

Um parente seu estava te procurando para que você consertasse o seu celular que não estava enviando correntes no "zap".

Embora você sempre explique que cursar Ciência da Computação não é isso, a pessoa não entende e ainda espalha seus serviços para toda a vizinhança, dizendo que você faz bem rápido e de graça.

Você está prestes a ser encontrado e obrigado a realizar tal tarefa. Mas antes que isso aconteça use seus conhecimentos para escapar!

Você está preso em um labirinto, só que ao invés de paredes são seus primos, tios, avós e vizinhos desejando seus serviços.

Corra o mais rápido que puder e não seja pego!

```
Em cada caso de teste você receberá um arquivo de texto contendo o labirinto, formatado da seguinte forma:
M N //Sequência de linhas e colunas do labirinto (Tamanho Total)
x y //Sequência de posição inicial (Partida) (Tamanho Total)
###...N //Sequência de linhas e colunas do labirinto (Tamanho)
0 0...00
0...0,00
0...0,00
0 0,0000
0 0,0000 //Saída do labirinto
#####
```

Observe que cada ponto '.' é uma posição válida enquanto que um '#' é uma pessoa bloqueando sua passagem;

Você deverá:

1. Organizar todos os trechos de código solicitados em funções;
2. Ler o labirinto de um arquivo de texto;
  1. Leia da entrada padrão o nome do arquivo e o abra em seguida para leitura;
3. Alocar dinamicamente a memória para o labirinto;
4. Percorrer o labirinto a partir de uma posição inicial p(x,y) de maneira recursiva.
  1. Respeitar a ordem de busca (Cima, Direita, Baixo, Esquerda);
5. Marcar o seu caminho no labirinto com um caractere asterisco (\*);
6. Ao encontrar a saída, você deve imprimir o labirinto com o seu percurso realizado;
7. Percorra o labirinto e calcule:
  1. *int npessoas* = A quantidade de pessoas (Somar todos os '#');
  2. *int caminhos* = A quantidade de caminhos disponíveis (Somar todos os '.' no início);
  3. *int visitados* = A quantidade percorrida até encontrar a saída (Somar todos os '\*' no final);
  4. *double exploracao* = O percentual de exploração do labirinto até encontrar a saída;
8. Imprima os resultados obtidos no passo (6) da seguinte forma:

```
printf("\nVoce escapou de todos! Ninguem conseguiu te segurar!\n");
printf("Veja abaixo os detalhes da sua fuga:\n");
printf("----Pessoas te procurando: %d\n", npessoas);
printf("----Numero total de caminhos validos: %d\n", caminhos);
printf("----Numero total de caminhos visitados: %d\n", visitados);
printf("----Exploracao total do labirinto: %.1lf%%\n", exploracao);
```

Exemplo de Entrada:

```
5 5
1 1
00000
0 0 0
0 0 0
0 0 0
00000
```

Exemplo de Execução:

Iterações	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
	00000	00000	00000	00000	00000	00000	00000	00000
	070...	070...	070...	070...	070...	070...	0707...	07077
	0 0 0	070 0	070 0	070 0	070 0	07070	07070	07070
	0 ...0	0 ...0	07...0	07...0	07770	07770	07770	07770
	00000	00000	00000	00000	00000	00000	00000	00000

Saída Esperada (Obs.: Somente a última iteração deve ser impressa na tela):

```
00000
07077
07070
07770
00000
```

Você escapou de todos! Ninguém conseguiu te segurar!

Veja abaixo os detalhes da sua fuga:

```
----Pessoas te procurando: 17
----Número total de caminhos válidos: 8
----Número total de caminhos visitados: 8
----Exploração total do labirinto: 100.0%
```