COMPUTACION

Tema 3 - 1er Final -Fila 2

Disponible desde: miércoles, 2 de diciembre de 2020, 11:50
Límite de entrega: miércoles, 2 de diciembre de 2020, 12:50
Ficheros requeridos: t3F1F2.cpp (Descargar)
Tipo de trabajo: Individual

Universidad Nacional de Asunción - Facultad de Ingeniería

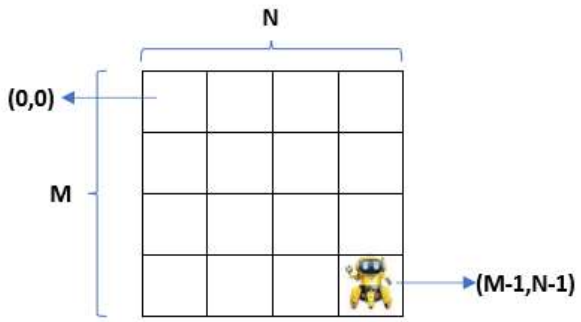
CATEDRA DE COMPUTACIÓN - EXAMEN FINAL 1 - 02/dic/2020 - TEMA 3

Fila 2

Imagine un robot que camina alrededor de una habitación bajo el control de un programa en C++. Las posiciones de la habitación se representan mediante una matriz de `M` filas y `N` columnas, y el robot siempre comienza sus movimientos desde la posición (`M-1,N-1`). Debemos escribir una función `void verificacionDeMovimientosRobot()` que consta de diferentes partes:

Parte 1:

Los valores `M` y `N` son ingresados por teclado, y se debe validar que los mismos sean enteros y positivos. Esto se hará con la ayuda de la función `int ingresarDimension()` (para ingresar cada dimensión), que deberá implementarse.



Parte 2:

Además, también se ingresa por teclado una serie de posibles movimientos del robot (obs: se ingresa un movimiento a la vez):

- `0`: un paso hacia arriba
- `1`: un paso hacia la derecha
- `2`: un paso hacia abajo
- `3`: un paso hacia la izquierda

El programa termina cuando se ingresa el número `4`. Se debe validar que los números que representan a los movimientos estén en el conjunto `{0,1,2,3,4}`. Esto se hará con la ayuda de la función `int ingresarMovimiento()` (para ingresar cada movimiento), que deberá implementarse.

Parte 3:

Si algún movimiento hace que el robot salga de la matriz de posiciones en la habitación, se debe indicar con un mensaje en pantalla. Se proporciona una función auxiliar `void imprimirMensajeError()` para dicho fin.

En cambio, si todos los movimientos hacen que el robot se mantenga dentro de los límites de la matriz, se debe indicar en pantalla su posición final. Para la verificación de que una cierta posición esté o no dentro de la matriz, se deberá implementar y emplear la función `bool verificarPosicion(int posF, int posC, int M, int N)`, que devuelve `true` si la posición `[posF][posC]` está dentro de los límites de la matriz de `M` filas y `N` columnas (y `false` en caso contrario). Se proporciona una función auxiliar `void imprimirPosicionFinal(int i, int j)` para imprimir en pantalla la posición final del robot.

Además (en el caso de que el robot no salga de la matriz), se debe imprimir una matriz que indique la cantidad de veces que el robot estuvo en cada posición de la habitación durante su recorrido. Se proporciona una función auxiliar `void imprimirMatriz(int *A, int m, int n)` para la impresión de la matriz calculada. El puntero `*A` contiene la dirección del elemento inicial `[0][0]` de la matriz.

Por ejemplo: si `M=N=4`, y la secuencia de movimientos del robot es `(0,3,3,0,1,2,2,4)`, entonces los resultados mostrados en pantalla serían:

```
Posición final del robot: (3,2)

Matriz
```

0	0	0	0
0	1	1	0
0	1	2	1
0	0	1	1

Nota: es sólo una representación esquemática. Para la impresión, debe utilizar las funciones proporcionadas

OTRAS DIRECTIVAS A SER NECESARIAMENTE TENIDAS EN CUENTA:

- **NO MODIFICAR LA FUNCIÓN PRINCIPAL** (`main`). **LA MODIFICACIÓN SERÁ PENALIZADA**, pues puede alterarse la secuencia de lecturas en la entrada y la salida esperada para el programa.
- **DEFINIR NECESARIAMENTE LA FUNCIÓN QUE REALICE LO SOLICITADO**. No se aceptará que la funcionalidad se desarrolle en el `main`.
- **LA SOLUCIÓN DEBE SER DEBE SER LO SUFICIENTEMENTE GENERAL para resolver cualquier entrada diferente a las propuestas en los ejemplos y los casos de prueba. Por lo tanto, la calificación mostrada por el sistema no necesariamente corresponderá con la calificación final.**
- **La distribución de puntajes para cada función es la siguiente:**
  - Función `ingresarDimension`: **10%**
  - Función `ingresarMovimiento`: **20%**
  - Función `verificarPosicion`: **20%**
  - Función `verificacionDeMovimientosRobot`: **50%**

Ficheros requeridos

t3F1F2.cpp

```cpp
#include <iostream>
using namespace std;
void imprimirMatriz(int *A, int m, int n){
    int i,j;
    for(i=0;i<m;i++) {
        for(j=0;j<n;j++) {
            cout<<*(A+i*n+j)<<" ";
            }
        cout<<endl;
    }
}
void imprimirPosicionFinal(int i, int j){
    cout<<"("<<i<<","<<j<<")"<<endl;
}
void imprimirMensajeError(){
    cout<<"El robot salio de la matriz!";
}

int ingresarDimension(){
 //Completar desde aqui

}


int ingresarMovimiento(){
    //Completar desde aqui

}
bool verificarPosicion(int posF,int posC, int M,int  N){
    //Completar desde aqui

}

void  verificacionDeMovimientosRobot(){
    int M = ingresarDimension();
    int N = ingresarDimension();
    int A[M][N]; //Matriz de conteo para cada posicion
    int i,j; //indices
    for(i=0;i<M;i++) for(j=0;j<N;j++) A[i][j]=0; //inicializacion
    //Completar desde aqui

}


int main(){
    /*---NO MODIFICAR El MAIN, se penaliza*/

    int opcion;
    cin>>opcion;
    if(opcion==1){
        cout<<ingresarDimension()<<endl;
    }
    if(opcion==2){
        cout<<ingresarMovimiento()<<endl;
    }
    if(opcion==3){
        int posF,posC,M,N;
        cin>>posF;
        cin>>posC;
        cin>>M;
        cin>>N;
        cout<<verificarPosicion(posF,posC,M,N)<<endl;
    }
    if(opcion==4){
        verificacionDeMovimientosRobot();
    }

    return 0;
}
```

VPL

◄ Final 1 - Tema 2 - Fila 2

Ir a...

Programa de estudio semana 1 ►