



## Fonaments de Programació

### Pract1 - Primera Pràctica

#### Presentació

En aquest document es presenta la primera pràctica. Aquesta prova té caràcter obligatori i constitueix el 50% de la nota final de pràctiques.

#### Competències

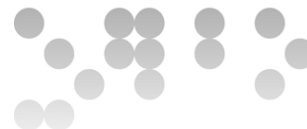
Les principals competències del grau que es treballen en aquesta pràctica són les següents:

- Capacitat per a analitzar un problema en el nivell d'abstracció adequat a cada situació i aplicar les habilitats i coneixements adquirits per a abordar-lo i resoldre'l.
- Capacitat de dissenyar i construir aplicacions informàtiques mitjançant tècniques de desenvolupament, integració i reutilització.

Per treballar aquestes competències, la pràctica planteja diversos exercicis al voltant de l'ús d'**accions i funcions** i **tractament seqüencial** de dades.

#### Objectius

- Dissenyar funcions i/o accions per a resoldre un problema.
- Treballar amb la noció de seqüència
- Triar adequadament els esquemes de tractament seqüencial que cal aplicar per a resoldre un problema.
- Codificar un algorisme en C.
- Fer el seguiment, pas a pas, de l'execució d'un algorisme codificat en C
- Fer proves per a veure si el funcionament és correcte.



## Descripció de la pràctica a realitzar

La pràctica consta de 3 exercicis, que treballen els diferents objectius que s'han enumerat com a claus en aquesta prova. Cada exercici consta d'un enunciat i un pes que indica el percentatge que representa respecte a la nota global.

## Recursos

Serà necessari llegir prèviament el capítol del temari relatiu a l'ús d'accions i funcions, tant dels apunts de l'assignatura com de les TS que trobareu en les cites del calendari de la vostra aula.

Adicionalment, per resoldre l'exercici de codificació, caldrà que reviseu les taules de traducció i tot el material relacionat amb la codificació en C que trobareu a l'apartat de "recursos" de la vostra aula.

## Criteris de valoració

S'avaluarà cada exercici d'acord amb el seu pes, que apareix indicat abans del seu enunciat. Els exercicis que constin de diversos apartats es valoraran com a completament correctes o incorrectes. Les respostes incorrectes no disminueixen la nota. Raoneu i justifiqueu totes les respostes, especialment si així ho indica l'enunciat (si, tot i demanar-ho l'enunciat, no justifiqueu la resposta, es comptabilitzarà com a incorrecte).

## Format i data de lliurament

Cal lliurar la solució dels exercicis 1 i 2 en un fitxer anomenat CognomsNom\_FP\_Pract1 adreçat a la bústia "Lliurament d'activitats" de l'aula de teoria. Cal lliurar la codificació de la pregunta 3 a través del Corrector Automàtic, que trobareu a l'aula de teoria.

Data límit per lliurar la solució: **dimecres, 19 de novembre de 2014** (a les 23:59 hores).



## Exercici 1. Treball amb seqüències [30%]

**Objectius:** Donada una seqüència de dades, aplicar l'esquema adequat per fer una acció/funció que solucioni un determinat problema.

**Materials:** Mòdul 3: Tractament seqüencial.

**Tasca:** Al finalitzar cada curs acadèmic, per tal de poder determinar les llistes d'accés a la universitat, es realitzen les PAU (Proves d'Accés a la Universitat), mes conegudes com "La Selectivitat". La realització d'aquestes proves té lloc en diferents centres arreu de Catalunya. En cada un d'aquests centres les proves s'estructuren en tribunals, de manera que cada estudiant té assignat un tribunal en que haurà de fer els exàmens de totes les matèries de que s'ha d'examinar. D'aquesta manera, els resultats de les proves queden també enregistrats en base a aquesta estructura de centres docents i tribunals.

