

Semestre 2 - Pràctica 3 LSVirtualClassrooms



Índice

1. Introducció.....	3
2. Objectius.....	4
3. Funcionament.....	5
4. Procediment de lliurament.....	8
4.1 Desenvolupant fent ús d'un TAD	10
4.2 Tests	11
5. Consideracions.....	12
6. Requisits mínims.....	13
7. Data de lliurament i avaluació.....	14

1. Introducció

A causa de l'augment de nombre d'alumnes a La Salle, des de Coordinació Acadèmica se'ns va demanar que desenvolupéssim una eina de suport per poder assignar els alumnes a les aules corresponents segons el grau que volen cursar (Pràctica 2.2). Ara ens comuniquen que necessitarien que féssim alguns canvis en aquest programa.

2. Objectius

Els objectius bàsics d'aquesta pràctica son:

- Entendre el funcionament de punters i memòria dinàmica.
- Entendre els avantatges, inconvenients i limitacions de la memòria dinàmica.
- Aprendre a gestionar correctament la memòria dinàmica.
- Posar en pràctica la utilització de punters per a crear, gestionar i modificar arrays dinàmics.
- Comprendre la diferència entre utilitzar un punter per simular el pas d'arguments per referència i el seu ús per a gestionar la memòria dinàmica.
- Comprendre el funcionament de les estructures de dades lineals i aprendre a utilitzar-les correctament.

3. Funcionament

El procés d'entrega d'aquesta pràctica varia respecte a les anteriors. Si us plau, llegiu la secció "Procediment de lliurament".

El funcionament d'aquest programa serà molt similar al de la pràctica anterior (Pràctica 2.2). Tot i així, ens indiquen que a causa del confinament, les aules s'han mogut a l'espai virtual, amb la qual cosa ja no tindran un límit d'alumnes. Això implica diversos canvis.

- El format del fitxer de **graus i classes** serà lleugerament diferent. Ja no s'indicarà el nombre màxim d'alumnes per classe. El número màxim serà il·limitat pel fet de tractar-se de classes virtuals. A continuació teniu un exemple del nou format de fitxer:

```
1
2 Computer Engineering
JH1.1
.JH1 2
```

- El registre de les classes (Classroom) serà diferent:

```
typedef struct {
    char name[MAX_STRING_LENGTH];
    int current_capacity;
    LinkedList students;
} Classroom;
```

- Pel que fa a la càrrega, emmagatzematge i assignació d'estudiants a les classes d'un grau, ara s'assignaran tots els estudiants d'aquest grau a una mateixa classe (a la primera que apareix en el fitxer). Serà tasca de l'usuari del programa moure els estudiants entre classes, si és necessari, fent servir l'opció 3 del menú. L'ordre d'emmagatzematge dels estudiants serà per ordre d'aparició. És a dir, si al fitxer apareix primer l'estudiant A i després el B, el primer element de la llista serà l'A i el segon el B.
- Tant a l'opció 1 (Summary) com a l'opció 3 (Move student), la capacitat màxima de la classe queda substituïda per la cadena "inf". Vegeu l'Output 1 i l'Output 2 respectivament.
- Per a l'opció 3, quan es mouen estudiants d'una classe, desapareix l'error produït quan l'estudiant no cap a una classe.

```
Welcome!

Type the name of the 'classrooms' file: class_1

Type the name of the 'students' file: stus_1

1. Summary | 2. Show degree students | 3. Move student | 4. Exit
Select option: 1

Computer Engineering
JH1.1 2/inf
JH1.2 0/inf

Multimedia Engineering
JH1.3 3/inf
JH1.4 0/inf

Telecommunications Systems Engineering
Measurements 3/inf

Telematics
CCNA 1/inf

Audiovisual Systems Engineering
JH1.5 1/inf

Electronic Engineering
JH2.1 3/inf
JH2.2 0/inf

Engineering in ICT Management
JH2.3 3/inf

1. Summary | 2. Show degree students | 3. Move student | 4. Exit
Select option: █
```

Output 1. A l'opció 1, la capacitat màxima de les classes és infinita. Es mostra "inf".

```
Welcome!

Type the name of the 'classrooms' file: class_1

Type the name of the 'students' file: stus_1

1. Summary | 2. Show degree students | 3. Move student | 4. Exit
Select option: 3

Degree? Computer Engineering

Classrooms and capacity:
1. JH1.1 2/inf
frostmourne
silvi
2. JH1.2 0/inf

Who do you want to move (login)?
```

Output 2. A l'opció 3, la capacitat màxima de les classes és infinita. Es mostra "inf".

