

## Dredd - Juiz Online

Principal

Perfil

Minhas Provas

Sair

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

## Exercícios de Recursão

Prova Aberta Até: 27/09/2019 23:59:00

Número Máximo de Tentativas: 6

Atenuação da Nota por Tentativa: 0%

**Instruções para a prova:** Lista de exercícios de Recursão para todas turmas de IAlg. Pode ser acessada de casa.

## Questão 1: Recursividade - Número áureo

O *número áureo*, frequentemente denotado pela letra grega  $\varphi$  (phi) é um número real irracional que ocorre espontaneamente na natureza e é frequentemente usado nas artes por estar relacionado à nossa percepção de beleza.

O *número áureo* pode ser calculado pela recorrência  $\varphi = 1 + 1/\varphi$ .

Por ser uma recorrência infinita, ela precisa ser limitada para ser usada na recursividade da Computação. Podemos definir o valor aproximado de  $\varphi$  em função do número de termos usados no cálculo, assim:

$$\varphi(n) = \begin{cases} 1, & \text{se } n = 1; \\ 1 + \frac{1}{\varphi(n-1)}, & \text{se } n > 1. \end{cases}$$

Faça um programa que tem uma função que calcula uma aproximação do *número áureo*, usando recursão.

O *número áureo*, deve ser do tipo *ponto flutuante de precisão dupla* (double) para possibilitar a precisão necessária nos cálculos. As operações de leitura e escrita devem ser realizadas na função principal.

Entradas:

1. O número de termos para o cálculo da aproximação do *número áureo*.

Saídas:

1. O valor aproximado do *número áureo*.

Exemplo de Entrada:

3

Exemplo de Saída:

1.5

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

**Peso: 1****Última tentativa realizada em:** 27/09/2019 19:35:14**Tentativas:** 1 de 6**Nota (0 a 100):** 100**Status ou Justificativa de Nota:** Nenhum erro encontrado.[Ver Código da Última Tentativa](#)

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#)

Nenhum arquivo selecionado

[Enviar Resposta](#)

## Questão 2: Recursividade - Somas sucessivas.

(BACKES,2012) A multiplicação de dois números naturais pode ser feita através de somas sucessivas (por exemplo,  $2*3=2+2+2$ ). Crie uma **função recursiva** que calcule a multiplicação por somas sucessivas de dois números naturais.

Entradas:

1. Dois números naturais.

Saídas:

1. Resultado da multiplicação dos dois números.

Exemplo de Entrada:

3 5

Exemplo de Saída:

15

**Peso: 1**

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

Última tentativa realizada em: 27/09/2019 19:39:41

Tentativas: 1 de 6

Nota (0 a 100): 10

Status ou Justificativa de Nota: Nem todos os subprogramas exigidos no enunciado foram declarados.

Ver Código da Última Tentativa

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

### Questão 3: Recursividade – Impressão por recursão

Crie um subprograma **recursivo** que receba um número inteiro N e imprima todos os números naturais de 0 até N em ordem crescente.

#### Entradas:

1. Um número inteiro

#### Saídas:

1. Uma sequência de números naturais de 0 até N

#### Exemplo de Entradas:

15

#### Exemplo de Saída:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Peso: 1

Última tentativa realizada em: 27/09/2019 19:43:31

Tentativas: 1 de 6

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

[Ver Código da Última Tentativa](#)

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#) Nenhum arquivo selecionado[Enviar Resposta](#)

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 4: Recursividade – Sequência de Ricci

A sequência de Ricci é uma sequência bastante semelhante à de Fibonacci, diferindo desta apenas pelo fato que os dois primeiros termos da sequência (  $F(0)$  e  $F(1)$  ) devem ser definidos pelo usuário.

Sabendo-se que a sequência de Fibonacci é definida por:

- $F(0) = 0$
- $F(1) = 1$
- $F(n) = F(n - 1) + F(n - 2)$ ,  $n \geq 2$

Crie um algoritmo que imprima os  $n$  primeiros termos da sequência de Ricci, utilizando um **subprograma** que retorna o  $n$ -ésimo termo da referida série.

#### Entrada:

1. Os valores iniciais da série de Ricci (  $F(0)$  e  $F(1)$  );
2. Os número de termos dessa sequência a serem impressos.

#### Saída:

1. Os  $n$  termos dessa sequência.

#### Exemplo de entrada:

5 8  
6

#### Exemplo de saída:

5 8 13 21 34 55

Peso: 1

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#) Nenhum arquivo selecionado[Enviar Resposta](#)

**Questão 5: Recursividade - Função recursiva 1**

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

Faça um programa para calcular a seguinte função recursiva, definida para inteiros **positivos**:

- $f(1) = 1$ .
- $f(2) = 2$ .
- $f(3) = 3$ .
- $f(N) = \text{MIN}(f(N-1), f(N-2), f(N-3)) + 1$ , para  $N > 3$ .