Les qualificacions queden doncs estructurades per cada tribunal en una seqüència amb el format següent:

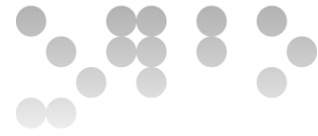
```
<idChair idSubject1 numEst1 idEst(1)(1) Marks(1)(1) idEst(1)(2) Marks(1)(2) ... idEst(1)(N1)
Marks(1)(N1) idSubject2 numEst2 idEst(2)(1) Marks(2)(1) idEst(2)(2) Marks(2)(2) ... idEst(2)(N2)
Marks(2)(N2) .... idSubjectM numEstM idEst(M)(1) Marks(M)(1) idEst(M)(2) Marks(M)(2) ... idEst(M)(NM)
Marks(M)(NM) 0>
```

on :

- $idChair_i$  és un enter positiu que identifica el tribunal.
- $idSubject_i$  és un enter més gran que zero que codifica la matèria dels resultats que venen a continuació.
- $numEst_i$  és un enter positiu que indica el nombre d'exàmens que hi ha hagut de la matèria  $idSubject_i$  en el tribunal  $idChair_i$
- $idEst_{(i)(j)}$  és un enter positiu que identifica l'estudiant. Aquest codi és el mateix per a totes les matèries de que s'ha examinat l'estudiant
- $Marks_{(i)(j)}$  és un real que indica la nota que ha tret l'estudiant  $idEst_{(i)(j)}$  en la matèria  $idSubject_i$ .
- **END** és la marca final de seqüència i val 0.

Anotacions:

- La seqüència contindrà, com a mínim, la identificació del tribunal i la marca final.
- Cada matèria contindrà com a mínim un estudiant.
- No tots els estudiants tenen perquè aparèixer en cada una de les matèries.
- Cada estudiant té un codi únic que l'identifica en totes les matèries.
- Les dades de la seqüència són coherents.



Partint d'una seqüència de notes per a un tribunal, que es llegirà del canal d'entrada estàndard, ens demanen que **dissenyem una acció** que escrigui pel canal de sortida estàndard la nota mitja que s'ha obtingut en una matèria l'identificador de la qual es passarà com a paràmetre a l'acció. En la seqüència de sortida s'escriurà en primer lloc el identificador del tribunal. Després, el identificador de la matèria seguit de la nota mitjana. La seqüència acabarà amb el codi de fi de seqüència 0.

Per exemple, suposem que l'entrada està formada per la seqüència següent:

<312 12 3 1 5.6 2 8.7 3 7.7 14 3 2 9.1 5 6.7 4 5.6 7 4 2 4.5 7 5.6 4 4.5 9 6.7 0>

Per tant, si volem saber la nota mitjana de la matèria 14, a la sortida tindrem:

<312 14 7.1 0>

mentre que si volem saber la nota mitjana de la matèria 10 a la sortida tindrem

<312 0>



## Exercici 2. Treball amb seqüències [35%]

**Objectius:** Donada una seqüència de dades, aplicar l'esquema adequat per fer una acció/funció que solucioni un determinat problema.

**Materials:** Mòdul 3: Tractament seqüencial.

**Tasca:** Al finalitzar cada curs acadèmic, per tal de poder determinar les llistes d'accés a la universitat, es realitzen les PAU (Proves d'Accès a la Universitat), mes conegudes com "La Selectivitat". La realització d'aquestes proves té lloc en diferents centres arreu de Catalunya. En cada un d'aquests centres les proves s'estructuren en tribunals, de manera que cada estudiant té assignat un tribunal en que haurà de fer els exàmens de totes les matèries de que s'ha d'examinar. D'aquesta manera, els resultats de les proves queden també enregistrats en base a aquesta estructura de centres docents i tribunals.

Les qualificacions queden doncs estructurades per cada tribunal en una seqüència amb el format següent:

```
<idChair idSubject1 numEst1 idEst(1)(1) Marks(1)(1) idEst(1)(2) Marks(1)(2) ... idEst(1)(N1)
Marks(1)(N1) idSubject2 numEst2 idEst(2)(1) Marks(2)(1) idEst(2)(2) Marks(2)(2) ... idEst(2)(N2)
Marks(2)(N2) .... idSubjectM numEstM idEst(M)(1) Marks(M)(1) idEst(M)(2) Marks(M)(2) ... idEst(M)(NM)
Marks(M)(NM) 0>
```

on :

- $idChair_i$  és un enter positiu que identifica el tribunal.
- $idSubject_i$  és un enter més gran que zero que codifica la matèria dels resultats que venen a continuació.
- $numEst_i$  és un enter positiu que indica el nombre d'exàmens que hi ha hagut de la matèria  $idSubject_i$  en el tribunal  $idChair_i$ .
- $idEst_{(i)(j)}$  és un enter positiu que identifica l'estudiant. Aquest codi és el mateix per a totes les matèries de que s'ha examinat l'estudiant.
- $Marks_{(i)(j)}$  és un real que indica la nota que ha tret l'estudiant  $idEst_{(i)(j)}$  en la matèria  $idSubject_i$ .
- **END** és la marca final de seqüència i val 0.

