

Fundamentos de Programación

PEC3 - 20181

Fecha límite de entrega: 19/03/2018

Estudiante

Apellidos: de Lima B. Campos

Nombre: Lidiane A.

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 **15 de octubre de 2018 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.
- El fichero de código respetando el nombre indicado en el enunciado.

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

Enunciado

Siguiendo con la ayuda que proporcionamos a la compañía UOCRailways, nos proponen que hagamos un programa que les permita estudiar el tipo de tren necesario para poder cargar una determinada mercancía y el precio que tendrán que cobrar por subir la mercancía al tren.

De momento quieren saber:

- Si un modelo de tren tiene capacidad para hacer frente a un volumen de

mercancía dado. Lo quieren estudiar para diferentes vagones y diferentes longitudes máximas de tren. Por eso quieren poder hacer simulaciones.

- Quieren estudiar el precio que deben cobrar para cargar la mercancía en el tren.
- Quieren establecer tres grupos de precios, en función del volumen de mercancía y de si la mercancía es frágil y / o peligrosa.

Nos piden que diseñemos un programa que les permita hacer simulaciones de tal manera que, entrando la longitud máxima del tren, el volumen de la mercancía, el volumen de cada vagón, la longitud de la locomotora y de los vagones, mire si el tren tiene capacidad para cargar la mercancía. En caso afirmativo, debe calcular el precio de cargar la mercancía en el tren (el precio del desplazamiento va aparte y no se calcula, ya que dependerá del destino).

Concretamente piden lo siguiente:

Ejercicio 1: Diseño en lenguaje algorítmico (50%)

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

```
const
  NUM: integer = 5;
  MAX_WAGON_CAPACITY: real = 0.85;
  MAX_WAGON_CAPACITY_ANIMALS: real = 0.5;
  LIMIT1: integer = 500;
  LIMIT2: integer = 2500
  FRAGILE: real = 1.10;
  DANGEROUS: real = 1.15;
  FIRST_PRICE: real = 0.50;
  SECOND_PRICE: real = 0.45;
  THIRD_PRICE: real = 0.40;
end const

type
{ Definition of new types}
...
end type

algorithm simulator
```

```
var
    {Declaration of variables}
    ...
end var
```

end algorithm

completarlo haciendo los siguientes pasos:

- a) (5%) Declaración de variables para guardar el identificador de la mercancía (un entero), volumen de la mercancía en metros cúbicos (un real), el tipo de mercancía (que puede ser FOOD, CHEMICAL, ANIMALS, VEHICLES, ELECTRONICS, CONSTRUCTION, OTHERS), información sobre si es frágil (un booleano) e información sobre si es peligrosa (un booleano).
- b) (10%) Declaración de una variable, llamada **train**, de tipo vector de reales de cinco posiciones.
- c) (20%) Lectura del canal estándar de entrada de los datos de una mercancía. Se pueden utilizar las siguientes funciones de lectura:
 - i. readGoodType(), para leer el tipo de mercancía.
 - ii. readBoolean(), para leer si la mercancía es frágil y/o peligrosa.
- d) Lectura por el canal estándar de entrada de cinco números reales que hay que guardar en el vector **train**. Los reales representan la longitud máxima del tren, la longitud de la locomotora, la longitud de cada vagón, el espacio entre vagones (los cuatro valores en metros) y el volumen de un vagón (en metros cúbicos). Se asume que se introducen ordenados por el canal estándar de entrada. Es decir, si los número leídos son 150, 10.1, 14.5, 2 y 108.5 significa que
 - o La longitud máxima del tren es de 150 metros.
 - o La longitud de la locomotora es de 10.1 metros.
 - o La longitud de cada vagón es de 14.5 metros.
 - o El espacio entre vagones es de 2 metros.
 - o El volumen de un vagón es de 108.5 metros cúbicos.
- e) (5%) El algoritmo debe calcular el número de vagones según la configuración de tren que se ha introducido en el apartado d. Para saber el número de vagones que tiene el tren podéis adaptar la expresión b1 de la PEC2.