$\text{MIN}(a, b, c)$  é o menor valor entre  $a$ ,  $b$  e  $c$ .

O programa deve ter uma função recursiva que cuida exclusivamente do processamento (cálculo) da função definida acima, enquanto que a função principal faz todas as leituras e escritas, além de fazer uso da função que realiza o processamento.

É permitido criar outras funções além da que realiza o cálculo acima.

**Entradas:**

- Argumento  $n$  da função  $f$ .

**Saídas:**

- $f(n)$ .

**Exemplo de Entrada:**

4

**Exemplo de saída:**

2

**Exemplo de entrada:**

12

**Exemplo de saída:**

5

Peso: 1

Última tentativa realizada em: 27/09/2019 20:06:08

Tentativas: 1 de 6

Nota (0 a 100): 100

**Status ou Justificativa de Nota:** Nenhum erro encontrado.

Ver Código da Última Tentativa

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 6: Recursividade - Torre de Hanoi

O número mínimo  $T(N)$  de movimentos necessários para resolver o problema das Torre de Hanoi com  $N$  discos é dado pela recorrência:

- $T(1) = 1$ .
- $T(N) = 2T(N-1)+1$ , para  $N > 1$ .

Faça um programa que tem uma função recursiva para calcular  $T(N)$ . O programa também deve ter uma função principal que faz as leituras, as escritas e executa a função de cálculo. O planejamento da passagem de parâmetros faz parte da avaliação.

#### Entradas:

- O número de discos ( $N$ )

#### Saídas

- O número mínimo de movimentos  $T(N)$ .

#### Exemplos de Entradas e Saídas:

##### Entrada:

5

##### Saída:

31

##### Entrada:

8

##### Saída:

255

**Peso: 1****Última tentativa realizada em:** 27/09/2019 19:46:18**Tentativas:** 1 de 6**Nota (0 a 100):** 100**Status ou Justificativa de Nota:** Nenhum erro encontrado.[Ver Código da Última Tentativa](#)

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#) Nenhum arquivo selecionado[Enviar Resposta](#)**Minutos Restantes:**  
?**Usuário:**  
Gabriel Nathan  
Almeida Silva**Notas:**  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 7: Recursividade - Coeficientes Binomiais

Denote por  $C(N,K)$  o número de possibilidades de se escolherem  $K$  dentre  $N$  elementos de um conjunto. Este valor é dado pela seguinte relação de recorrência:

- $C(N, 0) = 1.$
- $C(N, N) = 1.$
- $C(N, K) = C(N-1, K-1) + C(N-1, K).$

Faça um programa, utilizando uma função recursiva, para calcular  $C(N,K)$ .

**Entradas:**

- Argumentos  $N$  e  $K$  de  $C$ .

**Saídas**

- $C(N,K).$

**Exemplos de Entradas e Saídas:****Entradas:**

10 7

**Saídas:**

120

**Entradas:**

11 3

**Saídas:**

165

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

**Peso: 1**

**Última tentativa realizada em:** 27/09/2019 19:49:26

**Tentativas:** 1 de 6

**Nota (0 a 100):** 100

**Status ou Justificativa de Nota:** Nenhum erro encontrado.

[Ver Código da Última Tentativa](#)

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

[Escolher arquivo](#)

Nenhum arquivo selecionado

[Enviar Resposta](#)

### Questão 8: Recursividade - Imprimir inverso sem vetor

Faça um programa que leia uma quantidade de números inteiros e os escreva em ordem inversa, sem usar vetor, por meio de uma função recursiva.

É permitido colocar operações de leitura e escrita na função recursiva.

#### Entradas:

1. Um número inteiro indicando a quantidade de valores,
2. vários valores inteiros

#### Saídas:

1. Os valores da entrada, em ordem inversa.

#### Exemplo de Entradas:

15  
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

#### Exemplo de Saída:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

**Peso: 1**

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.



Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 9: Recursividade - Funções mutuamente recursivas

Escreva um programa que calcula o valor da função  $F$ , definida recursivamente:

$F(x) = 1$ , se  $x$  igual a 0,  
 $F(x) = 2H(x) + F(x-1)$ , se  $x$  maior que 0 e par,  
 $F(x) = 2H(x) - F(x-1)$ , se  $x$  maior que 0 e ímpar.

Observe que  $F$  é definida em termos de  $H$ , apresentada abaixo:

$H(x) = 0$ , se  $x$  igual a 0,  
 $H(x) = H(x-1) + F(x-1)$ , se  $x$  maior que 0.

