

Fonaments de Programació

PAC2 - 20182

Data límit de lliurament: 11/03/2019

Estudiant

Cognoms: GALEANO SANCHO

Nom: DANIEL

Objectius

- Saber dissenyar expressions algorísmiques
- Aprendre a fer operacions simples amb punters

Format i data de lliurament

Cal lliurar la PAC abans del dia 11 de març 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.
- El workspace de Codelite que contingui els fitxers .c demanats a l'apartat a de l'exercici 2.

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

Enunciat

La companyia UOCAirways té el següent algorisme a mig dissenyar.

type

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

end type

```

const
    WEIGHT_EMPTY1: real = 41415.0;
    MAX_WEIGHT_WITHOUT_GAS1: real = 61689.0
    NUM_PASSENGERS_PER_FLIGHT_ATTENDANT: integer = 50;
    AVERAGE_WEIGHT: real = 95.0;
end const

algorithm UOCRailways

var
    utility1, utility2, utility3: tUtility;
    year1, year2, year3: integer;
    seats1, seats2, seats3: integer;
    isActive1, isActive2, isActive3: boolean;
end var

utility1:= COMMERCIAL;
seats1:= 200;
.....
end algorithm

```

I, ens demanen que resolguem els següents exercicis:

Exercici 1: Disseny en llenguatge algorísmic [50%]

Completar l'algorisme per tal que:

- a. [10%] Llegeixi pel canal estàndard les següents dades corresponents a tres avions i les emmagatzemi en aquestes variables ja declarades.

- year1, year2, year3
- utility2, utility3
- seats2, seats3
- isActive1, isActive2, isActive3

Fixeu-vos que seats1 i utility1 tenen ja valors assignats dins del codi i no cal llegir els seus valors.

- b. [80%] Amb les dades llegides i les constants definides, calculi les següents expressions i les assigni a una variable de tipus adequat:

b1. [20%] Una expressió que calculi la càrrega màxima que pot portar l'avió 1, que és de passatgers (COMMERCIAL), en funció del passatge real que porti (dada que cal llegir del canal estàndard), sabent que

- El pes de l'avió buit és de 41415 Kg (WEIGHT_EMPTY1).

- El pes màxim que pot portar sense tenir en compte el combustible és de 64689 Kg (MAX_WEIGHT_WITHOUT_GAS1).
- La tripulació consta del comandant, el copilot i una persona de cabina per cada grup de fins a 50 passatgers. El vol sempre porta almenys 1 passatger.
- El pes mitjà per persona, incloent l'equipatge de mà, és de 95 Kg. (AVERAGE_WEIGHT). El personal de cabina també porta equipatge de mà.

b2. [20%] Una expressió que retorni cert si hi ha almenys un avió de tipus privat amb una capacitat superior o igual a 15 passatgers i fals en cas contrari.

b3. [20%] Una expressió que retorni cert si els tres avions són del mateix any i fals en cas contrari. Per fer aquest càlcul, ens han volgut ajudar i ens han donat la següent expressió

`year1 = year2 = year3`

- La podem utilitzar? És sintàcticament correcte?
- Si no ho és, abans de resoldre l'apartat expliqueu perquè i corregiu-la perquè faci el càlcul demanat correctament.

b4.[20%] Una expressió que retorni cert si els tres avions estan en el mateix estat, és a dir els tres actius o els tres no actius i fals en cas contrari. Per fer aquest càlcul, ens han volgut ajudar i ens han donat la següent expressió

`isActive1 = isActive2 = isActive3`

- La podem utilitzar? És sintàcticament correcte? Resol el problema que ens plantegen?
- Si no ho és, abans de resoldre l'apartat expliqueu perquè i corregiu-la perquè faci el càlcul demanat correctament

c. [10%] Escrigui pel canal estàndard el resultat de les quatre expressions de l'apartat b.

Es poden definir les variables auxiliars que calgui per guardar el resultat de cadascuna de les expressions demanades.

Nota:

- per llegir el valor de les variables de tipus enumerat *tUtility*, podeu utilitzar la funció *readUtility()*. Aquesta funció llegeix una variable del canal estàndard d'entrada de tipus *tUtility* i retorna el valor llegit.

type

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

end type

const

WEIGHT_EMPTY1: **real** = 41415.0;

MAX_WEIGHT_WITHOUT_GAS1: **real** = 61689.0;

NUM_PASSENGERS_PER_FLIGHT_ATTENDANT: **integer** = 50;