- f) (5%) El algoritmo debe calcular si la mercancía cabe en el tren. Con carácter general, los vagones se llenan, como máximo, al 85% de su capacidad. En el caso particular de que la mercancía sea de tipo ANIMALES, los vagones se llenan, como máximo, al 50% de su capacidad.
- g) (30%) El algoritmo debe calcular el precio de cargar la mercancía en los vagones con los siguientes criterios. Si la mercancía ocupa:
- **Tarifa Primera:** menos de 500m³, el precio base es 0.50 € / m³ de mercancía.
 - **Tarifa Segunda:** entre 500m³ Y 2500 m³, el precio base es 0.45 € / m³ de mercancía.
 - **Tarifa Tercera:** más de 2500 m³, el precio base es 0.40 € / m³ de mercancía.
- Además, si la mercancía es frágil el precio de cargar la mercancía en el tren tiene un recargo del 10% sobre el precio base y si la mercancía es peligrosa tiene un recargo adicional de un 15% también sobre el precio base.
- h) (5%) Salida de datos. Si la mercancía no hacia el tren hay que mostrar un mensaje por el canal estándar de salida diciendo que la mercancía no cabe en el tren. En caso contrario, si la mercancía sí cabe el tren, hay que mostrar un mensaje por el canal estándar de salida que muestre el número de vagones que tiene el tren, el identificador de la mercancía y el precio que calculado para cargar la mercancía en el tren.

Ejercicio 2: Programación en C (50%)

Implementar en C el algoritmo del ejercicio 1. Como en las anteriores PECs, se pide que probéis el algoritmo para diferentes valores de entrada y hagáis capturas de pantalla donde se vea la salida de. programa para los diferentes valores de entrada. Las capturas de pantalla se deben adjuntar en el documento donde está la respuesta del ejercicio 1,

En concreto, se pide que hagáis las siguientes seis pruebas:

- Un volumen de mercancía que no cabe en un tren (**prueba 1**).
- Para un volumen de mercancía que sí cabe en el tren, mostrad los precios para cargar la mercancía en el tren (apartado g).
 - Se deben hacer simulaciones con diferentes valores del vector *train* y *volumeGood* que den lugar a precios para cargar la mercancía en el tren en las tres tarifas (Primera, Segunda y Tercera) (**pruebas 2, 3 y 4**).
 - Se debe hacer una simulación donde se aplique el recargo a mercancías frágiles (**prueba 5**).

- Se debe hacer una simulación donde se aplique el recargo a mercancías peligrosas (**prueba 6**).

Nota: Recordad que en C las posiciones del vector serán train[0], train[1], train[2], train[3] y train[4].

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.

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..

Respuestas

const

```
    NUM: integer = 5;  
    MAX_WAGON_CAPACITY: real = 0.85;  
    MAX_WAGON_CAPACITY_ANIMALS: real = 0.5;  
    LIMIT1: integer = 500;  
    LIMIT2: integer = 2500  
    FRAGILE: real = 1.10;  
    DANGEROUS: real = 1.15;  
    FIRST_PRICE: real = 0.50;  
    SECOND_PRICE: real = 0.45;  
    THIRD_PRICE: real = 0.40;  
end const
```

type

```
    goodType = {FOOD, CHEMICAL, ANIMALS, VEHICLES, ELECTRONICS,  
    CONSTRUCTION, OTHERS};  
end type
```

algorithm simulator

var

```
    goodId: integer;  
    nWagons: integer;  
    goodVolume: real;  
    isFragile: boolean;  
    isDangerous: boolean;  
    totalVolAvailable: real;  
    trainData: vector[NUM] of real;  
    transportPrice: real;
```

end var

```
writeString("Enter the good data");  
writeString("Enter the good id: ");  
goodId := readInteger();
```

```
writeString("Enter the good volume (m³): ");  
goodVolume:= readReal();
```

```
writeString("Enter the good type ('FOOD, CHEMICAL, ANIMALS, VEHICLES, ELECTRONICS,  
CONSTRUCTION, OTHERS)");  
goodType:= readGoodType();
```

```
writeString("Is the good fragile? ");
isFragile:= readBoolean();
```

```
writeString("Is the good dangerous? ");
isDangerous:= readBoolean();
```

```
writeString("Enter the train informations la longitud máxima, de la locomotora, de cada
vagón, el espacio entre vagones y volumen de un vagón: ");
trainData[1]:=readReal();
trainData[2]:=readReal();
trainData[3]:=readReal();
trainData[4]:=readReal();
trainData[5] :=readReal();
```

```
nWagons:= realToInteger( ( trainData[1] - trainData[2]) / ( trainData[3] +
trainData[4]));
```

```
if goodType=ANIMALS then
    totalVolAvailable:=integerToReal((nWagons*trainData[5])*MAX_WAGON_CAP
ACITY_ANIMALS;
else
    totalVolAvailable:=integerToReal((nWagons*trainData[5])*MAX_WAGON_CAPA
CITY;
end if
```

```
if goodVolume ≤ totalVolAvailable then
    if goodVolume ≤ LIMIT1 then
        transportPrice := goodVolume * FIRST_PRICE;
    else
        if goodVolume > LIMIT1 and goodVolume ≤ LIMIT2 then
            transportPrice := goodVolume * SECOND_PRICE;
        else
            if goodVolume > LIMIT2 then
                transportPrice := goodVolume * THIRD_PRICE;
            end if
        end if
    end if
```

```
if isFragile then
    transportPrice:= transportPrice * FRAGILE;
```

```
if isDangerous then
    transportPrice:= transportPrice * DANGEROUS;
```

```
writeString("Wagons number : ");
writeInteger("%d\n", nWagons);
writeString("Good identifier: ");
writeInteger("%d\n", goodId);
writeString("Transport price: ");
writeReal("%f\n", transportPrice);
```

```
else
    writeString("The product volume is over the maximum allowed");
end if
```