#### Entradas:

1. Valor inteiro  $x$  positivo para o qual se deseja calcular o valor de  $F(x)$ .

#### Saídas:

1. Valor de  $F(x)$

Exemplo de Entrada:  
8

Exemplo de Saída:  
2705

=====  
Exemplo de Entrada:  
5

Exemplo de saída:  
73

Peso: 1

Última tentativa realizada em: 27/09/2019 20:20:25

Tentativas: 1 de 6

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 10: Recursividade - Sem adição

Considere que um determinado sistema numérico fictício não possua a operação de adição. Desenvolva uma função recursiva que calcule a soma de dois números inteiros  $X$  e  $Y$ . Ao implementar esta função recursiva é permitido utilizar outras duas funções auxiliares não recursivas, chamadas sucessor e antecessor. A função sucessor é responsável por receber um número inteiro  $N$  e retornar o valor deste número incrementado em 1, enquanto a função antecessor é responsável por receber um número inteiro  $N$  e retornar o valor deste número decrementado em 1. Note que apenas estas duas funções auxiliares (sucessor e antecessor) podem utilizar os operadores de incremento ( $++$ ) e decremento ( $--$ ). Seu programa **não** pode utilizar estruturas de repetição (for, while e do-while).

#### Entradas:

1. Um inteiro  $X$ .
2. Um inteiro  $Y$ .

#### Saídas:

1. Um inteiro que indica o valor de  $X + Y$

#### Exemplo de entrada:

2 3

#### Exemplo de saída:

5

#### Exemplo de entrada:

7 -15

#### Exemplo de saída:

-8

Peso: 1

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 11: Recursividade - Série Harmônica na matriz

Considerando a definição de série harmônica  $H(N)$  a seguir:

$$H(N) = 1, \text{ para } N = 1$$

$$H(N) = (1/N) + H(N-1), \text{ para } N \geq 2$$

Faça um subprograma que calcule o valor da série harmônica para um  $N$  qualquer positivo. Além disso, faça um programa que leia um número inteiro positivo  $M$  e crie uma matriz quadrada de tamanho  $M$  de números em ponto flutuante (float). Os elementos dessa matriz são dados pelos valores da série harmônica com índices variando no intervalo  $[1, M \times M]$ , ou seja, o primeiro elemento da matriz é  $H(1)$ , o segundo ( $H(2)$ ), e assim por diante. Note que o último elemento é definido como  $H(M \times M)$ . Os valores devem ser armazenados na matriz percorrendo-se a orientação das linhas da mesma. Exemplo de matriz com  $M = 3$ :

1 1.5 1.83333

2.08333 2.28333 2.45

2.59286 2.71786 2.82897

Seu programa deverá informar a diferença do somatório da diagonal secundária da matriz pelo somatório da diagonal principal.

#### Entradas:

1. Um inteiro  $M$ , representando as dimensões da matriz

#### Saídas:

1. Um número em ponto flutuante (float) representando a diferença da diagonal secundária da matriz pela da diagonal principal

#### Exemplo de entrada:

3

#### Exemplo de saída:

0.597222

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

### Questão 12: Recursividade – Média aritmética

A média aritmética de  $n$  valores é facilmente calculada apenas somando todos os valores e dividindo esse resultado pela quantidade de valores somados. O cálculo da média pela forma descrita é dito iterativo, pois é um método sequencial que a cada iteração acumula-se os valores anteriores para finalmente efetuar a divisão. Porém também é possível calcular a média através de um método recursivo.

Elabore um programa que peça para o usuário um número que defina o tamanho de um vetor de inteiros e, em seguida, preencha-o com cada um de seus elementos. Em seguida, crie uma função recursiva que retorne a média aritmética dos elementos desse vetor e exiba esse valor na saída padrão.

#### Entrada:

1. O tamanho do vetor;
2. Os elementos do vetor.

#### Saída:

1. O valor da média, retornado por uma função recursiva.

#### Exemplo de entrada:

```
10
2 4 6 8 10 12 14 16 18 20
```

#### Exemplo de saída:

```
11
```

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

**Questão 13: Recursividade - Encontrar elemento no vetor**

Faça um programa que recebe um vetor V de N elementos inteiros e um elemento inteiro adicional x e determine, de forma recursiva, se x está ou não no vetor. Podemos usar a seguinte ideia:

- Se x está na primeira posição do vetor, então foi encontrado.
- Senão, procuramos no restante do vetor.

**Entradas:**

- Tamanho N do vetor.
- Elementos do vetor.
- Elemento a ser procurado.

**Saídas**

- 1 se x estiver no vetor. 0 senão.

