

Fonaments de Programació

PAC5 - 20182

Data límit de lliurament: 01/04/2019

Estudiant

Cognoms: **GALEANO SANCHO**

Nom: **DANIEL**

Objectius

- Saber definir correctament els tipus de dades estructurats.
- Aprendre a utilitzar l'entrada i sortida per modificar camps concrets d'estructures de dades senzilles.
- Saber utilitzar les funcions de conversió de tipus quan calgui.

Format i data de lliurament

Cal lliurar la PAC abans del dia **1 d'abril de 2019 a les 23:59**.

Per al lliurament 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 lliurament a l'apartat de lliuraments d'AC de l'aula de teoria.

Enunciat

Seguint amb l'ajuda que proporcionem a la companyia UOCAirways, ens han demanat la nostra col·laboració per a crear un programa que els ajudi a gestionar les bases de dades dels seus avions. En els següents exercicis, treballarem amb tipus de dades estructurats, conjuntament amb l'entrada i sortida interactiva per gestionar les dades dels avions.

Nota: veureu que les dades que es demanen dels avions poden variar lleugerament respecte PAC anteriors, segons l'objectiu de l'exercici.

Exercici 1: Tipus estructurats de dades [50%]

Apartat a: [20%] Definiu en llenguatge algorísmic el tipus de dades estructurat *tPlane*, que representa la informació d'un avió. Les dades que es volen guardar són les següents:

- *id*. Identificador de l'avió. Es defineix com un enter.
- *model*. Nom del model de l'avió. Es defineix com un *string*, amb una longitud màxima de 15 caràcters.
- *year*. Any de fabricació de l'avió. Es defineix com un enter.
- *utility*. Ús que se li dona a l'avió, que pot ser COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL o OTHERS. Cal definir prèviament un tipus enumerat que contingui els valors possibles.
- *weight*. Un real que indica el pes de l'avió, en tonelades (sense passatge, ni combustible, ni càrrega).
- *maxSpeed*. Un real que indica la velocitat màxima de l'avió (en Km/h)
- *maxHeight*. Un enter que indica l'altura màxima a la que pot arribar l'avió (en metres).
- *motors*. Un enter que indica el número de motors de l'avió.
- *seats*. Un enter que indica el número de seients.
- *isActive*. Un booleà que indica si l'avió està en actiu o no.

type

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

`tPlane = record`

`id: integer;`

`model: string;`

`year: integer;`

```
utility: tUtility;  
weight: real;  
maxSpeed: real;  
maxHeight: integer;  
motors: integer;  
seats: integer;  
isActive: boolean;  
end type
```

Apartat b: [60%] Implementeu en llenguatge algorísmic un algorisme que permeti donar d'alta de manera interactiva dos nous avions, i desar cada un dels avions en una variable diferent. Per treballar amb aquests dos avions, no s'han d'utilitzar vectors.

L'algorisme va demanant a l'usuari que introdueixi el valor de cada camp de l'avió mitjançant l'entrada i sortida estàndard (teclat/pantalla), i el desa al camp corresponent de l'estructura de dades definida a l'apartat anterior.

Prèviament a la lectura de cada un dels camps, s'ha de mostrar pel canal estàndard de sortida un missatge informatiu d'ajuda que indica a l'usuari els valors possibles o altres informacions d'interès.

Per llegir les dades del tipus *tUtility*, en llenguatge algorísmic disposeu de la funció *readUtility()*. Un cop s'han llegit totes les dades de cada avió, l'algorisme ha de mostrar per pantalla:

- L'identificador de l'avió que té un nombre major de seients entre els avions en actiu i comercials entrats. En cas de dos avions actius i comercials que tinguin el mateix nombre de seients, es mostrarà per pantalla l'identificador del primer avió que s'ha entrat pel canal estàndard. En cas que cap dels dos avions sigui comercial i estigui en actiu, en lloc de l'identificador, s'escriurà un missatge dient que cap dels dos avions entrats està en actiu i és comercial alhora.
- El número total de seients entre els avions en actiu i comercials entrats. En cas que cap dels dos avions estigui en actiu i sigui comercial s'escriurà el valor zero.