AVERAGE_WEIGHT: **real** = 95.0;

end const

algorithm UOCAirways

var

utility1, utility2, utility3: tUtility

year1, year2, year3: **integer**;

seats1, seats2, seats3: **integer**;

passengers1: **integer**;

isActive1, isActive2, isActive3: **boolean**;

b1: **real**;

b2, b3, b4: **boolean**;

end var

utility1 := COMMERCIAL;

seats1 := 200;

writeString("Enter the manufacturing year of the plane 1: >> ");

year1 := readInteger();

writeString("Enter the manufacturing year of the plane 2: >> ");

year2 := readInteger();

writeString("Enter the manufacturing year of the plane 3: >> ");

year3 := 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): >> ");

utility2 := readUtility ();

writeString("Enter the utility of the plane 3 (0 for COMMERCIAL, 1 for PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS): >> ");

utility3 := readUtility ();

writeString("Enter the number of seats of the plane 2: >> ");

seats2 := readInteger();

writeString("Enter the number of seats of the plane 3: >> ");

seats3 := readInteger();

writeString ("Enter if the plane 1 is active (FALSE or TRUE): >> ");

isActive1 := readBoolean();

writeString ("Enter if the plane 2 is active (FALSE or TRUE) : >>");

isActive2 := readBoolean();

```
writeString ( "Enter if the plane 3 is active (FALSE or TRUE) : >>");  
isActive3 := readBoolean();
```

```
writeString( "Enter the number of passengers of plane 1: >> ");  
passengers1 := readInteger();
```

```
{ Apartat b1 }  
b1 := MAX_WEIGHT_WITHOUT_GAS1  
      - WEIGHT_EMPTY1  
      - AVERAGE_WEIGHT*integerToReal(passengers1 + 2 + ((passengers1-1)/50)+1);
```

```
{ Apartat b2 }  
b2 := (utility1 = PRIVATE and seats1 >= 15) or  
      (utility2 = PRIVATE and seats2 >= 15) or  
      (utility3 = PRIVATE and seats3 >= 15);
```

```
{ Apartat b3 }  
{ L'expressió year1 = year2 = year3 no la podem utilitzar perquè no és sintàcticament correcta  
perquè la primera comparació (year1 = year2) dona un resultat tipus booleà i s'està comparant  
amb year3 que és de tipus enter. Per tant, l'hem d'utilitzar realitzant una modificació per tal que  
sigui correcta. }  
b3 := (year1 = year2) and (year2 = year3);
```

```
{ Apartat b4 }  
{ L'expressió isActive1 = isActive2 = isActive3 la podríem utilitzar perquè és sintàcticament  
correcta. El problema és que no ens dona el que demana l'apartat. Aquesta expressió sí que ens  
retorna cert si els tres avions estan actius, però també dona cert quan l'avió 1 i l'avió 2 estan a  
diferent estat i l'avió 3 no està actiu, entre d'altres. Es resol utilitzant la mateixa expressió que en  
l'apartat anterior. }  
b4 := (isActive1 = isActive2) and (isActive2 = isActive3);
```

```
writeString( "Apartat b1: ");  
writeReal(b1);  
writeString( "Apartat b2: ");  
writeBoolean(b2);  
writeString( "Apartat b3: ");  
writeBoolean(b3);  
writeString( "Apartat b4: ");  
writeBoolean(b4);
```

end algorithm

Exercici 2: Programació en C [50%]

Apartat a [70%] Codificació

Implementar en llenguatge C l'algorisme anterior.

Notes:

- En C, per llegir els tipus enumerats cal utilitzar el format “%u”.
Per exemple, si tenim el següent enumerat:

```
(...)  
typedef enum {ELECTRICAL, CARBON, SOLAR, GASOLINE, DIESEL,  
             MAGNETIC } tPropulsion;  
tPropulsion propulsion;  
(...)
```

