***func.c***

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "robo.h"

char \*\*criar\_matriz(int l, int c)

{

char \*\*ret=NULL;

int i=0;

// (ret recebe o endereÃ§o de um ponteiro pra ponteiro, reserve x de memoria.)

if(!(ret=(char\*\*)malloc(sizeof(char\*)\*l))){

return NULL;}

for (i=0; i<l; i++){

// ret recebe um ponteiro que vai apontar para X de memÃ³ria

ret[i]=(char\*)malloc(sizeof(char)\*c);

if(!ret[i]){//if(ret[i]==NULL)

return NULL;

}

}

return ret;

}

void liberar\_matriz(char \*\*m, int l, int c){

int i;

for (i=0; i<l; i++){

free(m[i]);

}

free(m);

}

void printar\_relatorio(FILE \*relatorio, rodada r, int quantidade)

{

fprintf(relatorio,"Caso: %d\n",quantidade);

fprintf(relatorio,"Bolinhas coletadas = %d\nSaltos sobre cones =%d\n",r.pontos,r.contcones);

fprintf(relatorio,"Andou para frente:\nN: %d ", r.contadores[0]);

fprintf(relatorio,"NE: %d ", r.contadores[1]);

fprintf(relatorio,"L: %d ", r.contadores[2]);

fprintf(relatorio,"SE: %d ", r.contadores[3]);

fprintf(relatorio,"S: %d " , r.contadores[4]);

fprintf(relatorio,"SO: %d ", r.contadores[5]);

fprintf(relatorio,"O: %d " , r.contadores[6]);

fprintf(relatorio," NO: %d\n" , r.contadores[7]);

}

void mover\_frente\_s(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha+1][\*coluna])!='#')

{

r->contadores[4]++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+1][\*coluna])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)+1][\*coluna]='r';

(\*linha)++;

}

}

void mover\_frente\_se(rodada \*r, int \*linha,int \*coluna, char \*sentido)

{

if((r->arena[\*linha+1][\*coluna+1])=='#'){

(r->contadores[3])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+1][\*coluna+1])=='\*'){

(r->pontos)++;

}

r->arena[(\*linha)+1][\*coluna+1]='r';

(\*linha)++;

(\*coluna)++;

}

}

void mover\_frente\_l(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha][\*coluna+1])!='#')

{

(r->contadores[2])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)][\*coluna+1])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)][\*coluna+1]='r';

(\*coluna)++;

}

}

void mover\_frente\_ne(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha-1][\*coluna+1])!='#'){

(r->contadores[1])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-1][\*coluna+1])=='\*'){

(r->pontos)++;

}

r->arena[(\*linha)-1][\*coluna+1]='r';

(\*linha)--;

(\*coluna)++;

}

}

void mover\_frente\_n(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha-1][\*coluna])!='#')

{

(r->contadores[0])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-1][\*coluna])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)-1][\*coluna]='r';

(\*linha)--;

}

}

void mover\_frente\_no(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha-1][\*coluna-1])!='#')

{

(r->contadores[7])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-1][\*coluna-1])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)-1][\*coluna-1]='r';

(\*linha)--;

(\*coluna)--;

}

}

void mover\_frente\_o(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha][\*coluna-1])!='#')

{

(r->contadores[6])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)][\*coluna-1])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)][\*coluna-1]='r';

(\*coluna)--;

}

}

void mover\_frente\_so(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if((r->arena[\*linha+1][\*coluna-1])!='#')

{

(r->contadores[5])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+1][\*coluna-1])=='\*')

{

(r->pontos)++;

}

r->arena[(\*linha)+1][\*coluna-1]='r';

(\*linha)++;

(\*coluna)--;

}

}

void saltar\_s(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*linha!=r->linha-2 && \*linha!=r->linha-1)

{

if((r->arena[\*linha+2][\*coluna])!='#'){

(r->contadores[4])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+2][\*coluna])=='\*')

(r->pontos)++;

r->arena[(\*linha)+2][\*coluna]='r';

if(r->arena[\*linha+1][\*coluna]=='#')

{

(r->contcones)++;

}

\*linha=\*linha+2;

}

}

}

void saltar\_se(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=r->coluna-1 && \*coluna!=(r->coluna)-2 && \*linha!=r->linha-1 &&\*linha!=(r->linha)-2)

{

if((r->arena[\*linha+2][\*coluna+2])!='#')

{

(r->contadores[3])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+2][\*coluna+2])=='\*')

(r->pontos)++;

r->arena[(\*linha)+2][\*coluna+2]='r';

if(r->arena[\*linha+1][\*coluna+1]=='#')

(r->contcones)++;

\*linha=\*linha+2;

\*coluna=\*coluna+2;