algorithm b

```
var  
plane1: tPlane;  
plane2: tPlane;  
totalSeats: integer;  
end var  
  
writeString("Enter id of the plane 1: >> ");  
plane1.id := readInteger();  
writeString("Enter model of the plane 1: >> ");
```

```

plane1.model := readString();
writeString("Enter year of manufacturing of the plane 1: >> ");
plane1.year := readInteger();
writeString("Enter the utility of the plane 1 (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> ");
plane1.utility := readUtility();
writeString("Enter the weight of the plane 1 (in t): >> ");
plane1.weight := readReal();
writeString("Enter the maximum speed of the plane 1 (in km/h): >> ");
plane1.maxSpeed := readReal();
writeString("Enter the maximum height of the plane 1 (in m): >> ");
plane1.maxHeight := readInteger();
writeString("Enter the number of motors of the plane 1: >> ");
plane1.motors := readInteger();
writeString("Enter the number of seats of the plane 1: >> ");
plane1.seats := readInteger();
writeString("Enter if the plane 1 is active (TRUE or FALSE): >> ");
plane1.isActive := readBoolean();

writeString("Enter id of the plane 2: >> ");
plane2.id := readInteger();
writeString("Enter model of the plane 2: >> ");
plane2.model := readString();
writeString("Enter year of manufacturing of the plane 2: >> ");
plane2.year := readInteger();
writeString("Enter the utility of the plane 2 (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> ");
plane2.utility := readUtility();
writeString("Enter the weight of the plane 2 (in t): >> ");
plane2.weight := readReal();
writeString("Enter the maximum speed of the plane 2 (in km/h): >> ");
plane2.maxSpeed := readReal();
writeString("Enter the maximum height of the plane 2 (in m): >> ");
plane2.maxHeight := readInteger();
writeString("Enter the number of motors of the plane 2: >> ");
plane2.motors := readInteger();
writeString("Enter the number of seats of the plane 2: >> ");
plane2.seats := readInteger();
writeString("Enter if the plane 2 is active (TRUE or FALSE): >> ");
plane2.isActive := readBoolean();

totalSeats := 0;
writeString("ID of the plane with more seats: ");

if plane1.isActive and (plane1.utility = COMMERCIAL) then
  if plane2.isActive and (plane2.utility = COMMERCIAL) then
    if plane1.seats ≥ plane2.seats then
      writeInteger(plane1.id);
    else
      writeInteger(plane2.id);
    end if
    totalSeats := plane1.seats + plane2.seats;
  else
    writeInteger(plane1.id);

```

```

        totalSeats := plane1.seats;
    end if
else
    if plane2.isActive and (plane2.utility = COMMERCIAL) then
        writeInteger(plane2.id);
        totalSeats := plane2.seats;
    else
        writeString("No planes found that are active and are commercial.");
    end if
end if

writeString("Total seats: ");
writeInteger(totalSeats);

```

end algorithm

Apartat c: [20%] Explica com es podria enfocar aquest exercici per poder donar d'alta de manera interactiva un número elevat d'avions. No cal fer el disseny, només cal que raoneu la vostra proposta.

Si definim un vector *plane[N]* de dimensió *N* igual al número d'avions necessari i del tipus estructurat *tPlane*, podem accedir a qualsevol dada, per exemple *year*, de qualsevol avió a la posició *i* de la següent manera:

plane[i].year;

Amb qualsevol estructura de control iterativa podrem escriure i llegir tots els avions per molt elevat que sigui el número, anant modificant la variable *i* des de 1 fins a *N*.

Exercici 2: Programació en C [50%]

Apartat a: [70%] Implementeu en **llenguatge C**, l'algorisme de l'exercici anterior.

