
Enter the half tickets sold at the business class:
10
Enter the half tickets sold at the economy class:
120
You need a more detailed study.
Press ENTER to continue...
█
```

Prueba 4

```
Enter the half tickets sold at the first class:
17
Enter the half tickets sold at the business class:
10
Enter the half tickets sold at the economy class:
120
Data error.
For the first class, you need a number between 0 and 16.
Press ENTER to continue...
█
```

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 se aplique correctamente la estructura de control alternativa.
- Que se razone correctamente la respuesta del apartado b de la primera pregunta

En el ejercicio 2:

- Que el programa se adecue a las indicaciones dadas.
- Que el programa compile y funcione de acuerdo con lo que se pide.
- Que se respeten los criterios de estilo de programación C. Ved la Guía de estilo de programación en C que tenéis en la xWiki.