Anotacions:

- La seqüència contindrà, com a mínim, l'identificador de tribunal i la marca final.
- Cada matèria contindrà com a mínim un estudiant.
- No tots els estudiants tenen perquè aparèixer en cada una de les matèries.
- Cada estudiant té un codi únic que l'identifica en totes les matèries.
- Les dades de la seqüència són coherents.



D'aquesta manera si considerem tots els tribunals junts, en resulta una seqüència com la que segueix:

$\langle \text{TRIB}_1 \text{ TRIB}_2 \text{ TRIB}_3 \dots \text{TRIB}_N -1 \rangle$

on cada un dels  $\text{TRIB}_i$  es una (sub)seqüència amb la informació de cada tribunal segons el format descrit mes amunt. La seqüència global acaba amb l'indicador -1.

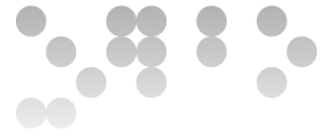
Partint d'una seqüència de notes com la indicada en la que tenim, de manera consecutiva, les notes de tots els tribunals, que es llegirà del canal d'entrada estàndard, ens demanen que **dissenyem una acció** que escrigui pel canal de sortida estàndard aquells tribunals per als que hi ha alguna matèria en la que han aprovat menys del 25% dels estudiants presentats. A la vegada es desitja també saber, per aquestes matèries, l'estudiant que, en aquest tribunal, ha tret la millor nota. La seqüència de sortida obeeirà al següent format:

- En la seqüència de sortida s'escriurà, per cada tribunal, el codi del tribunal, seguit del codi de matèria i percentatge d'aprovat per totes aquelles matèries en que els aprovats siguin menors del 25% així com l'identificador de l'estudiant que ha tret millor nota i aquesta nota.
- La relació de matèries, per cada tribunal, amb menys de 25% d'aprovat finalitzarà amb un 0.
- En cas de que no hi hagi cap matèria que compleixi aquesta condició no s'haurà d'escriure el codi del tribunal.
- La seqüència acabarà amb un codi de final de seqüència igual a -1.

Per exemple, suposem que l'entrada està formada per la seqüència següent:

```
<325 1 4 132 8.0 133 5.0 158 7.2 115 8.6
      5 2 133 6.6 158 3.8
      3 3 132 4.4 144 7.0 115 6.4
      0
127  1 1 413 7.6
      3 3 413 6.6 455 5.8 438 7.8
      4 3 413 7.8 455 6.0 438 8.8
      0
248  2 2 288 5.6 224 7.6
      4 4 288 4.2 255 6.8 224 3.8 237 5.0
      5 1 255 6.8
      0
      -1>
```

a la sortida tindríem:



< 325 5 50.0 133 6.6 3 66.6 144 7.0 0 248 4 50.0 255 6.8 0 -1 >



### Exercici 3. Codificació en C [35%]

**Objectius:** Donat un algorisme escrit en pseudocodi, es demanarà la traducció al C.

**Tasca:** Traducció del següent algorisme a llenguatge C.

Al finalitzar cada curs acadèmic, per tal de poder determinar les llistes d'accés a la universitat, es realitzen les PAU (Proves d'Accès a la Universitat), mes conegudes com "La Selectivitat". La realització d'aquestes proves té lloc en diferents centres arreu de Catalunya. En cada un d'aquests centres les proves s'estructuren en tribunals, de manera que cada estudiant té assignat un tribunal en que haurà de fer els exàmens de totes les matèries de que s'ha d'examinar. D'aquesta manera, els resultats de les proves queden també enregistrats en base a aquesta estructura de centres docents i tribunals.

