

Actividad Evaluada: Segunda Práctica de Laboratorio

Leeremos los valores de un *problem.msh* para llevar nuestra malla desde la matemática hacia el código.

Insumos:

En el repositorio, se le proporciona un archivo llamado *activity_problem.msh* con el siguiente contenido:

```
0.2 5
10 9 1 1

Coordinates
1 0.1
2 0.4
3 0.9
4 1.4
5 1.5
6 1.8
7 2.3
8 2.6
9 2.7
10 3.3
EndCoordinates
```

Deberá descargar y renombrar ese archivo a *problem.msh* para utilizarlo como insumo de esta actividad.

Del mismo modo, se le proporciona un archivo *main.cpp* el cual se detalla a continuación:

```
#include <fstream>
#include <iostream>

using namespace std;

void readCoordinates(ifstream &file, int n, item* item_list){
    ANSWER e; ANSWER r;
    for(int i=0; i<n; i++){
        file >> e >> r;
        item_list[i].METHOD_NAME(e,r);
    }
}
```

Observe que se han incluido macros no declaradas como ANSWER.

Deberá sustituir esas macros por código de manera que le permita resolver esta actividad.

Empezamos declarando un arreglo que servirá para leer el nombre del archivo, en este caso, nuestro problem.msh

```
int main() {  
    char filename[10];
```

El valor de 10 es la longitud de caracteres que tiene el nombre del archivo.

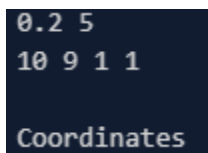
Para asegurarnos de validar si hemos escrito mal el nombre del archivo, leeremos el nombre del archivo hasta que se introduzca el nombre correcto.

```
do{  
    cout << "Ingrese el nombre del archivo: ";  
    cin >> filename;  
    file.open(filename);  
}while(!file);
```

Declaramos algunas variables que nos servirán para almacenar los valores que se lean desde el archivo.

```
char filename[10];  
string line;  
mesh m;  
ifstream file;  
float k,Q;  
int nnodes,neltos,ndirich,nneu;
```

Recordemos un poco la estructura del problem.msh



```
0.2 5  
10 9 1 1  
Coordinates
```

El primer dato representa la conductividad térmica k.

Leemos la primera línea del archivo.

```
file >> k >> Q;
```

Los valores se almacenarán en las variables correspondientes.

La segunda fila del problem.msh representa la cantidad de nodos, elementos y condiciones.

```
file >> nnodes >> neltos >> ndirich >> nneu;
```

Las siguientes dos líneas:

```
file >> line;  
file >> line;
```

Se utilizan para que no almacenemos la siguiente información:

Coordinates

Un espacio en blanco y una cadena de texto, solo nos sirven como referencia para facilitar la lectura del archivo.

Se tiene además una función para almacenar los valores de las coordenadas:

```
void readCoordinates(ifstream &file, int n, item* item_list){  
    ANSWER e; ANSWER r;  
    for(int i=0; i<n; i++){  
        file >> e >> r;  
        item_list[i].METHOD_NAME(e,r);  
    }  
}
```

La cual deberán modificar respectivamente.

Observar el orden de ejecución de estas líneas:

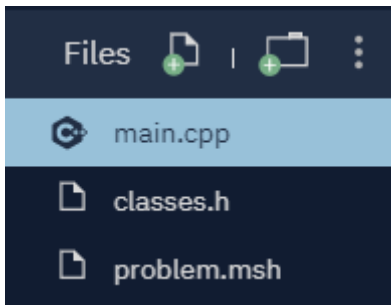
```
m.setParameters(ANSWER, ANSWER);  
m.setSizes(ANSWER, ANSWER, ANSWER, ANSWER);  
m.createData();  
  
readCoordinates(ANSWER,ANSWER,m.getNodes());
```

Se tiene que crear primero la mesh de manera que podamos añadir los valores de los elementos, nodos y condiciones después, ya que necesitamos definir el tamaño de estos:

```
void createData(){  
    node_list = new node[sizes[NODES]];  
    element_list = new element[sizes[ELEMENTS]];  
    dirichlet_list = new condition[sizes[DIRICHLET]];  
    neumann_list = new condition[sizes[NEUMANN]];  
}
```

Consideraciones:

- La estructura del proyecto deberá ser la siguiente:

**Resultado Esperado:**

Dado el `problem.msh`, deberá leer cada valor y almacenarlo en un objeto tipo *mesh*, luego, deberá mostrar los valores de las coordenadas almacenadas, desde este objeto:

```
Ingrese el nombre del archivo: problem.msh
0.1
0.4
0.9
1.4
1.5
```

- Para mostrar cada valor, utilice los métodos, tanto de la clase *mesh*, como de la clase *ítem*.
- El `problem.msh` solo es un insumo, los instructores pueden cambiar los valores para probar el código, pero siempre respetando la estructura de ese archivo.

Entrega:

Deberá subir el código a un repositorio y entregar el enlace de ese repositorio.

Enlace del Repositorio: <https://github.com/Sortweste/-TSC-Laboratorio2>