

Trabalho Prático 16

Sinuca

Algoritmos e Estruturas de Dados I

Prazo: **01** de **dezembro** de 2017

1 Introdução

Joãozinho e Zezinho inventaram um passatempo com bolas de sinuca, pretas e brancas, que são colocadas uma por vez na mesa, de acordo com uma regra fixa. Funciona assim. No início, são colocadas N bolas formando a primeira fileira. Em seguida, um triângulo equilátero é formado, fileira a fileira, de acordo com a seguinte regra. Ao se colocar uma bola na nova fileira, ela ficará encostada em duas bolas da fileira anterior e sua cor será:

- Preta, se estiver encostada em duas bolas de mesma cor;
- Branca, se estiver encostada em duas bolas de cores diferentes.

A Figura 1 ilustra a formação de um triângulo para $N = 5$.

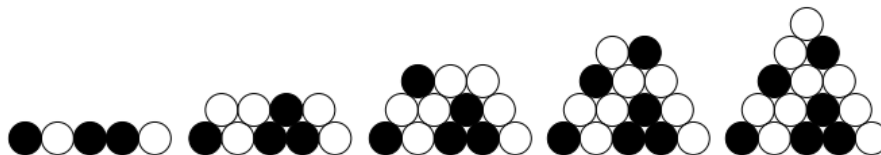


Figura 1: Formação de um triângulo para $N = 5$

2 Tarefa

Agora eles estão tentando descobrir, com um computador, como serão as cores das bolas colocadas. Por isso, eles pediam sua ajuda. Você deve escrever um programa que, dadas as cores das bolas da primeira fileira, imprima, começando da primeira fileira e da esquerda para a direita, as cores de cada bola colocada no triângulo.

3 Entrada

A entrada é composta de apenas uma linha contendo uma sequência de caracteres x e o , representando as cores das bolas da primeira fileira, onde x é uma bola preta e o é uma bola branca. Tal sequência nunca terá mais que 1000 caracteres.

4 Saída

Seu programa deverá produzir várias linhas, onde cada uma representa as cores, no formato x para preta e o para branca, das bolas da fileira atual, **começando da primeira**. Assim como no exemplo, o número de linhas produzido para cada entrada é igual ao número de bolas na primeira linha. Na dúvida, siga o exemplo abaixo.

5 Exemplo

Entrada

```
xxxxo
```

Saída

```
xxxxo
ooxo
xoo
ox
o
```

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 programas 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
- Note que `scanf("%s", linha)` lê um arranjo de caracteres (*string*) para a variável `linha`
- Você deve utilizar *strings* de tamanho no mínimo 1001 para conseguir guardar 1000 caracteres (devido ao caractere `\0` ao final)
- Qualquer dúvida que tiver, utilize o fórum de dúvidas no Moodle. Inicie o assunto do tópico com a tag [TP16]
- 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!