**Exemplos de Entradas e Saídas:****Entradas:**

4  
2 4 3 1  
3

**Saídas:**

1

**Entradas:**

5  
9 4 6 8 7  
2

**Saídas:**

0

**Peso: 1**

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

**Questão 14: Recursividade - Menor do Vetor**

Faça um programa que recebe um vetor V de N elementos e determina, de forma recursiva, o menor elemento do vetor. Podemos usar a seguinte ideia:

**Minutos Restantes:**  
?

**Usuário:**  
Gabriel Nathan  
Almeida Silva

**Notas:**  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

- O menor elemento de um vetor de uma única posição é o seu único elemento.
- O menor elemento de um vetor de mais de uma posição é o menor entre o primeiro elemento e o menor do restante do vetor.

#### Entradas:

- Tamanho N do vetor.
- Elementos do vetor.

#### Saídas

- Menor elemento do vetor.

#### Exemplos de Entradas e Saídas:

##### Entradas:

4  
2 4 3 1

##### Saídas:

1

Peso: 1

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

### Questão 15: Recursividade - Remoção de letra

Faça um programa que remove uma letra de uma sequência de caracteres. O programa deve **ter uma sub-rotina recursiva** que faz a remoção propriamente dita. As operações de leitura e escrita devem estar na sub-rotina principal.

É permitido usar operações da linguagem ou de bibliotecas que contam o número de caracteres numa string.

Entradas:

1. Uma sequência de caracteres (considere que ela não tem espaços e não terá mais que 79 caracteres).
2. Uma posição para remover (a primeira posição é zero, considere que a posição é sempre válida).

Saídas:

1. O resultado da remoção da letra na posição indicada.

Minutos  
Restantes:  
?

Usuário:  
Gabriel Nathan  
Almeida Silva

Notas:  
Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

Exemplo de Entrada:

corta  
2

Exemplo de Saída:

cota

**Peso: 1**

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta

### Questão 16: Recursividade - Caminho na Matriz

Dada uma matriz para representar posições no espaço, queremos encontrar um caminho entre duas posições. Encontrar um caminho significa encontrar posições por onde é possível passar.

Este processo pode ser implementado com facilidade usando recursão. Para simplificar ainda mais uma implementação, vamos supor que só é possível caminhar para a direita ou para baixo na matriz. Podemos descrever um caminho de uma origem para um destino assim:

- O caminho é mover-se para posição da direita e depois percorrer o caminho da posição da direita até o destino, se é possível mover-se para direita e se o caminho da posição da direita até o destino existe.
- O caminho é mover-se para posição de baixo e depois percorrer o caminho da posição de baixo até o destino, se a alternativa anterior não resolveu o problema, se é possível mover-se para baixo e se o caminho da posição de baixo até o destino existe.

A estratégia recursiva acima não tem um caso base, isso fica para você determinar. Note que a estratégia está baseada em tentar à direita antes de tentar abaixo.

Faça um programa que lê uma matriz 10x15 de caracteres representando um lugar. Nessa matriz haverá o caractere '.' (ponto) nas posições em que é possível transitar e haverá o caractere '#' (cerquilha) nas posições em que não é possível transitar (obstáculos). Encontre na matriz um caminho do canto superior esquerdo até o canto inferior direito usando a estratégia acima. O programa deve alterar a matriz, alterando todas as posições analisadas com um 'x' (xis minúsculo), o que permite não apenas ver o caminho encontrado, mas também quanto processamento foi necessário para encontrá-lo.

**Minutos Restantes:**  
?

**Usuário:**  
Gabriel Nathan  
Almeida Silva

**Notas:**

Q1: 100  
Q2: 10  
Q3: 100  
Q4: ?  
Q5: 100  
Q6: 100  
Q7: 100  
Q8: ?  
Q9: 100  
Q10: ?  
Q11: ?  
Q12: ?  
Q13: ?  
Q14: ?  
Q15: ?  
Q16: ?  
Total: 38

Seu programa deverá ter uma função recursiva que encontra caminhos numa matriz e faz as marcações das posições analisadas. Planeje os parâmetros com cuidado. Não é permitido o uso de variáveis globais.

A função principal deve escrever a matriz com as posições marcadas depois do caminho ter sido encontrado.

Exemplo de Entrada:

```
.....#.....
.....###.####
###.....#..
..#.#####.###
..#.#####.###
..#.#####.###
..#.#####.###
..#.#####.###
###.#####..
.....###
.....
```

Exemplo de Saída:

```
xxxxxx#.....
...xxx###.####
###xxxxxxxxxx#..
..#x#####x###
..#x#.....#x#..
..#x#.....#x#..
..#x#.....#x#..
###x#####..
...xxxxxxxxx###
.....xxxx
```

**Peso: 1**

Nova Resposta: \_\_\_\_\_

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo

Nenhum arquivo selecionado

Enviar Resposta



Desenvolvido por Bruno  
Schneider a partir do programa  
original (Algod) de Renato R.  
R. de Oliveira.

