

Fonaments de Programació

PAC4 - 20182

Data límit de Iliurament: 25/03/2019

Estudiant

Cognoms: GALEANO SANCHO

Nom: DANIEL

Objectius

- Saber aplicar correctament l'estructura de control iterativa.
- Aprendre a utilitzar el tipus de dades vector per representar estructures de dades senzilles.
- Saber utilitzar les funcions de conversió de tipus quan calgui.

Format i data de Iliurament

Cal Iliurar la PAC abans del dia **25 de març de 2019 a les 23:59**.

Per al Iliurament caldrà que entregueu un fitxer en format ZIP, que contingui:

- Aquest document amb la resposta de l'exercici 1 i l'apartat b de l'exercici 2
- Un workspace de Codelite que contingui els fitxers .c demanats a l'exercici 2a

Cal fer el Iliurament a l'apartat de Iliuraments d'AC de l'aula de teoria.

Enunciat

Seguint amb l'ajuda que proporcionem a la companyia UOCAirways, ens ha demanat que continuem la nostra col·laboració per a crear un programa que els ajudi a gestionar els avions. En aquest cas, tots els avions que es tractin seran del mateix fabricant (*manufacturer*) i del mateix model (*model*), però tindran configuracions diferents. En els següents exercicis, treballarem amb vectors, les

estructures iteratives i l'entrada i sortida interactiva per gestionar les dades dels avions.

Donats els següents tipus de dades i variables definits per gestionar les dades de diversos avions:

Llenguatge algorítmic

type

 tUtility = {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL, OTHERS}

end type

const

 MAX_PLANES: **integer** = 10;

end const

var

 idPlane: **vector[MAX_PLANES] of integer**; {planes id's vector}

 manufacturer: **string**; {manufacturer – unique. The same for all planes }

 model: **string**; {model – unique. The same for all planes }

 year: **vector[MAX_PLANES] of integer**;

 seats: **vector[MAX_PLANES] of integer**;

 utility: **vector[MAX_PLANES] of tUtility**;

 isActive: **vector[MAX_PLANES] of boolean**;

end var

Exercici 1: [50%]

Apartat a [80%]

Implementeu, en llenguatge algorísmic, un algoritme que permeti introduir de manera interactiva pel canal estàndard d'entrada les dades de diversos avions (fins al màxim permès) en les variables que teniu declarades a l'enunciat.

El funcionament de l'algorisme és el següent:

- Es demana a l'usuari que introdueixi el nombre d'avions que es volen emmagatzemar als vectors.

- En cas que el nombre d'avions a introduir sigui vàlid (superior a zero i inferior o igual al nombre màxim permès), es continua endavant.
- En cas que el nombre d'avions NO sigui vàlid, es mostra un missatge d'error pel canal estàndar de sortida i es torna a sol·licitar el nombre d'avions que es volen introduir.
- El procés anterior es repeteix indefinidament fins que l'usuari introduceixi un nombre d'avions vàlid, és a dir, un nombre més gran que 0 i menor o igual que MAX_PLANES.
- Un cop introduït un nombre d'avions vàlid, l'algorisme demana pel canal estàndard d'entrada que s'introduceixin les següents dades:
 - El nom del fabricant i del model dels avions, comú a tots els avions que es tracten en aquest exercici.
 - Dades de cada un dels avions, que es guarden dins el vector corresponent. La informació que cal sol·licitar és la següent:
 - Id de l'avió, de tipus enter
 - Nombre de seients, de tipus enter. Considereu que només hi ha una classe.
 - Any de fabricació, de tipus enter
 - Ús de l'avió, de tipus tUtility
 - Si està o no actiu, de tipus booleà
- Un cop introduïdes les dades dels avions, l'algorisme calcula la mitjana de seients dels avions introduïts.
- Per acabar, l'algorisme ha de mostrar pel canal estàndard de sortida les dades de cada avió, amb un text que expliqui què és cada valor mostrat i indicant si la capacitat de cada avió supera o no la mitjana de seients calculada en el punt anterior. Les dades comuns a tots els avions s'han de mostrar una sola vegada abans de mostrar el detall de cada avió.

