

Trabalho Prático 20

Labirintos

Algoritmos e Estruturas de Dados I

Prazo: **05** de **dezembro** de 2017

1 Introdução

Joãozinho e Zezinho encontraram um novo passatempo: resolver labirintos. Toda banca que passam, insistem para que o pai deles, Pedro, compre revistas com labirintos para eles resolverem.

2 Tarefa

Joãozinho, muito perfeccionista, quer dar mais um passo e saber se está resolvendo o labirinto pelo caminho mais curto o possível. Por isso, ele pediu sua ajuda para criar um programa onde, dado um labirinto, o programa imprima a quantidade mínima de movimentos para resolvê-lo.

3 Entrada

A primeira linha da entrada é composta de 6 inteiros, sendo eles, L (L nunca será maior que 100), que representa a largura do labirinto, A (A nunca será maior que 100), que representa a altura do labirinto, X_c e Y_c , que representam a coordenada (X, Y) inicial no labirinto, e X_f e Y_f , que representam a coordenada (X, Y) final no labirinto. As próximas L linhas são formadas de A caracteres. Cada caractere pode ser "#", caso exista uma parede no labirinto nessa posição, ou ".", caso a posição esteja livre.

As linhas representam uma linha da dimensão Y do labirinto começando em 0 e cada caractere corresponde à uma coluna da dimensão X do labirinto, também começando em 0 (Figura 1).

4 Saída

Seu programa deverá imprimir apenas um inteiro representando o número mínimo de passos para chegar do ponto inicial (X_c, Y_c) até o ponto final do labirinto (X_f, Y_f) passando apenas por posições livres, incluindo o passo em ambos os pontos. Na dúvida, siga o exemplo mostrado abaixo.

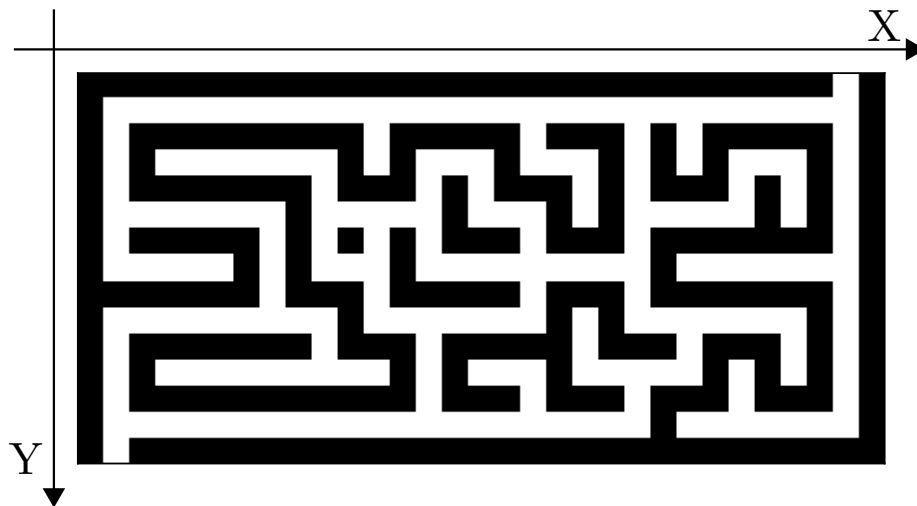


Figura 1: Exemplo de labirinto com os eixos X e Y como definidos

5 Exemplo

Entrada (Representação da Figura 1)

```
31 15 1 14 29 0
#####.#
#.....#
#.#####.#####.###.#.#####.#
#.#.....#.#...#...#.#.#...#.#
#.#####.###.#.###.#.###.#.#.#
#.....#.....#...#.#.....#.#.#
#.#####.#.#.#.###.###.#####.#
#.....#.#...#.....#.....#
#####.###.#####.###.#####.#
#.....#.....#.#.....#.#
#.#####.###.#####.###.###.#.#
#.#.....#.#...#.....#.#.#.#
#.#####.###.###.###.###.#
#.....#.....#
#.#####
```

Saída

43

Outros exemplos estão disponíveis na página deste trabalho no Moodle.

6 Avisos

Avisos mandatórios para o envio do trabalho:

- O código deve ser escrito em linguagem C
- As entradas que serão utilizadas para teste não conterão erros, então não será necessário testar a validade das mesmas
- Não utilize chamadas da função `system` (por exemplo, `system("pause")`) pois essas podem variar de acordo com o sistema operacional e os *softwares* instalados da máquina onde o programa está rodando
- Deixe seu código bem comentado para facilitar a correção
- Não utilize espaços ou caracteres especiais nos nomes dos arquivos. Utilize apenas os caracteres de A a Z (tanto maiúsculas como minúsculas) sem acento, os números de 0 a 9 e os caracteres - (hífen), _ (*underscore*) e . (ponto final)
- Utilize a extensão `.c` para arquivos de código e `.h` para arquivos de cabeçalho, quando aplicar
- Se for submeter os arquivos via upload, não envie um arquivo comprimido (por exemplo, `.zip`, `.rar`, etc.). Utilize os diversos campos da aba “Submissão”, um para cada arquivo
- Envie apenas os arquivos `.c` e `.h`
- Não copie o trabalho de algum colega ou baixe da internet. Lembre-se que o prejudicado será você pois o aprendizado obtido nessa disciplina será utilizado durante diversas outras etapas do seu curso

7 Dicas

Dicas importantes para o desenvolvimento deste trabalho:

- Utilize a função `scanf` para ler as entradas do usuário
- Utilize a função `printf` para imprimir os resultados das operações
- Guarde o labirinto em uma matriz
- Utilize arranjos de caracteres (*strings*) para ler as linhas do labirinto, pois o `scanf` com `%c` também lê espaços em branco e quebras de linha!
- Você deve utilizar *strings* de tamanho no mínimo 101 para conseguir guardar 100 caracteres (devido ao caractere `\0` ao final)
- Note que `scanf("%s", linha)` lê um arranjo de caracteres (*string*) para a variável `linha`

- Você pode acessar um caractere de uma *string* através dos colchetes, assim como é feito em arranjos. Por exemplo, na *string* "abcdef", o índice 0 é a letra **a**, o índice 1 é a letra **b** e assim sucessivamente
- Qualquer dúvida que tiver, utilize o fórum de dúvidas no Moodle. Inicie o assunto do tópico com a tag [TP20]
- Caso prefira, participe das monitorias toda quarta das 17h às 18h na sala 2012

8 Checklist

Checklist não exaustiva de passos até a entrega do trabalho:

1. Pesquise o funcionamento das funções citadas acima para facilitar o uso
2. Implemente e compile o programa
3. Teste para o exemplo dado acima. Compare as saídas para garantir o funcionamento correto
4. Faça o mesmo do item anterior para os exemplos disponibilizados na página deste trabalho no Moodle
5. Envie o trabalho pelo Moodle, onde ele será testado automaticamente para todos os casos disponíveis
6. Caso algum teste dê errado, volte ao passo 2

Bom trabalho e divirta-se!
Boas férias!