Recordeu que en llenguatge C cal definir prèviament la longitud màxima de les cadenes de caràcters, típicament mitjançant una constant.

```
/*
** File: main.c
** Author: Daniel Galeano Sancho
** Date: 29-03-2019
** Description: PAC5
*/

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

/* Constants */
#define MAX_LENGTH 15

/* User defined types */
typedef enum {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL,
OTHERS} tUtility;
typedef enum {FALSE, TRUE} bool;
typedef struct{
    int id;
    char model[MAX_LENGTH];
    int year;
    tUtility utility;
    float weight;
    float maxSpeed;
    int maxHeight;
    int motors;
    int seats;
    bool isActive;
}tPlane;

/* Main function */
int main(int argc, char **argv){

    /* Variables */
    tPlane plane1;
    tPlane plane2;
    int totalSeats;

    /* Initialization of variables */
    totalSeats = 0;

    /* Reading plane 1 data */
    printf("Enter id of the plane 1: >> ");
    scanf("%d",&plane1.id);
    printf("Enter model of the plane 1: >> ");
    getchar();
    gets(plane1.model);
    printf("Enter year of manufacturing of the plane 1: >> ");
    scanf("%d",&plane1.year);
    printf("Enter the utility of the plane 1 (0 for COMMERCIAL, 1 for
PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for
OTHERS): >> ");
```

```

scanf("%u",&plane1.utility);
printf("Enter the weight of the plane 1 (in t): >> ");
scanf("%f",&plane1.weight);
printf("Enter the maximum speed of the plane 1 (in km/h): >> ");
scanf("%f",&plane1.maxSpeed);
printf("Enter the maximum height of the plane 1 (in m): >> ");
scanf("%d",&plane1.maxHeight);
printf("Enter the number of motors of the plane 1: >> ");
scanf("%d",&plane1.motors);
printf("Enter the number of seats of the plane 1: >> ");
scanf("%d",&plane1.seats);
printf("Enter if the plane 1 is active (0 for FALSE, 1 for TRUE): >>
");
scanf("%u",&plane1.isActive);

    /* Reading plane 2 data */
printf("\nEnter id of the plane 2: >> ");
scanf("%d",&plane2.id);
printf("Enter model of the plane 2: >> ");
getchar();
    gets(plane2.model);
printf("Enter year of manufacturing of the plane 2: >> ");
scanf("%d",&plane2.year);
printf("Enter the utility of the plane 2 (0 for COMMERCIAL, 1 for
PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for
OTHERS): >> ");
scanf("%u",&plane2.utility);
printf("Enter the weight of the plane 2 (in t): >> ");
scanf("%f",&plane2.weight);
printf("Enter the maximum speed of the plane 2 (in km/h): >> ");
scanf("%f",&plane2.maxSpeed);
printf("Enter the maximum height of the plane 2 (in m): >> ");
scanf("%d",&plane2.maxHeight);
printf("Enter the number of motors of the plane 2: >> ");
scanf("%d",&plane2.motors);
printf("Enter the number of seats of the plane 2: >> ");
scanf("%d",&plane2.seats);
printf("Enter if the plane 2 is active (0 for FALSE, 1 for TRUE): >>
");
scanf("%u",&plane2.isActive);

printf("\nID of the plane with more seats: ");

if (plane1.isActive && (plane1.utility == COMMERCIAL) ){
    if (plane2.isActive && (plane2.utility == COMMERCIAL) ){
        if (plane1.seats >= plane2.seats) {
            printf("%d",plane1.id);
        }else{
            printf("%d",plane2.id);
        }
        totalSeats = plane1.seats + plane2.seats;
    }else{
        printf("%d",plane1.id);
        totalSeats = plane1.seats;
    }
}
else{
    if (plane2.isActive && (plane2.utility == COMMERCIAL) ){
        printf("%d",plane2.id);
        totalSeats = plane2.seats;
    }else{

```

```

        printf("No planes were found that are active and are
commercial.");
    }
}

printf("\nTotal seats: %d\n",totalSeats);

return 0;
}

```

Apartat b: [30%] Com a les anteriors PAC es demana que mostreu el funcionament de l'algorisme **fent jocs de prova**. És a dir que completeu les taules següents indicant, per a uns valors d'avions que s'han introduït a les diferents estructures, quina sortida s'espera en l'execució del programa. Després comproveu que efectivament el programa fa el que esperàveu. Podeu afegir més files a les taules.

Amb l'objectiu que aquest apartat sigui més intel·ligible i, degut a que hi ha variables que es demanen però que, tinguin el valor que tinguin, no s'utilitzen ni es mostren a la sortida, al següent joc de proves s'ometen les següents variables: *model*, *year*, *weight*, *maxSpeed*, *maxHeight* i *motors*.

També recordem que quan la variable *utility* val *COMMERCIAL*, correspon al valor enter 0.

b1) Algun dels dos avions introduïts està en actiu i és comercial:

Dades d'entrada	
Nom variable	Valor entrada
plane1.id	10
plane1.utility	0
plane1.seats	200
plane1.isActive	1
plane2.id	20
plane2.utility	3
plane2.seats	300
plane2.isActive	1

Dades de sortida
ID of the plane with more seats: 10
Total seats: 200

b2) Els dos avions introduïts estan en actiu i són comercials, i tenen el mateix nombre de seients:

Dades d'entrada	
Nom variable	Valor entrada
plane1.id	10
plane1.utility	0
plane1.seats	200
plane1.isActive	1
plane2.id	20
plane2.utility	0
plane2.seats	200
plane2.isActive	1

Dades de sortida
ID of the plane with more seats: 10
Total seats: 200

b3) Cap dels dos avions introduïts està en actiu i és comercial a l'hora.

Dades d'entrada	
Nom variable	Valor entrada
plane1.id	10
plane1.utility	3
plane1.seats	200
plane1.isActive	1
plane2.id	20
plane2.utility	0
plane2.seats	200
plane2.isActive	0

Dades de sortida
ID of the plane with more seats: No planes were found that are active and are commercial
Total seats: 0

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 alternativa i el tipus de dades estructural.
- Que es raoni correctament la resposta de l'apartat c de l'exercici 1.

A l'exercici 2:

- Que el programa s'adeqüi a les indicacions donades.
- Que el programa compili i funcioni 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 xWiki de contingut.
- Que es declarin els tipus adequats segons el tipus de dades que representa.