}

}

}

void saltar\_l(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=r->coluna-2 && \*coluna!=(r->coluna)-1)

{

if((r->arena[\*linha][\*coluna+2])!='#')

{

(r->contadores[2])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)][\*coluna+2])=='\*')

(r->pontos)++;

r->arena[(\*linha)][\*coluna+2]='r';

if(r->arena[\*linha][\*coluna+1]=='#')(r->contcones)++;

\*coluna=\*coluna+2;

}

}

}

void saltar\_ne(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=r->coluna-2 && \*coluna!=(r->coluna)-1 && \*linha!=0 && \*linha!=1)

{

if((r->arena[\*linha-2][\*coluna+2])!='#')

{

(r->contadores[1])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-2][\*coluna+2])=='\*')

(r->pontos)++;

r->arena[(\*linha)-2][\*coluna+2]='r';

if(r->arena[\*linha-1][\*coluna+1]=='#')

(r->contcones)++;

\*linha=\*linha-2;

\*coluna=\*coluna+2;

}

}

}

void saltar\_n(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*linha!=0 && \*linha!=1)

{

if((r->arena[\*linha-2][\*coluna])!='#')

{

(r->contadores[0])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-2][\*coluna])=='\*')

(r->pontos)++;

r->arena[(\*linha)-2][\*coluna]='r';

if(r->arena[\*linha-1][\*coluna]=='#')

(r->contcones)++;

\*linha=\*linha-2;

}

}

}

void saltar\_no(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=0 && \*coluna!=1 && \*linha!=0 && \*linha!=1)

{

if((r->arena[\*linha-2][\*coluna-2])!='#')

{

(r->contadores[7])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)-2][\*coluna-2])=='\*')

(r->pontos)++;

r->arena[(\*linha)-2][\*coluna-2]='r';

if(r->arena[\*linha-1][\*coluna-1]=='#')

(r->contcones)++;

\*linha=\*linha-2;

\*coluna=\*coluna-2;

}

}

}

void saltar\_o(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=0 && \*coluna!=1)

{

if((r->arena[\*linha][\*coluna-2])!='#')

{

(r->contadores[6])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)][\*coluna-2])=='\*')

(r->pontos)++;

r->arena[(\*linha)][\*coluna-2]='r';

if(r->arena[\*linha][\*coluna-1]=='#')

(r->contcones)++;

\*coluna=\*coluna-2;

}

}

}

void saltar\_so(rodada \*r, int \*linha, int \*coluna, char \*sentido)

{

if(\*coluna!=0 && \*coluna!=1 && \*linha!=r->linha-2 && \*linha!=(r->linha)-1)

{

if((r->arena[\*linha+2][\*coluna-2])!='#')

{

(r->contadores[5])++;

r->arena[\*linha][\*coluna]='.';

if((r->arena[(\*linha)+2][\*coluna-2])=='\*')

(r->pontos)++;r->arena[(\*linha)+2][\*coluna-2]='r';

if(r->arena[\*linha+1][\*coluna-1]=='#')

(r->contcones)++;

\*linha=\*linha+2;

\*coluna=\*coluna-2;

}

}

}

robo.h

#ifndef ROBO\_H

#define ROBO\_H

#define DIRECOES 8

#include <stdio.h>

typedef struct {

int linha;

int coluna;

int comando;

char \*\*arena;

int contcones;

int pontos;

int contadores[DIRECOES];

}rodada;

/\* funcoes para manipular a matriz \*/

char \*\*criar\_matriz(int linhas, int colunas);

void liberar\_matriz(char \*\*m,int l, int c);

void printar\_relatorio(FILE \*relatorio, rodada r, int quantidade);

void mover\_frente\_s(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_l(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_ne(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_n(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_no(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_o(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_so(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void mover\_frente\_se(rodada \*r, int \*linha,int \*coluna, char \*sentido);

void saltar\_s(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_l(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_ne(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_n(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_no(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_o(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_so(rodada \*r, int \*linha, int \*coluna, char \*sentido);

void saltar\_se(rodada \*r, int \*linha, int \*coluna, char \*sentido);

#endif

main.c

#include <stdio.h>

#include <string.h>

#include "robo.h"

int main(void)

{

int linha = 0;

int coluna = 0;

int quantidade = 0;

char sentido[2]; //N NE L SE S SO O NO

FILE \*relatorio;

relatorio = fopen("relatorio.txt","a");

rodada r;

//("digite quantas linhas e colunas tera o mapa, assim como a quantidade demovimentos");

while(1)

{

scanf("%d %d %d", &r.linha, &r.coluna, &r.comando);

char fazer[r.comando];

if(r.linha==0 || r.coluna==0 || r.comando==0)