Per llegir la variable propulsion escriurem:

```
scanf("%u", &propulsion);
```

- Recordeu que en C no existeix el tipus booleà així que cal declarar els booleans com un tipus enumeratiu com s'indica als materials.

```
/*  
** File: main.c  
** Author: Daniel Galeano Sancho  
** Date: 06-03-2019  
** Description: PAC2, Exercici 2  
*/  
  
/* System header file */  
#include <stdio.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 float WEIGHT_EMPTY1=41415.0;  
    const float MAX_WEIGHT_WITHOUT_GAS1=61689.0;  
    const unsigned int NUM_PASSENGERS_PER_FLIGHT_ATTENDANT=50;  
    const float AVERAGE_WEIGHT=95.0;  
  
    /* Variables */  
    tUtility utility1, utility2, utility3;  
    unsigned int year1, year2, year3;  
    unsigned int seats1, seats2, seats3;  
    unsigned int passengers1;  
    bool isActive1, isActive2, isActive3;
```

```

float b1;
bool b2, b3, b4;

utility1 = COMMERCIAL;
seats1 = 200;

printf("Enter the manufacturing year of the plane 1: >> ");
scanf("%d",&year1);
printf("Enter the manufacturing year of the plane 2: >> ");
scanf("%d",&year2);
printf("Enter the manufacturing year of the plane 3: >> ");
scanf("%d",&year3);

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",&utility2);
printf("Enter the utility of the plane 3 (0 for COMMERCIAL, 1 for
PRIVATE, 2 for GOVERNMENTAL, 3 for MILITAR, 4 for EXPERIMENTAL, 5 for OTHERS):
>> ");
scanf("%u",&utility3);

printf("Enter the number of seats of the plane 2: >> ");
scanf("%d",&seats2);
printf("Enter the number of seats of the plane 3: >> ");
scanf("%d",&seats3);

printf("Enter if the plane 1 is active (0 for FALSE, 1 for TRUE): >> ");
scanf("%u",&isActive1);
printf("Enter if the plane 2 is active (0 for FALSE, 1 for TRUE) : >> ");
scanf("%u",&isActive2);
printf("Enter if the plane 3 is active (0 for FALSE, 1 for TRUE) : >> ");
scanf("%u",&isActive3);

printf("Enter the number of passengers of plane 1: >> ");
scanf("%d",&passengers1);

/* Apartat b1 */
b1 = MAX_WEIGHT_WITHOUT_GAS1-WEIGHT_EMPTY1-
AVERAGE_WEIGHT*((float)(passengers1 + 2 + ((passengers1-
1)/NUM_PASSENGERS_PER_FLIGHT_ATTENDANT)+1));

/* Apartat b2 */
b2 =
((utility1==PRIVATE)&&(seats1>=15))||((utility2==PRIVATE)&&(seats2>=15))||((uti
lity3==PRIVATE)&&(seats3>=15));

/* Apartat b3 */
/* L'expressió year1 = year2 = year3 no la podem utilitzar perquè
no és sintàcticament correcta perquè la primera comparació (year1 =
year2)
dona un resultat tipus booleà i s'està comparant amb year3 que és de
tipus
enter. Per tant, l'hem d'utilitzar realitzant una modificació per tal
que
sigui correcta. */
b3 = (year1==year2)&&(year2==year3);

/* Apartat b4 */
/* L'expressió isActive1 = isActive2 = isActive3 la podríem
utilitzar perquè és sintàcticament correcta. El problema és

```

```

    que no ens dona el que demana l'apartat. Aquesta expressió sí
    que ens retorna cert si els tres avions estan actius, però també
    dona cert quan l'avió 1 i l'avió 2 estan a diferent estat i l'avió
    3 no està actiu, entre d'altres. Es resol utilitzant la mateixa
    expressió que en l'apartat anterior. */
    b4 = (isActive1==isActive2) && (isActive2==isActive3);

    printf("Apartat b1: %.2f kg\n",b1);
    printf("Apartat b2 (0 for FALSE, 1 for TRUE): %u\n",b2);
    printf("Apartat b3 (0 for FALSE, 1 for TRUE): %u\n",b3);
    printf("Apartat b4 (0 for FALSE, 1 for TRUE): %u\n",b4);

    return 0;
}

```

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

A l'igual que en la PAC1, es demana que indiqueu les dades de sortida esperada de cada expressió en funció de les dades d'entrada. Indiqueu dos jocs de prova per cada expressió.

Expressió1

	Input Data	Output
Case	Num. Passengers	Cargo
1	20	18089
2	70	13244

Expressió2

	Input Data						Output
Case	utility1	utility2	utility3	seats1	seats2	seats3	result
1	COMMERCIAL	PRIVATE	MILITAR	200	16	50	TRUE
2	COMMERCIAL	PRIVATE	PRIVATE	200	10	12	FALSE

Expressió3

	Input Data			Output
Case	year1	year2	year3	result
1	1988	1988	1988	TRUE
2	1987	1988	1988	FALSE

Expressió4

	Input Data			Output
Case	isActive1	isActive2	isActive3	result
1	TRUE	TRUE	TRUE	TRUE
2	FALSE	FALSE	FALSE	TRUE
3	FALSE	TRUE	FALSE	FALSE

Criteris 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 compleixin les instruccions donades i l'algorisme respongui al problema plantejat.
- Que les expressions dissenyades siguin formalment correctes.
- Que per les expressions b3 i b4 és faci un correcte raonament de si són o no correctes.

A l'exercici 2:

- Que el programa compila.
- Que el programa 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 Wiki de contingut.
- Es valorarà que es declarin en C els tipus enters més eficients pel que fa a l'ús de memòria segons el tipus de dades que cal representar.
- Que els jocs de prova donats provin diferents casuístiques del programa.