Observacions:

- Cal utilitzar estructures iteratives, en les seves diverses variants, segons quina sigui la més útil en cada cas.
- Les dades introduïdes dels avions es poden considerar correctes. Per exemple, quan s'introduceixi *idPlane* (l'identificador d'un avió) es dóna per suposat que l'usuari introduirà un valor enter, etc.
- Cal declarar les variables auxiliars necessàries per dissenyar l'algorisme.
- Recordeu que, com en les PAC anteriors, en el cas del llenguatge algorísmic podeu utilitzar la funció *readUtility()* per llegir el tipus de *tUtility*.

```

type
    tUtility = {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL,
    OTHERS}
end type

const
    MAX_PLANES: integer = 10;
end const

algorithm simulator

var
    idPlane: vector[MAX_PLANES] of integer;
    manufacturer: string;
    model: string;
    year: vector[MAX_PLANES] of integer;
    seats: vector[MAX_PLANES] of integer;
    utility: vector[MAX_PLANES] of tUtility;
    isActive: vector[MAX_PLANES] of boolean;
    numPlanes: integer;
    i := integer;
    avgSeats := real;
end var

numPlanes := 0;

while numPlanes <= 0 and numPlanes > MAX_PLANES do
    writeString("How many planes do you want to enter?: >> ");
    numPlanes := readInteger();
    if numPlanes <= 0 and numPlanes > MAX_PLANES then
        writeString("Incorrect value. Try again.");
        end if
end while

writeString("Enter the name of the manufacturer of all planes: >> ");
manufacturer := readString();
writeString("Enter the name of the model of all planes: >> ");
model := readString();

for i:=1 to numPlanes do
    writeString("Plane number");
    writeln(i);
    writeString("Enter id of the plane: >> ");
    idPlane [i] := readInteger();
    writeString("Enter number of seats of the plane: >> ");
    seats [i] := readInteger();
    writeString("Enter year of manufacturing of the plane: >> ");
    year [i] := readInteger();
    writeString("Enter the utility of the plane (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> ");
    utility [i] := readUtility();
    writeString("Enter if the plane is active: >> ");
    isActive [i] := readBoolean();

```

```

avgSeats := avgSeats + integerToReal(seats[i]);
end for

avgSeats := avgSeats / integerToReal(numPlanes);

writeString("Manufacturer of the planes:");
writeString(manufacturer);
writeString("Model of the planes:");
writeString(model);

for i:=1 to numPlanes do
  writeString("ID:");
  writeln(idPlane[i]);
  writeString("Seats:");
  writeln(seats[i]);
  writeString("Manufacturing year:");
  writeln(year[i]);
  writeString("Utility: (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3
for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS)");
  writeUtility/utility[i];
  writeString("Active:");
  writeBoolean(isActive[i]);
  if seats[i] > avgSeats then
    writeString("Number of seats are greater than the average.");
  else
    writeString("Number of seats are not greater than the average.");
  end if
end for

```

Apartat b [20%]

Explicar què caldria fer si el càlcul de les mitjanes dels seients es volgués fer separats per categoria: primera, *business*, turista. No cal dissenyar-ho.

Necessitarem tres variables de tipus real per emmagatzemar les mitjanes de cada categoria. Pel que fa al nombre de seients, podríem convertir el vector seats[i] a una matriu seats[i][j], on j serà la categoria.

Per exemple, si l'avió 2 té 5 seients de primera, 10 de *business* i 100 de turista, ho emmagatzemarem de la següent manera:

```

seats[2][1] = 5;
seats[2][2] = 10;
seats[2][3] = 100;

```

Com que només tenim tres categories, no fa falta afegir cap bucle per la j, només cal que afegim un writeString per preguntar el nombre de seients de cada categoria i mostrar-ho per pantalla de la mateixa manera.

Exercici 2: [50%]

Apartat a [80%] Codificació

Codificar en C l'algorisme de l'exercici 1.

