

Tecnológico Nacional de México Campus Querétaro.

Laberinto.

Alumno:

Arellano Ochoa Daniel Ignacio

Carrera:

Ingeniera en sistemas computacionales

Docente:

Felipe Estrada Rojas

Fecha: 04/10/2020

Código: El archivo cpp se anexo junto con este trabajo, de igual manera el código se encuentra hasta el final de este trabajo.

Ejecución:

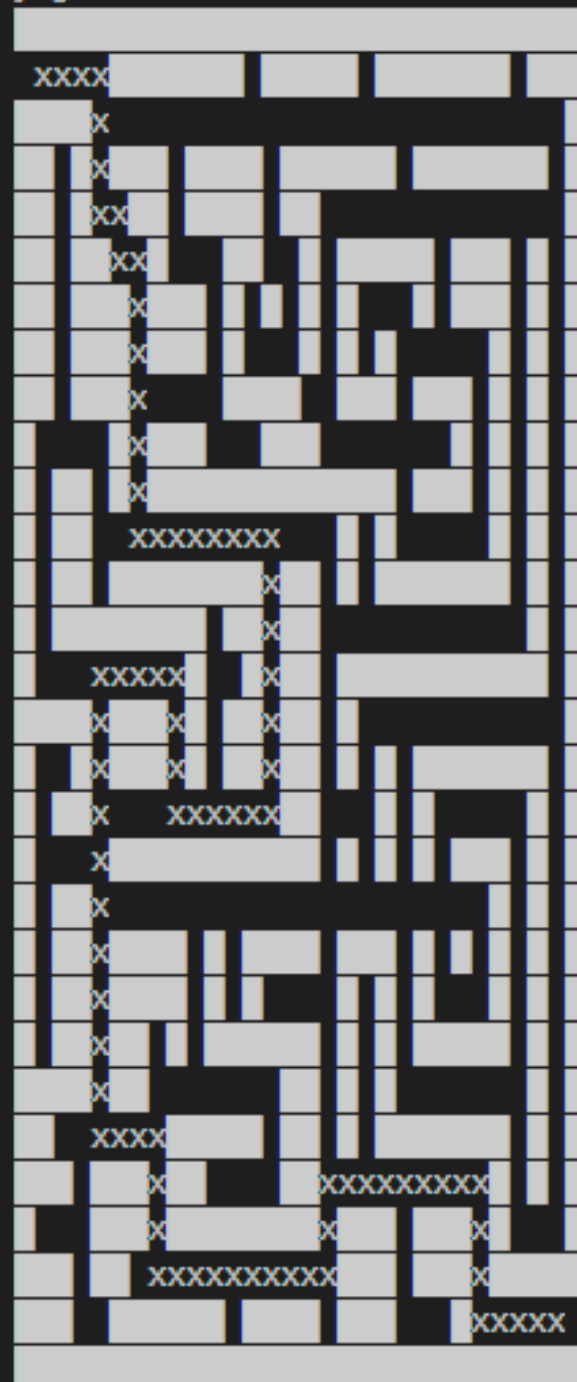




Entrada:

x/i= 1

y/j= 0



Salida:

x/i= 28

y/j= 29

PS C:\Users\Daniel\Desktop\Laberinto>

Conclusión:

El problema del laberinto resulta ser complicado de solucionar de manera no recursiva (lo intente), obteniendo solamente como resultado terminar en una posición atrapado entre muros, es por eso que el realizarlo recursivamente da la oportunidad de regresar a una posición donde se tomó un camino para después eliminar esta opción y continuar con otras opciones, con este ejercicio se logra concluir que en ocasiones la recursividad es la única o de las únicas soluciones más efectivas y cortar para la solución de un problema pues como se lograra observar en el código, realmente el código para solucionar el laberinto es más corto de lo que uno pensaría siendo la verdadera dificultad de este problema identificar los casos base, recursivos y en particular para mi encontrar la manera de marcar el camino seguido para encontrar la salida, con esto último también se logra observar que la solución no es la más óptima pero logra solucionar que al fin de cuentas es lo que se busca en un laberinto real.

