

Tecnologia em Jogos Digitais

Algoritmos e Programação I

Score Game

Na maioria dos games, ao final da partida, a pontuação do jogador é armazenada para posterior consulta e comparação entre os jogadores. Assim uma das funcionalidades importante em um game é armazenar, recuperar e administrar as pontuações dos jogadores. As pontuações dos jogadores são armazenadas no game como se fosse uma lista de scores, sendo assim, o objetivo do segundo projeto da disciplina de Algoritmos e Programação I é implementar um programa que administra uma lista de scores de um game. No seu programa o usuário poderá solicitar as seguintes operações:

1. Adicionar uma pontuação no final da lista de scores;
2. Adicionar uma pontuação em uma dada posição na lista de scores;
3. Remover uma pontuação de uma dada posição na lista de scores;
4. Remover todas as ocorrências de uma pontuação na lista de scores;
5. Verificar se uma pontuação está contida lista de scores;
6. Buscar a maior pontuação na lista de scores;
7. Calcular a soma total de pontuação na lista de scores
8. Verificar se dado uma pontuação existem duas pontuações na lista de scores que somadas são iguais à pontuação informada.

Para o usuário escolher as opções, o seu programa deverá mostrar um **menu de opções**, sendo que para cada opção deverá ser executada a operação correspondente, por exemplo, para adicionar uma pontuação no final da lista de scores, o usuário deve escolher a opção **Adicionar no Final** e digitar uma pontuação que será inserida no final da lista de scores.

O seu programa deverá executar até o momento que usuário solicite finalizar a aplicação, ou seja, o usuário pode escolher outras operações do **menu de opção** várias vezes. Ao final da execução de cada operação **é impresso** os elementos válidos na lista de score. Para armazenar a lista de scores utilizaremos um **vetor de inteiro**. A quantidade de elementos na lista é **limitada a 100**, mas o seu programa deverá ter uma variável para controlar a quantidade de pontuações válidas armazenadas no vetor, além disso, a lista pode conter mais de uma cópia de uma mesma pontuação.

A lista é armazenada de forma contínua e estável, ou seja, não poderemos ter “buracos” no meio do vetor e as operações de adicionar e de remover não alteram a posição relativa dos elementos no vetor, e caso tenham sucesso deve ser atualizada a quantidade de elementos na de score. A seguir são descritas detalhadamente as operações:

1. Adicionar uma pontuação no final da lista de scores.

Adiciona a nova pontuação na primeira posição disponível no final da lista. Caso a lista já esteja no limite, a pontuação não é adicionada e a quantidade de elementos não é modificada.

2. Adicionar uma pontuação em uma dada posição na lista de scores.

A operação de adicionar uma pontuação em uma determinada posição é mais delicada. Primeiro, precisamos verificar se a posição faz sentido ou não, ou seja, só podemos adicionar uma pontuação em alguma posição que já estava ocupada ou na primeira posição disponível no final do vetor. Caso a posição seja válida, devemos tomar cuidado para não colocar um elemento sobre outro. É preciso deslocar todos os elementos a “direita” da posição onde vamos inserir uma vez para a “frente”. Isso abrirá um espaço para guardar o novo elemento. Se conseguirmos inserir a nova pontuação a quantidade de elementos no vetor é atualizada.

3. Remover uma pontuação de uma dada posição na lista de scores;

Para essa operação precisamos verificar se a posição está ocupada ou não, se a posição estiver ocupada então podemos remover a pontuação. Para isso basta deslocar os elementos que estavam à direita daquele que removemos uma vez para esquerda e fechamos o “buraco” aberto pela remoção.

4. Remover todas as ocorrências de uma pontuação na lista de scores;

Essa operação remove todas as ocorrências de uma dada pontuação no vetor, e também não pode deixar “buracos” após a remoção.

5. Verificar se uma pontuação está contida lista de scores;

Nessa operação será informado o valor de uma pontuação o seu programa deve informar e a pontuação está lista sim ou não.

6. Buscar a maior pontuação na lista de scores;

Nessa operação não informado nada, basta o seu programa buscar a maior pontuação.

7. Calcular a soma total de pontuação na lista de scores

Nessa operação não informado nada, basta o seu programa calcular a soma das pontuações.

8. Verificar se dado uma pontuação existem duas pontuações na lista de scores que somadas são iguais à pontuação informada.

E por fim, temos a operação que verifica se existe no vetor duas pontuações, em posições distintas, que somadas tem como resultado a pontuação informada, o seu programa deve informar sim ou não se existe os a pontuação igual a soma de duas outras pontuações distintas.

Critérios de avaliação

O programa entregue **será avaliado** de acordo com os seguintes itens:

- Funcionamento do programa;
- O programa deve estar na linguagem **Java**.
- O quão fiel é o programa quanto à descrição do enunciado;
- Identação, comentários e legibilidade do código;
- **Implementar pelo menos 4 operações utilizando funções**. lembre-se que dentro de uma função não se pode fazer leitura e nem saída de dados, as informações serão passadas por parâmetro e devolvidas com retorno.
- Clareza na nomenclatura de variáveis e funções.

Grupo

A atividade pode ser feita em grupo de **até 3 alunos** (somente uma entrega no **Blackboard** com o nome dos integrantes do grupo). Não serão aceitos trabalhos por e-mail. Como a atividade pode ser feito em um grupo de **três pessoas**, evidente você pode “discutir” o problema dado com outros grupos, inclusive as “dicas” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento do seu programa. Ou seja, qualquer tentativa de fraude será punida com a **nota zero**.