

# Trabalho Prático 5

Roleta

## Algoritmos e Estruturas de Dados I

Prazo: **24** de **outubro** de 2017

### 1 Introdução

Todo domingo à tarde, quando a família está reunida, Joãozinho, Zezinho, Pedro e Maria gostam de jogar um jogo baseado em uma roleta. O jogo é bem simples. Todos começam com 0 pontos e, a cada rodada, eles giram a roleta. A roleta é formada por dois tipos de comandos, sendo que o primeiro define qual jogador vai rodar a roleta e o segundo define quantos pontos ele vai ganhar ou perder.

Quando um jogador é selecionado, ele poderá ficar por várias rodadas ganhando ou perdendo pontos, até que a roleta selecione um novo jogador. O primeiro a jogar é definido anteriormente pelos próprios jogadores. Um jogador nunca poderá ter menos de 0 pontos, nesse caso, se, por exemplo, um jogador tem 100 pontos e perde 200, eles ficará com 0 pontos ao invés de  $-100$ .

### 2 Tarefa

A família descobriu que é muito cansativo ficar acompanhando todas as alterações de pontos no jogo, fazendo com que ele perca um pouco da graça. Então eles resolveram te pedir para criar um programa que, dadas as rodadas, faz isso automaticamente e imprime o placar. **Você deve usar ponteiros para implementar o programa!** Mais especificamente, você deve utilizar um ponteiro para o número de pontos do jogador atual.

### 3 Entrada

A entrada é composta de várias jogadas. A primeira linha da entrada define quem é o jogador inicial, sendo que 1 representa Joãozinho, 2 representa Zezinho, 3 representa Pedro e 4 representa Maria. As próximas linhas são compostas de dois números  $C$  e  $V$ , separados por espaço, que descrevem as jogadas.  $C$  define qual o comando a ser tomado, sendo  $C = 1$  indicando que o jogador será trocado e  $C = 2$  que os pontos do jogador selecionado serão atualizados.

No caso de  $C = 1$ ,  $V$  qual o novo jogador selecionado (segundo a mesma numeração da primeira linha) e, no caso de  $C = 2$ ,  $V$  define quantos pontos

serão acrescidos ou removidos (caso  $V < 0$ ) do jogador. O final da entrada é indicado por  $C = V = 0$ .

## 4 Saída

Para cada rodada, seu programa deve imprimir qual o nome do jogador ativo (sem acentuação!) e, separado por espaço, qual a pontuação de todos os jogadores no formato `J x Z x P x M` onde J representa a pontuação de Joãozinho, Z de Zezinho, P de Pedro e M de Maria. Na dúvida, siga o exemplo mostrado abaixo.

## 5 Exemplo

### Entrada

```
3
2 10
2 -5
1 4
2 30
1 3
2 5
2 -15
1 2
2 -5
2 0
2 7
0 0
```

### Saída

```
Pedro 0 x 0 x 10 x 0
Pedro 0 x 0 x 5 x 0
Maria 0 x 0 x 5 x 0
Maria 0 x 0 x 5 x 30
Pedro 0 x 0 x 5 x 30
Pedro 0 x 0 x 10 x 30
Pedro 0 x 0 x 0 x 30
Zezinho 0 x 0 x 0 x 30
Zezinho 0 x 0 x 0 x 30
Zezinho 0 x 0 x 0 x 30
Zezinho 0 x 7 x 0 x 30
```

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
- Para utilizar ponteiros, pense que a cada troca de jogadores, você atualiza o ponteiro para quem está selecionado
- Você pode descobrir para onde um ponteiro está apontando se comparar ele com o endereço da variável na forma `ponteiro == &variavel`. Essa expressão retornará 1 caso `ponteiro` aponte para `variavel` e 0 caso contrário
- Lembre-se que apenas a primeira linha da entrada não produz saída!

- Qualquer dúvida que tiver, utilize o fórum de dúvidas no Moodle. Inicie o assunto do tópico com a tag [TP5]
- 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!