Les qualificacions queden doncs estructurades per cada tribunal en una seqüència amb el format següent:

```
<idChair idSubject1 numEst1 idEst(1)(1) Marks(1)(1) idEst(1)(2) Marks(1)(2) ... idEst(1)(N1)
Marks(1)(N1) idSubject2 numEst2 idEst(2)(1) Marks(2)(1) idEst(2)(2) Marks(2)(2) ... idEst(2)(N2)
Marks(2)(N2) .... idSubjectM numEstM idEst(M)(1) Marks(M)(1) idEst(M)(2) Marks(M)(2) ... idEst(M)(NM)
Marks(M)(NM) 0>
```

on :

- *idChair<sub>i</sub>* és un enter positiu que identifica el tribunal.
- *idSubject<sub>i</sub>* és un enter més gran que zero que codifica la matèria dels resultats que venen a continuació.
- *numEst<sub>i</sub>* és un enter positiu que indica el nombre d'exàmens que hi ha hagut de la matèria *idSubject<sub>i</sub>* en el tribunal *idChair<sub>i</sub>*.
- *idEst<sub>(i)(j)</sub>* és un enter positiu que identifica l'estudiant. Aquest codi és el mateix per a totes les matèries de que s'ha examinat l'estudiant.
- *Marks<sub>(i)(j)</sub>* és un real que indica la nota que ha tret l'estudiant *idEst<sub>(i)(j)</sub>* en la matèria *idSubject<sub>i</sub>*.
- *END* és la marca final de seqüència i val 0.

Anotacions:

- La seqüència contindrà, com a mínim, la identificació del tribunal i la marca final.
- Cada matèria contindrà com a mínim un estudiant.
- No tots els estudiants tenen perquè aparèixer en cada una de les matèries.
- Cada estudiant té un codi únic que l'identifica en totes les matèries.
- Les dades de la seqüència són coherents.





Partint d'una seqüència d'entrada, que es llegirà del canal d'entrada estàndard, amb una estructura com:

<idEst TRIB >

on idEst es un enter corresponent a l'identificador d'un estudiant, i TRIB es una seqüència de notes per a un tribunal com la mostrada mes amunt. Ens demanen que **dissenyem un algorisme que** escrigui pel canal de sortida les notes de les PAU de l'estudiant idEst. A la seqüència de sortida, per a cada una de les matèries, s'ha d'indicar

- El codi de la matèria
- La nota de l'estudiant idEst
- El percentatge que ocupa en aquesta matèria respecte a la millor nota.

Si l'estudiant no ha efectuat l'examen d'una matèria, aquesta apareixerà igualment al llistat però amb una nota igual a -1.0 i un percentatge igual a 0.0.

La seqüència acabarà amb el codi de fi de seqüència 0. Així per exemple, si a l'entrada tinguéssim la seqüència:

```
< 133 325 1 4 132 8.0 133 5.0 158 7.2 115 8.6
          5 2 133 6.6 158 3.8
          3 3 132 4.4 144 7.0 115 6.4
0 >
```

l'algorisme hauria d'escriure:

```
<1 5.00 58.14 5 6.00 100.00 3 -1 0.00 0>
```



```

const
  ENDSEQ : enter = 0;
  NO_SUBJECT : real = -1.0;
  NO_PERCENTAGE : real = 0.0;
fconst

```

**algoritme** marksStudent

```

var
  idStudent, idSubject, idChair : enter;
fvar

  idStudent:=readInteger();
  idChair:=readInteger();
  idSubject:=readInteger();
  mentre (idSubject <> ENDSEQ) fer
    readSubject(idSubject, idStudent);
    idSubject:=readInteger();
  fmentre
  writeInteger(ENDSEQ);
falgoritme

```

**accio** readSubject (**ent** idSubject: **enter**, **ent** idStudent: **enter**)

```

var
  numStudents: enter;
  studentMark, bestMark: real;
  found: boolean;
  k : enter;
fvar

  found:=false;
  bestMark:=0;
  writeInteger(idSubject);
  numStudents:=readInteger();
  per k:=1 fins numStudents fer
    processStudent(idStudent, studentMark, bestMark, found)
  fper

  si found llavors
    writeReal(studentMark);
    writeReal(100.0*studentMark/bestMark);
  sino
    writeReal(NO_SUBJECT);
    writeReal(NO_PERCENTAGE);
  fsi

```



**faccio**

**accio** processStudent(**ent** idStudent: **enter**, **entsor** studentMark: **real**, **entsor** bestMark: **real**, **entsor** found: boolea)

```
var
    id: enter;
    marks: real;
fvar

id:= readInteger();
marks := readReal();
si (id = idStudent) llavors
    found:=true;
    studentMark:= marks;
fsi
si (marks > bestMark) llavors
    bestMark:=marks;
fsi
```

**faccio**