{

break;

}

memset(r.contadores, 0, sizeof(int) \* DIRECOES);

r.pontos=0;

r.contcones=0;

r.arena=criar\_matriz(r.linha,r.coluna);

//("digite a posicao inicial do robo e pra onde ele esta virado");

scanf("%d %d %s", &linha, &coluna, sentido);

linha--;

coluna--;

int i=0;

//printf("digite o mapa");

for(i=0;i<r.linha;i++)

{

scanf("%s", r.arena[i]);

}

r.arena[linha][coluna]='r';

scanf("%s", fazer);

for(i=0;i<r.comando;i++)

{

//("digite o comando, D para virar para direita, E para virar pra esquerda, F paraandar pra frente e S para saltar");

if(fazer[i] == 'F') /\*andar pra frente \*/

{

if(strcmp(sentido, "S")==0)

{

if(linha+1!=r.linha)

mover\_frente\_s(&r, &linha, &coluna, sentido);

}

else if(strcmp(sentido, "SE")==0)

{

if(linha+1!=r.linha && coluna+1 !=r.coluna)

mover\_frente\_se(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "L")==0)

{

if(coluna+1!=r.coluna)

mover\_frente\_l(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "NE")==0)

{

if(linha!=0 && coluna+1!=r.coluna)

mover\_frente\_ne(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "N")==0)

{

if(linha!=0)

mover\_frente\_n(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "NO")==0)

{

if(linha!=0 && coluna!=0)

mover\_frente\_no(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "O")==0)

{

if(coluna!=0)

mover\_frente\_o(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "SO")==0)

{

if(linha+1!=r.linha && coluna!=0)

mover\_frente\_so(&r,&linha,&coluna, sentido);

}

}

if( fazer[i] == 'S')/\*saltar\*/

{

if(strcmp(sentido, "S")==0)

{

saltar\_s(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "SE")==0)

{

saltar\_se(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "L")==0)

{

saltar\_l(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "NE")==0)

{

saltar\_ne(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "N")==0)

{

saltar\_n(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "NO")==0)

{

saltar\_no(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "O")==0)

{

saltar\_o(&r,&linha,&coluna, sentido);

}

else if(strcmp(sentido, "SO")==0)

{

saltar\_so(&r,&linha,&coluna, sentido);

}

}

if(fazer[i] == 'D')/\*virar pra direita\*/

{

if(strcmp(sentido, "N")==0)

{

strcpy(sentido,"NE");

}

else if(strcmp(sentido, "NE")==0)

{

strcpy(sentido,"L");

}

else if(strcmp(sentido, "L")==0){

strcpy(sentido,"SE");

}

else if(strcmp(sentido, "SE")==0)

{

strcpy(sentido,"S");

}

else if(strcmp(sentido, "S")==0)

{

strcpy(sentido, "SO");

}

else if(strcmp(sentido, "SO")==0)

{

strcpy(sentido,"O");

}

else if(strcmp(sentido, "O")==0)

{

strcpy(sentido,"NO");

}

else if(strcmp(sentido, "NO")==0)

{

strcpy(sentido,"N");

}

}

if(fazer[i] == 'E')/\*virar pra esquerda\*/

{

if(strcmp(sentido, "N")==0)

{

strcpy(sentido,"NO");

}

else if(strcmp(sentido, "NO")==0)

{

strcpy(sentido,"O");

}

else if(strcmp(sentido, "O")==0)

{

strcpy(sentido,"SO");

}

else if(strcmp(sentido, "SO")==0)

{

strcpy(sentido,"S");

}

else if(strcmp(sentido, "S")==0)

{

strcpy(sentido, "SE");

}

else if(strcmp(sentido, "SE")==0)

{

strcpy(sentido,"L");

}

else if(strcmp(sentido, "L")==0)

{

strcpy(sentido,"NE");

}

else if(strcmp(sentido, "NE")==0)

{

strcpy(sentido,"N");

}

}

}

printf("\n");

quantidade++;

printar\_relatorio(relatorio, r, quantidade);

printf("Caso: %d\n",quantidade);

printf("Bolinhas coletadas = %d\nSaltos sobre cones =%d\n",r.pontos,r.contcones);

printf("Andou para frente:\nN: %d ", r.contadores[0]);

printf("NE: %d ", r.contadores[1]);

printf("L: %d ", r.contadores[2]);

printf("SE: %d ", r.contadores[3]);

printf("S: %d " , r.contadores[4]);

printf("SO: %d ", r.contadores[5]);

printf("O: %d " , r.contadores[6]);

printf("NO: %d\n" , r.contadores[7]);

liberar\_matriz(r.arena,r.linha,r.coluna);

}

fclose(relatorio);

return 0;

}