4. Procediment de lliurament

El lliurament d'aquesta pràctica s'haurà de realitzar tenint en compte alguns canvis respecte les pràctiques anteriors.

Com sabeu, les pràctiques s'han d'implementar en el servidor Matagalls, compilant amb GCC. En canvi, per a que es puguin realitzar les proves d'execució a CodeRunner, haureu de canviar el nom de la vostra funció **main** per **myMain**. D'aquesta manera la capçalera de la vostra funció quedaria:

```
int myMain ()
```

Observeu la Imatge 1 per a tenir un exemple:

```
1 // Code
2
3 // Rewrite your "main" to "myMain"
4 int myMain() {
5     // Code
6 }
7
8 /**
9  * There should NOT be an "int main() { ... }" here
10 *
11 */
12
```

Imatge 1. Reanomenem a la funció main, myMain

Recordeu que aquest canvi només s'ha de realitzar quan vulgueu executar els tests de CodeRunner. Per poder compilar satisfactòriament el vostre codi quan estigueu treballant a Matagalls haureu d'usar un *main* estàndard com heu fet fins ara.

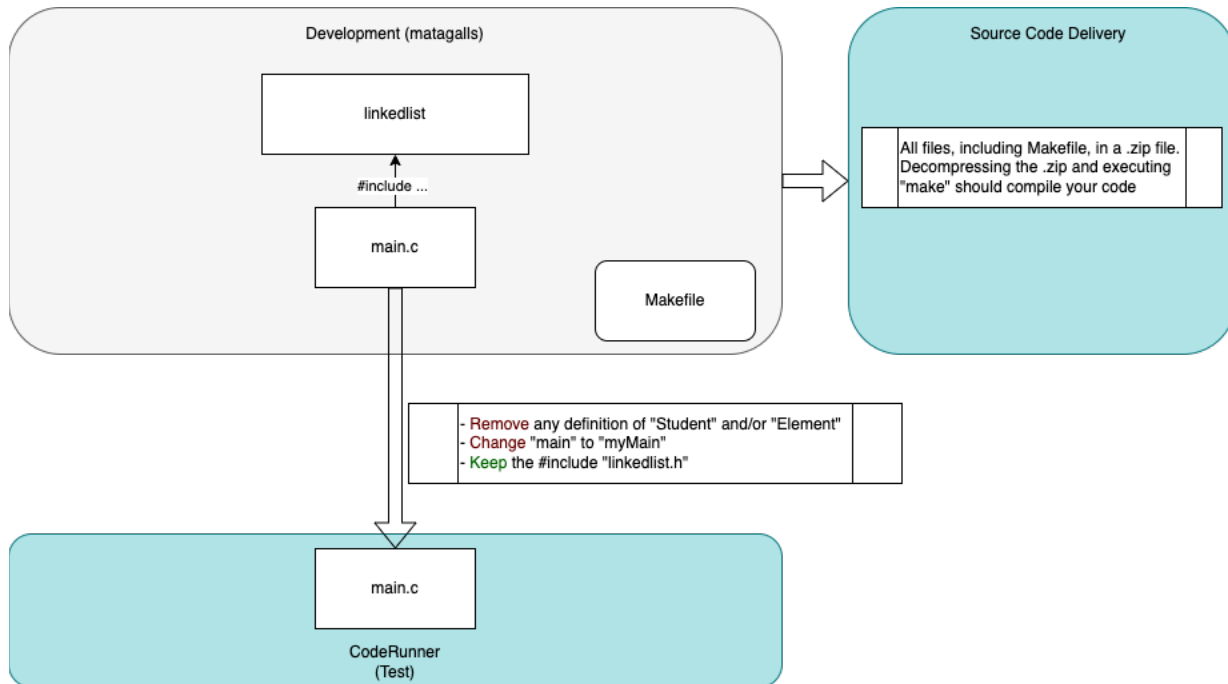
Els tests de CodeRunner utilitzen el tipus "Student" i el tipus "Element" genèric vist a classe. És molt important que utilitzeu exactament aquests tipus al vostre programa i a l'estructura de dades lineal que utilitzeu. Quan intenteu passar els tests, com veureu a continuació, no heu d'incloure ni el TAD utilitzat ni la definició dels tipus "Student" i "Element". El tipus "Student" i la seva equivalència amb "Element" ja es troben inclosos en l'entorn d'execució CodeRunner. El TAD de la LinkedList a utilitzar és el que teniu disponible a l'eStudy:

<https://estudy.salle.url.edu/mod/folder/view.php?id=761870>

Tanmateix, a la vostra pràctica sí que heu de mantenir l'include a l'estructura de dades lineal `linkedlist.h`.