Un volumen de mercancía que no cabe en un tren (**prueba 1**).

```
./PEC3
File Edit Tabs Help
Enter the good id:
12345
Enter the good volume (m³):
9000
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
3
Is the good fragile? 1-YES 0-NO
0
Is the good dangerous?1-YES 0-NO
0
Enter the train informations
Maximum length's train:500

Maximum length's locomotive: 100

Maximum length's each wagon: 20

The space between wagons: 2

The volume of a wagon: 20
The product volume is over the maximum allowed
Press ENTER to continue...
```

Tarifas (Primera) (**prueba 2**)

```
./PEC3
File Edit Tabs Help
Enter the good volume (m³):
499.99
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
1
Is the good fragile? 1-YES 0-NO
0
Is the good dangerous?1-YES 0-NO
0
Enter the train informations
Maximum length's train:1000

Maximum length's locomotive: 100

Maximum length's each wagon: 20

The space between wagons: 2

The volume of a wagon: 20
Wagons number : 40
Good identifier: 123456
Transport price: 249.994995
Press ENTER to continue...
```

Tarifas (Segunda) (**prueba 3**)

```
./PEC3
File Edit Tabs Help
Enter the good volume (m³):
1500.69
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
2
Is the good fragile? 1-YES 0-NO
0
Is the good dangerous?1-YES 0-NO
0
Enter the train informations
Maximum length's train:2000

Maximum length's locomotive: 100

Maximum length's each wagon: 20.60

The space between wagons: 2

The volume of a wagon: 120.69
Wagons number : 84
Good identifier: 7584
Transport price: 675.310486
Press ENTER to continue...
```


Tarifas (Tercera) (prueba 4)

```
./PEC3
File Edit Tabs Help
Enter the good volume (m³):
2600
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
4
Is the good fragile? 1-YES 0-NO
0
Is the good dangerous?1-YES 0-NO
0
Enter the train informations
Maximum length's train:3000

Maximum length's locomotive: 200

Maximum length's each wagon: 30

The space between wagons: 2

The volume of a wagon: 250.69
Wagons number : 87
Good identifier: 9636
Transport price: 1040.000000
Press ENTER to continue...
```

Se debe hacer una simulación donde se aplique el recargo a mercancías frágiles (prueba 5).

```
./PEC3
File Edit Tabs Help
Enter the good volume (m³):
500
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
3
Is the good fragile? 1-YES 0-NO
1
Is the good dangerous?1-YES 0-NO
0
Enter the train informations
Maximum length's train:200

Maximum length's locomotive: 12.50

Maximum length's each wagon: 16.80

The space between wagons: 2

The volume of a wagon: 110.50
Wagons number : 9
Good identifier: 523632
Transport price: 275.000000
Press ENTER to continue...
```

Se debe hacer una simulación donde se aplique el recargo a mercancías peligrosas (prueba 6).

```
./PEC3
File Edit Tabs Help
Enter the good volume (m³):
500
Enter the good type (1-FOOD, 2-CHEMICAL, 3-ANIMALS, 4-VEHICLES, 5-ELECTRONICS, 6-CONSTRUCTION, 7-OTHERS)
1
Is the good fragile? 1-YES 0-NO
0
Is the good dangerous?1-YES 0-NO
1
Enter the train informations
Maximum length's train:180

Maximum length's locomotive: 10.1

Maximum length's each wagon: 16.50

The space between wagons: 2

The volume of a wagon: 110.50
Wagons number : 9
Good identifier: 897845
Transport price: 287.500000
Press ENTER to continue...
```