MC202 - Estruturas de Dados

Lab 03

Data da Primeira Chance: 28 de agosto de 2023
Link da atividade: https://classroom.github.com/a/3uKzhNI-
Peso: 2

Jonathan e Elisa tem um campo recreativo de minigolfe que possui 50 circuitos (ou buracos) no total. Atualmente, todos os registros de partidas são manuscritos, de forma que os donos desejam um sistema para facilitar o gerenciamento, e também auxiliar os jogadores a verem seus resultados após as partidas.

Cada grupo de X pessoas que deseja utilizar o campo deve escolher C circuitos. O preço do aluguel é dado devido à quantidade de equipamentos (E) e circuitos escolhidos (C), lembrando que cada pessoa do grupo deverá alugar um equipamento (os donos não permitem compartilhar os equipamentos alugados ou que as pessoas tragam seus próprios equipamentos). O tempo de uso do campo não é importante, porém cada grupo utilizará apenas uma vez cada circuito incluso no pacote. O valor final do pacote de aluguel (V) é dado conforme a seguinte função:

$$V = 10 * C + 17.50 * E$$

Nos requisitos do sistema do campo de minigolfe, os donos precisam que seja gerado um relatório com os dados de cada grupo que alugou, bem como os resultados de cada circuito e o valor final do pacote alugado.

Finalmente, os donos também definem as regras de um circuito e da partida.

- Uma partida é composta por diversos circuitos e um determinado número de participantes
- Cada participante tem, no máximo, nove jogadas para terminar cada circuito

Para fazer esse sistema, crie um TAD do tipo ***circuito***. Nesse TAD, armazene as seguintes informações referentes a um dado circuito do pacote alugado:

- Identificação do circuito
- Número de participantes

- Um vetor **dinamicamente** alocado, armazenando o número de jogadas de cada participante do circuito

Para gerenciar um circuito, você deve utilizar pelo menos a seguinte função:

- leitura dos dados de um circuito
    - Identificação do circuito
    - Número de jogadas de cada jogador na partida (a ordem dos jogadores será sempre a mesma para todos os circuitos)

Crie também um TAD do tipo ***partida***. Nesse TAD, armazene as informações referentes ao pacote que um grupo escolheu:

- Um *float* armazenando o valor do aluguel da partida
- Identificação da partida (o sistema deve gerar automaticamente)
- Número de participantes no grupo
- Número de circuitos no pacote
- Número de equipamentos alugados
- um vetor do tipo *circuito,* alocado **dinamicamente**, com as informações de cada circuito no pacote

Para gerenciar uma partida, você deve utilizar pelo menos as seguintes funções:

- leitura dos dados de uma partida
    - número de circuitos
    - número de jogadores
    - informações de cada circuito no pacote
    - número de equipamentos alugados

- impressão do relatório da partida
    - número de circuitos e jogadores
    - valor total do aluguel da partida
    - jogador vencedor
    - circuito mais difícil (maior número de jogadas necessárias para todos os jogadores completarem o circuito, primeiro circuito registrado em caso de empate)
    - tabela com o resultado de cada participante

Obs.: Neste laboratório, você pode (e provavelmente terá que) criar novas funções além das já indicadas.

Entrada

A primeira linha da entrada contém o número de partidas a ser registrado.

Para cada partida, temos uma linha de entrada contendo um inteiro $X \geq 1$, representando o número de jogadores, seguido por um inteiro, $1 \leq C \leq 50$ representando o número de circuitos da partida, terminando com um inteiro $0 \leq E \leq X$ representando o número de equipamentos alugados.

Para cada circuito, temos uma linha de entrada com $X + 1$ inteiros. O primeiro inteiro representa o número de identificação do circuito, e o restante, inteiros no intervalo $1 \leq J < 10$, representam o número de jogadas de cada participante no circuito.
Obs.: a ordem dos jogadores é sempre a mesma para todos os circuitos de uma partida.