4.1 Desenvolupant fent ús d'un TAD

S'ha de desenvolupar la pràctica a matagalls fent servir tres fitxers i un makefile que els compili: el main.c amb la vostra pràctica i els dos corresponents al TAD de l'Estructura de Dades Lineal que aneu a utilitzar (.c i .h). Haureu de lliurar un fitxer .zip amb els diferents fitxers usats al pou de lliurament del codi font. Vegeu la Imatge 3:



Imatge 3.

4.2 Tests

Tal i com s'indica en aquest enunciat, en aquesta pràctica s'haurà d'usar memòria dinàmica per a crear certes estructures de dades. Es per això, que en l'execució dels tests de CodeRunner es comprovarà que això s'ha realitzat satisfactòriament. Concretament, si no es compleixen certes condicions, en els tests apareixerà, a la columna "Got", un missatge d'error al final de la sortida del programa. Es distingeixen els següents casos:

- "ERROR: Mismatch size": Aquest error apareixerà si s'ha demanat menys memòria de la necessària per a crear les estructures dinàmiques descrites a l'enunciat.
- "ERROR: Some pointer exceeded its allocated memory": Aquest error apareixerà si s'ha accedit a alguna direcció de memòria que no ha sigut reservada pel programa.
- "ERROR: Some pointers have not been deallocated" Aquest error apareixerà si no s'ha alliberat tota la memòria dinàmica en finalitzar el programa.
- També apareixerà un error si s'intenta alliberar memòria dinàmica ja alliberada (doble *free*).

5. Consideracions

Per a la implementació d'aquesta pràctica heu de tenir en compte les següents consideracions:

1. S'ha de seguir el format mostrat en els exemples.
2. La forma de gestionar els errors es estrictament la mostrada en l'enunciat.
3. El programa s'ha d'estructurar correctament en procediments i funcions. No s'acceptarà un programa amb codi mal estructurat.
4. Podeu suposar que els fitxers no estaran buits i seguiran el format específic, tot i que s'han de controlar els errors indicats i els possibles errors d'obertura.
5. Disposareu dels fitxers d'exemple usats a les imatges d'aquest enunciat per comprovar el funcionament del programa.
6. Es obligatori utilitzar únicament estructures de dades equivalents a les llistades en aquest enunciat (Degrees, Degree, Classroom, Student).

6. Requisits mínims

Aquesta pràctica ha de complir amb una sèrie de requisits de qualitat mínims per poder ser avaluada:

- El codi ha d'estar correctament comentat de manera que sigui llegible sense dificultats.
- Ha de seguir la guia d'estils de l'assignatura.
- Per poder aprovar es necessari que la pràctica superi satisfactòriament els tests de Coderunner.
- El programa hauria de desenvolupar-se íntegrament en l'entorn Linux de Matagalls, utilitzant l'editor Vim per escriure el codi C i el compilador gcc per a l'obtenció de l'executable corresponent.
- No es pot utilitzar cap eina ni instrucció de C que no s'hagi explicat a classe.
- El codi ha d'estar estructurat correctament utilitzant **un mínim de 6 procediments o funcions** significatius/ves. En aquest sentit, es valorarà especialment que en el codi **no es repeteixin conjunts de sentències**.
- No s'acceptarà cap pràctica que no compleixi amb la normativa de pràctiques.

7. Data de lliurament i avaluació

La data de lliurament d'aquesta pràctica per poder obtenir la màxima qualificació és el 15 de maig de 2022.

Per a que aquesta pràctica es consideri **lliurada** s'hauran de complir les següents condicions:

1. Executar el codi en el CodeRunner habilitat expressament i superar satisfactòriament els tests.
2. Lliurar al pou corresponent un fitxer .zip que contingui els diferents arxius que conformen la pràctica.
3. Lliurar una memòria final de les pràctiques d'aquest semestre, incloent:
 - a. Portada i índex
 - b. Breu resum de l'enunciat de cada pràctica
 - c. Estructuració de les pràctiques 2.1, 2.2 i 2.3
 - d. Principals dificultats i solucions aplicades
 - e. Conclusions finals
 - f. Estimació del temps dedicat per realitzar cada pràctica

Recordeu que la pràctica serà avaluada de la següent manera:

- Execució: té un pes del 80% i avalua el correcte funcionament de la pràctica. Serà la qualificació obtinguda en els tests de CodeRunner.
- Qualitat del SW: té un pes del 20% i s'avaluarà la qualitat del codi lliurat i el seguiment de la Guia d'Estils de Programació que teniu disponible a l'eStudy.

Important: El fet de superar únicament els tests no implica que la pràctica s'hagi aprovat.