Código:

```
/*Nombre: Arellano Ochoa Daniel Ignacio
No. de control: 19141118
Programa que permite solucionar un laberinto
*/
#include <string>
#include <iostream>

using namespace std;
//estructura para generar una matriz de 30x30 con el laberinto
struct Laberinto{
    //M muro, E entrada, S salida, ' ' camino
    char laberinto[30][30]={
        'M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M',
        'M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M',
        'E',' ',' ',' ',' ',' ','M','M','M','M','M','M','M',' ','M','M','M','M',
        'M',' ','M','M','M','M','M','M','M',' ','M','M','M',
        'M','M','M','M',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',
        ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','M',
        'M','M',' ',' ','M',' ','M','M','M',' ','M','M','M','M',' ','M','M','M',
        'M','M','M',' ','M','M','M','M','M','M','M',' ','M',
        'M','M',' ','M',' ',' ','M','M',' ','M','M','M','M',' ','M','M',' ',
        ' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ',' ','M',
        'M','M',' ','M','M',' ',' ','M',' ',' ','M','M',' ',' ','M',' ',
        'M','M','M','M','M',' ','M','M','M',' ','M',' ','M',
        'M','M',' ','M','M','M',' ','M','M','M',' ','M',' ','M',' ','M',' ',
        'M',' ',' ',' ','M',' ','M','M','M',' ','M',' ','M',
    }
```

[illegible]

```

        'M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M','M',
'M','M','M','M','M','M','M','M','M','M','M','M','M','M','M',

    };

}lab;

//Codigo para imprimir el laberinto de 30x30
void imprimirLaberinto(Laberinto *lab){
    for (size_t i = 0; i < 30; i++)
    {
        for (size_t j = 0; j < 30; j++)
        {
            if (lab->laberinto[i][j]=='M')
                printf("%c",219);
            else if(lab->laberinto[i][j]=='x')
                printf("x");
            else
                printf(" ");
        }
        printf("\n");
    }
}

//codigo recursivo para solucionar el laberinto de 30x30
bool solucionarLaberinto(int x, int y, int primer, Laberinto *lab){
    char* aux;
    if(x<0 || x>29 || y<0 || y>29)
        return 0;
    else
    {
        aux=&lab->laberinto[x][y];

        if (*aux=='S')
            return true;
        else if(*aux=='E')
        {
            if(primer==1)
            {
                if(solucionarLaberinto(x+1, y, 0, lab)||solucionarLaberinto(
x-
1, y, 0, lab)||solucionarLaberinto(x, y+1, 0, lab)||solucionarLaberinto(x, y
-1, 0, lab))
                    return true;
            }
        }
    }
}

```

```

        else
            return false;
    }
    else if(*aux==' ')
    {
        *aux='x';
        if(solucionarLaberinto(x+1, y, 0, lab)||solucionarLaberinto(x-
1, y, 0, lab)||solucionarLaberinto(x, y+1, 0, lab)||solucionarLaberinto(x, y
-1, 0, lab))
            return true;
        else {
            *aux=' ';
            return false;
        }
    }
    else
        return false;
}

}

//Codigo para encontrar la entrada 'E' del laberinto de 30x30, para despues
solucionarlo
void encontrarEntrada(Laberinto *lab) {
    int entrada=0;
    int x,y,x1,y1;
    for (size_t i = 0; i < 30; i++)
    {
        for (size_t j = 0; j < 30; j++)
        {
            if(lab->laberinto[i][j]=='E'){
                x=i;
                y=j;
                entrada=1;
            }
            if(lab->laberinto[i][j]=='S'){
                x1=i;
                y1=j;
            }
        }
    }

}

cout<<"Entrada:\nx/i= "<<x<<"\ny/j= "<<y<<endl;
bool res=solucionarLaberinto(x, y, entrada, lab);
if (res==true){

```

```
        imprimirLaberinto(lab);
        cout<<"Salida:\nx/i= "<<x1<<"\ny/j= "<<y1<<endl;
    }

    else
        cout<<"Sin solucion"<<endl;
}

//Menu de inicio del programa
int main(int argc, const char * argv[]){
    Laberinto* auxLaberinto=&lab;
    imprimirLaberinto(auxLaberinto);
    encontrarEntrada(auxLaberinto);
}
```