

# Fonaments de Programació

## PAC1 - 20182

Data límit de lliurament: 04/03/2019

Estudiant

Cognoms: **GALEANO SANCHO**

Nom: **DANIEL**

### Objectius

- Saber identificar les variables de tipus bàsics de dades necessàries per a resoldre un problema.
- Saber llegir del teclat i escriure per pantalla dades de tipus bàsics

### Format i data de lliurament

Cal lliurar la PAC abans del dia 04 de març de 2019 a les 23:59.

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

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

Cal fer el lliurament a l'apartat de lliuraments d'AC de l'aula de teoria.

### Enunciat

Aquest semestre la companyia UOCAirways ens han demanat crear una aplicació per gestionar un companyia de transport aeri, els avions, els passatgers i els vols.

Per donar resposta a aquesta petició, a través de les PAC, anirem creant una petita part de l'aplicació, que gestionarà els avions. La resta, ho completarem a les pràctiques.

Per començar ens demanen el següent:

### Exercici 1: Declaració de variables [50%]

Dissenyar, en llenguatge algorísmic, un algorisme que faci el següent:

Apartat a [40%]. Declarar les variables i tipus enumerats necessaris per gestionar les dades d'un avió. Per a cada avió, de moment, es necessita guardar la següent informació:

- Un identificador de tipus enter, que és el número que identifica l'avió.
- Un caràcter que indiqui el model de l'avió.
- Un enter que indiqui l'any de fabricació de l'avió.
- Un enumerat que indiqui el tipus d'ús que té l'avió i que pot ser: COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL, OTHERS.
- Un real que indiqui el pes de l'avió buit en tones (sense passatgers ni combustible ni càrrega).
- Un real que indiqui la velocitat màxima de l'avió (en Km/h).
- Un enter que indiqui l'alçada màxima a la que pot arribar l'avió (en metres).
- Un enter que indiqui el nombre de motors.
- Un enter que indiqui el nombre de seients.
- Un booleà que indiqui si l'avió està actiu o no.

Apartat b [30%]. Demanar i llegir pel canal estàndard d'entrada els valors de les variables de tipus enter, caràcter i real declarades a l'apartat anterior d'un avió concret.

Per a la lectura s'ha d'indicar pel canal estàndard de sortida quina informació s'espera que l'usuari introdueixi. Per exemple, per llegir una variable *age* corresponent a l'edat d'una persona, caldria fer:

```
writeString("Enter the current age:");  
age := readInteger();
```

Apartat c [30%]. Mostrar pel canal estàndard de sortida el valor de les variables llegides de l'avió, indicant què és cada informació. Seguint amb l'exemple faríem:

```
writeString("Age:");  
writeInteger(age);
```

**type**

tUse = {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL, OTHERS};

**end type**

**algorithm** UOCAirways

**var**

id: **integer**;  
model: **string**;  
age: **integer**;  
use: tUse;  
weight: **real**;  
speed: **real**;  
height: **integer**;  
engines: **integer**;  
seats: **integer**;  
active: **boolean**;

**end var**

```
writeString("Type the plane identification number: ");
id := readInteger();
writeString("Type the plane model: ");
model := readString();
writeString("Type the plane manufacturing year: ");
age := readInteger();
writeString("Type the plane's use (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): ");
use := readUse();
writeString("Type the plane weight in tons: (without passengers, gas and load) ");
weight := readReal();
writeString("Type the plane speed in km/h: ");
speed := readReal();
writeString("Type the plane maximum height in meters: ");
height := readInteger();
writeString ("Type the plane number of engines: ");
engines := readInteger();
writeString ("Type the plane number of seats: ");
seats := readInteger();
writeString ("Type if the plane is active (0 for FALSE , 1 for TRUE): ");
active := readBoolean();

writeString("ID: ");
writeInteger(id);
writeString ("Model: ");
writeString(model);
writeString ("Age: ");
writeInteger(age);
writeString("Use (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR,
4 for EXPERIMENTAL, 5 for OTHERS): ");
writeUse(use);
```

```

writeString ("Weight in tons: ");
writeReal(weight);
writeString ("Speed in km/h: ");
writeReal(speed);
writeString ("Maximum height in meters: ");
writeInteger(height);
writeString ("Number of engines: ");
writeInteger(engines);
writeString ("Number of seats: ");
writeInteger(seats);
writeString ("Active: (0 for FALSE , 1 for TRUE) ");
writeBoolean(active);