Saída

A saída é composta por um relatório de cada partida. O relatório deve ser mostrado da seguinte forma:

- Uma linha com a identificação da partida no formato: `Partida %d` - (**Obs**.: identifique as partidas na ordem de leitura, começando por 1).
- Uma linha com a identificação do número de jogadores e circuitos no formato: `Num. de Jogadores: %d - Num. de Circuitos: %d - Num. de Equipamentos: %d`
- Uma linha com o valor do aluguel da partida: `Valor do Aluguel: R$ %.2f`
- Para cada participante, imprima uma linha contendo a identificação do participante e a quantidade somada de jogadas necessárias para todos os circuitos, no formato: `Jogador %d: %d` - (**Obs**.: identifique os participantes conforme a leitura das jogadas no circuito, começando por 1).
- Uma linha dizendo qual foi o circuito mais difícil, no formato: `Circuito mais dificil: %d` - (**Obs**.: em caso de empate, o mais difícil é o primeiro circuito registrado. Cada circuito é jogado apenas 1 vez).
- Finalize o relatório com uma linha contendo: `##########`

Exemplos

Exemplo 1:

Entrada

```
1
2 5 2
1 3 3
2 1 2
3 9 7
4 2 8
5 7 6
```

Saída

```
Partida 1
Num. de Jogadores: 2 - Num. de Circuitos: 5 - Num. de
Equipamentos: 2
Valor do Aluguel: R$ 85.00
Jogador 1: 22
Jogador 2: 26
Circuito mais dificil: 3
##########
```

Exemplo 2:

Entrada

```
2
3 2 3
8 1 1 2
27 3 4 2
10 1 10
50 1 2 3 3 2 7 8 3 9 5
```

Saída

```
Partida 1
Num. de Jogadores: 3 - Num. de Circuitos: 2 - Num. de
Equipamentos: 3
Valor do Aluguel: R$ 72.50
Jogador 1: 4
Jogador 2: 5
Jogador 3: 4
Circuito mais dificil: 27
##########
Partida 2
Num. de Jogadores: 10 - Num. de Circuitos: 1 - Num. de
Equipamentos: 10
Valor do Aluguel: R$ 185.00
Jogador 1: 1
Jogador 2: 2
Jogador 3: 3
Jogador 4: 3
Jogador 5: 2
Jogador 6: 7
Jogador 7: 8
Jogador 8: 3
Jogador 9: 9
Jogador 10: 5
Circuito mais dificil: 50
##########
```

Sobre este Laboratório

Este laboratório possui algumas particularidades. Para guiá-la (o), aqui vão algumas dicas:

- Quando realizar impressão de variáveis ***float***, ou ***double***, você pode especificar o número de casas decimais a ser mostrado após o ponto flutuante. Por ex: `%.2f` fará que a impressão do número resultante tenha 2 casas decimais.
- Um dos objetivos desse laboratório é treinar a criação de módulos. Por isso, mantenha um módulo ***partida*** e outro ***circuito***. A modularização serve não apenas para separar trechos de código que são distintos, mas também para facilitar a eventuais atualizações nos módulos caso necessário.
- Lembre-se que suas interfaces também podem usar outras interfaces!
- Divida cada uma de suas interfaces em conjuntos de arquivos (.c e .h) diferentes. Na hora de compilar seu programa, é melhor compilar por partes, conforme visto em aula. Gere todos seus arquivos objeto (.o) e então compile o arquivo final fazendo o *link* de todos os arquivos compilados anteriormente.
- Finalmente, a alocação de vetores nesse laboratório é ***dinâmica***, não esqueça que se você alocou memória dinamicamente, você deve liberá-la antes de encerrar o programa.

Regras e Avaliação

Seu código será avaliado não apenas pelos testes do GitHub, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código analisaremos neste laboratório:

- A escolha de bons nomes de funções e variáveis;
- A ausência de trechos de código repetidos desnecessariamente;
- O uso apropriado de funções;
- A eficiência dos algoritmos propostos;
- Uso das TADs propostas;
- A liberação correta da memória alocada dinamicamente;
- Todos os vetores devem ser **alocados dinamicamente**.

Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Submissão

Você deverá criar 5 arquivos: os arquivos `partida.c` e `partida.h` que implementam o TAD relacionado a partida; os arquivos `circuito.c e circuito.h` que implementam o TAD relacionado a um circuito; e o arquivo `golfe.c` com o corpo principal do programa, e submeter no repositório criado no aceite da tarefa. Você também pode enviar arquivos adicionais caso necessário.

Lembre-se que sua atividade será corrigida automaticamente na aba “Actions” do repositório. Confirme a correção e o resultado, já que o que vale é o que está lá e não na sua máquina.

Após a correção da primeira entrega, será aberta uma segunda chance, com prazo de entrega apropriado.