

# Trabalho Prático 4

## Poodle

### Algoritmos e Estruturas de Dados I

Prazo: **20** de **outubro** de 2017

## 1 Introdução

Maria, mãe de Joãozinho e Zezinho, vive perdendo coisas dentro de casa. Isso acontece porque ela é muito desorganizada, deixa tudo espalhado pela casa, tornando impossível localizar algum objeto no meio de tanta confusão. Ela sempre contou com a ajuda infalível de seu cãozinho Poodle, que consegue localizar seus objetos perdidos.

Uma vez ela queria me mostrar a eficiência do seu Poodle. Escondeu propositalmente uma bola em um dos quartos, e gritou: “Pooooooooodle!”. Então ela disse “Bola” e ele partiu para buscá-la. Ela ficou preocupada porque depois de 30 segundos ele ainda não tinha retornado com a bola. A surpresa foi que logo depois ele apareceu, triunfante, carregando 3 bolas! A que Maria acabara de esconder e outras 2 de seus filhos, que ninguém se lembrava que existiam!

Recentemente Maria, sendo muito inteligente, teve a ideia de criar uma ferramenta de busca. Em homenagem a seu cãozinho que sempre buscou suas coisas, ela batizou sua ideia de Poodle. Como no caso real da bola acima, na maioria das vezes a busca retorna bem mais resultados que o esperado. Os resultados são então exibidos agrupados em páginas. Por exemplo, se a ferramenta for configurada para exibir 10 resultados por página, e a busca retornar 143 resultados, eles serão exibidos em 15 páginas: 14 delas com 10 em cada uma, e a última com os 3 restantes.

Ao exibir o resultado de uma busca, Poodle mostra um logotipo com o nome da ferramenta, sendo que os o’s de Poodle podem ser um ou mais, dependendo da quantidade de páginas de resultado. A ideia é que “Poodle” seja escrito com tantas letras “o” quantas forem as páginas de resultado. No exemplo acima, o logotipo seria “Pooooooooooooooooodle”, que contém 15 letras “o”.

## 2 Tarefa

Você, que já se tornou o programador oficial da família, recebeu a tarefa de ajudar Maria a montar o logotipo.

### 3 Entrada

A entrada é composta de vários casos de teste. Cada caso de teste é uma linha contendo dois números inteiros,  $N$  e  $P$ , sendo  $N$  o número de documentos encontrados pelo Poodle, e  $P$  o número de resultados exibidos por página. A entrada termina quando  $N = P = 0$ .

### 4 Saída

Para cada caso de teste da entrada seu programa deve produzir uma linha na saída contendo a palavra Poodle, ajustando a quantidade de letras “o” de acordo com as regras descritas no enunciado. Na dúvida, siga o exemplo abaixo.

### 5 Exemplo

#### Entrada

```
15 8
20 4
143 10
42 5
10 15
80 3
26 49
81 26
79 12
79 44
0 0
```

#### Saída

```
Poodle
Poooooodle
Poooooooooooooooooodle
Poooooooooodle
Podle
Poooooooooooooooooooooooooooooooooodle
Podle
Poooooodle
Poooooooooodle
Poodle
```

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
- Chamadas do `printf` não quebram linha se você não utilizar o caractere “`\n`” ao final da chamada
- Muito cuidado com divisões. Se você divide dois inteiros (do tipo `int`, `long` e derivados), o padrão da linguagem `C` é arredondar para baixo. Converta para `float` ou `double` antes!
- Qualquer dúvida que tiver, utilize o fórum de dúvidas no Moodle. Inicie o assunto do tópico com a tag [TP4]
- 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!