- En la programació els reals s'han d'escriure amb dos decimals.
- La longitud màxima del nom del fabricant ha de ser de 25 caràcters.
- La longitud màxima del model dels avions ha de ser de 15 caràcters.

```
/*
** File: main.c
** Author: Daniel Galeano Sancho
** Date: 19-03-2019
** Description: PAC4
*/

/* System header file */
#include <stdio.h>
#include <string.h>

/* User defined types */
typedef enum {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL,
OTHERS} tUtility;
typedef enum {FALSE, TRUE} bool;

/* Main function */
int main(int argc, char **argv)
{
    /* Constants */
    const int MAX_PLANES = 10;
    const int MAX_LENGTH = 25+1;

    /* Variables */
    int idPlane[MAX_PLANES];
    char manufacturer[MAX_LENGTH];
    char model[MAX_LENGTH-10];
    int year[MAX_PLANES];
    int seats[MAX_PLANES];
    tUtility utility[MAX_PLANES];
    bool isActive[MAX_PLANES];
    int numPlanes;
    int i;
    float avgSeats;

    /* Inicialization of variables */
    numPlanes = 0;
    avgSeats = 0;

    /* Loop to keep values between conditions */
    do{
        printf("How many planes do you want to enter?: >> ");
        scanf("%d",&numPlanes);
        if((numPlanes <= 0) || (numPlanes > MAX_PLANES)){
            printf("Incorrect value. Value must be between 1 and %d.\nTry again.\n",MAX_PLANES);
        }
    }while((numPlanes <= 0) || (numPlanes > MAX_PLANES));
```

```

/* Reading variables */
printf("Enter the name of the manufacturer of all planes: >> ");
scanf("%s",manufacturer);
printf("Enter the name of the model of all planes: >> ");
scanf("%s",model);

for (i=0;i<numPlanes;i++) {
    printf("\nPlane number %d\n",i+1);
    printf("Enter id of the plane: >> ");
    scanf("%d",&idPlane[i]);
    printf("Enter number of seats of the plane: >> ");
    scanf("%d",&seats[i]);
    printf("Enter year of manufacturing of the plane: >> ");
    scanf("%d",&year[i]);
    printf("Enter the utility of the plane (0 for COMMERCIAL, 1 for
PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for
OTHERS): >> ");
    scanf("%u",&utility[i]);
    printf("Enter if the plane is active (0 for FALSE, 1 for TRUE):
>> ");
    scanf("%u",&isActive[i]);

    avgSeats += (float)seats[i];
}

avgSeats /= (float)numPlanes;

/* Showing variables */
printf("\nManufacturer of the planes: %s\n",manufacturer);
printf("Model of the planes: %s\n",model);

for (i=0;i<numPlanes;i++) {
    printf("\nID: %d\n",idPlane[i]);
    printf("Seats: %d\n",seats[i]);
    printf("Manufacturing year: %d\n",year[i]);
    printf("Utility (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS):
%u\n",utility[i]);
    printf("Active (0 for FALSE, 1 for TRUE): %u\n",isActive[i]);
    if (seats[i] > avgSeats){
        printf("Number of seats are greater than the
average.\n");
    }
    else{
        printf("Number of seats are not greater than the
average.\n");
    }
}

return 0;
}

```

Apartat b [20%] Proves / Execució de l'algorisme

Com en les pacs anteriors, es demana que doneu tres jocs de prova per l'exercici.

Per cada joc, indiqueu els valors de les dades d'entrada i les dades de sortida esperades.

Input Data	Output Data
Number of Planes	
-10	Incorrect value. Value must be between 1 and 10. Try again.
15	Incorrect value. Value must be between 1 and 10. Try again.
2	Enter the name of the manufacturer of all planes: >>

A continuació, es mostra la sortida d'una execució del programa amb les dades anteriors introduïdes. En negre, la sortida automàtica i, en vermell, la entrada de l'usuari.

```
How many planes do you want to enter?: >> -10
Incorrect value. Value must be between 1 and 10.
Try again.
How many planes do you want to enter?: >> 15
Incorrect value. Value must be between 1 and 10.
Try again.
How many planes do you want to enter?: >> 2
Enter the name of the manufacturer of all planes: >> Boing
Enter the name of the model of all planes: >> 747

Plane number 1
Enter id of the plane: >> 1
Enter number of seats of the plane: >> 200
Enter year of manufacturing of the plane: >> 1997
Enter the utility of the plane (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> 0
Enter if the plane is active (0 for FALSE, 1 for TRUE): >> 0

Plane number 2
Enter id of the plane: >> 2
Enter number of seats of the plane: >> 300
Enter year of manufacturing of the plane: >> 2007
Enter the utility of the plane (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> 2
Enter if the plane is active (0 for FALSE, 1 for TRUE): >> 1

Manufacturer of the planes: Boing
Model of the planes: 747

ID: 1
Seats: 200
Manufacturing year: 1997
Utility (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): 0
Active (0 for FALSE, 1 for TRUE): 0
Number of seats are not greater than the average.
```

```
ID: 2
Seats: 300
Manufacturing year: 2007
Utility (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for
MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): 2
Active (0 for FALSE, 1 for TRUE): 1
Number of seats are greater than the average.
Press ENTER to continue...
```

Criteris de correcció:

A l'exercici 1:

- Que se segueixi la notació algorísmica utilitzada a l'assignatura. Vegeu document Nomenclator a la xWiki de contingut.
- Que se segueixen les instruccions donades i l'algorisme respongui al problema plantejat.
- Que s'utilitzi correctament l'estructura iterativa, el tipus de dades vector i, si s'escau, les funcions de conversió de tipus.
- Que es raoni correctament la resposta de l'apartat b de la primera pregunta.

A l'exercici 2:

- Que el programa s'adeqüi a les indicacions donades.
- Que el programa compila i funciona d'acord amb el que es demana.
- Que es respectin els criteris d'estil de programació C. Vegeu la Guia d'estil de programació en C que teniu a la Wiki de contingut.
- Que es declarin els tipus adequats segons el tipus de dades que representa.