```

**end algorithm**

## Exercici 2: programació en C [50%]

### Apartat a [70%] Codificació

Codificar en C l'algorisme de l'exercici 1. En la programació els reals s'han d'escriure amb dos decimals.

```

/*
** File: main.c
** Author: Daniel Galeano Sancho
** Date: 26-02-2019
** Description: PAC1, Exercici 2
*/

/* Fitxers de capçalera */
#include <stdio.h>

int main(int argc, char **argv)
{
    /* Declaració de tipus */
    typedef enum {COMMERCIAL, PRIVATE, GOVERNMENTAL, MILITAR, EXPERIMENTAL,
OTHERS} tUse;
    typedef enum {FALSE, TRUE} bool;

    /* Declaració de variables */
    int id;
    char model;
    int manufacturingYear;
    tUse use;
    float weight;
    float speed;
    int maximumHeight;
    int engines;
    int seats;
    bool active;

```

```

/* Lectura de dades de l'avió */
printf("Type the plane identification number: \n");
scanf("%d",&id);
printf("Type the plane model: (max 1 character) \n");
scanf(" %c",&model);
printf("Type the plane manufacturing year: \n");
scanf("%d",&manufacturingYear);
printf("Type the plane's use (0 for COMMERCIAL, 1 for PRIVATE, 2 for
GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): \n");
scanf("%d",&use);
printf("Type the plane weight in tons: (without passengers, gas and load)
\n");
scanf("%f",&weight);
printf("Type the plane speed in km/h: \n");
scanf("%f",&speed);
printf("Type the plane maximum height in meters: \n");
scanf("%d",&maximumHeight);
printf("Type the plane number of engines: \n");
scanf("%d",&engines);
printf("Type the plane number of seats: \n");
scanf("%d",&seats);
printf("Type if the plane is active (0 for FALSE , 1 for TRUE): \n");
scanf("%d",&active);

/* Es mostren les dades de l'avió per pantalla */
printf("\n\nID: %d\n",id);
printf("Model: %c\n",model);
printf("Manufacturing year: %d\n",manufacturingYear);
printf("Use (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for
MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): %d\n",use);
printf("Weight: %.2f tons\n",weight);
printf("Speed: %.2f km/h\n",speed);
printf("Maximum height: %d m\n",maximumHeight);
printf("Number of engines: %d\n",engines);
printf("Number of seats: %d\n",seats);
printf("Active (0 for FALSE , 1 for TRUE): %d\n",active);

return 0;
}

```

### Apartat b [30%] Proves / Execució de l'algorisme

Els algorismes codificats cal sempre provar-los. És a dir, cal executar els programes donant diferents valors a les variables d'entrada i comprovant que la sortida correspon als valors esperats. D'això se'n diu fer *Jocs de proves*. Els jocs de prova han de ser diferents per comprovar les diferents situacions que pot executar el programa.

### Apartat b1 [50%]

En aquest apartat es demana que dissenyeu un joc de prova. És a dir que completeu la taula següent indicant per uns valors d'entrada concrets, quina sortida s'espera en l'execució del programa. Després comproveu que efectivament el programa fa el que esperaveu.

Dades d'entrada		Dades de sortida
Nom variable	Valor entrada	
id	12345	12345
model	A	A
manufacturingYear	1988	1988
use	3	3
weight	100	100.00
speed	900	900.00
maximumHeight	12000	12000
engines	4	4
seats	200	200
active	1	1

### Apartat b2 [50%].

Executeu el programa entrant dades incorrectes, per exemple un real quan s'espera un enter o un caràcter quan s'espera un enter. Observeu que passa en l'execució i comenteu el resultat.

Quan el programa ens pregunta per la variable entera "id", entrem el valor real "1.1", i l'execució del programa no funciona com voldríem, degut a que emmagatzema el valor "1" a la variable "id", i el valor ".1" a la variable "model", sense haver-nos preguntat sobre aquesta variable i passant a preguntar directament sobre la variable "manufacturingYear". A més, la execució del programa ens informa de que alguna cosa ha anat malament mostrant el següent missatge:

```
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
```

Si ara, entrem un caràcter a la variable "id", veiem que torna a saltar-se la línia en la qual ens demana el valor de la variable "model", però aquest cop, s'executa normalment. L'error es veu quan el programa ens mostra els valors de les variables "id" i "model". Pel que fa a la variable "model", veiem que ha emmagatzemat el valor que li havíem assignat a la variable "id", però si ens fixem el valor que té la variable "id", veiem que és un valor arbitrari que no s'ha assignat en cap moment, com es mostra a continuació:

```
ID: -1208410112
```

## **Creris de correcció:**

### A l'exercici 1:

- Que es segueixi la notació algorísmica utilitzada a l'assignatura. Vegeu document *Nomenclator* a la xWiki de contingut.
- Que es segueixin les instruccions donades i l'algorisme respongui al problema plantejat

### 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.
- En la codificació en C de l'algorisme, que es declarin els tipus adequats segons el tipus de dades que representa.
- 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.