

Aula Prática 3 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho
Universidade Federal de Ouro Preto
Departamento de Computação

7 de junho de 2021

Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

1 Alocação dinâmica de tipos abstratos de dados

No ano de 2021 o Papai Noel não poderá sair de casa para entregar presentes por conta da pandemia do Coronavírus. Então ele ordenou que seus duendes fossem entregar os presentes no lugar dele no dia do natal. Como eles são bastante inexperientes, irão se dividir em vários times compostos de três membros: Um **líder**, um **entregador** e um **piloto** de trenó. O plano do Papai Noel é que os líderes das equipes seja sempre os duendes mais velhos. Por esse motivo ele pediu para todos os duendes escreverem seus nomes e idades em uma lista. Como você é um duende programador, resolveu ajudar o Papai Noel a organizar a lista e montar os times a partir dela. Seguem abaixo algumas regras e fatos:

- A lista deve ser organizada em ordem decrescente de idade;
- Não existem dois duendes com a mesma idade;
- Não existem dois duendes de mesmo nome;
- Nenhum duende tem mais de 20 caracteres em seu nome;
- Os duendes da lista têm idade entre 10 e 100 anos;
- O primeiro 1/3 dos duendes (os mais velhos), serão os líderes dos times;
- O próximo 1/3 dos duendes serão os entregadores dos times;
- O último 1/3 dos duendes serão os pilotos dos times

Exemplo: Se há 6 duendes na lista, haverá dois times, onde o duende mais velho é líder do time 1, e o segundo mais velho é líder do time 2. O terceiro mais velho é entregador do time 1 e o quarto mais velho é entregador do time 2. O quinto é piloto de trenó do time 1 e o último é piloto do time 2.

Especificação da Entrada

A entrada é composta de um número inteiro n , múltiplo de 3, que representa a quantidade de duendes na lista. Em seguida as próximas n linhas contém o nome e a idade de cada duende.

Especificação da Saída

A saída é composta de 4 linhas por time. A primeira linha deve seguir o formato “Time X”, onde X é o número do time. A segunda, terceira e quarta linha contém, respectivamente, o nome e idade do duende líder, entregador e piloto de trenó. Depois de cada time, deverá haver uma linha em branco, inclusive após o último time.

Exemplo de Entrada

```
9
Kepeumo 67
Necoi 62
Seies 77
Ciule 49
Gyun 99
Finron 27
Norandir 66
Galvaindir 55
Pinhuobor 70
```

Exemplo de Saída

```
Time 1
Gyun 99
Kepeumo 67
Galvaindir 55
```

```
Time 2
Seies 77
Norandir 66
Ciule 49
```

```
Time 3
Pinhuobor 70
Necoi 62
Finron 27
```

Estrutura do código

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. A informação de cada duende deve ser armazenada em um tipo abstrato de dados criado especificamente para isso, contendo campos *nome*, *idade* e *escalado*. O último campo deste tipo abstrato de dados deve ser utilizado para determinar se o duende já foi escalado para um time ou não. Todas as informações dos duendes de um caso de teste devem ser armazenadas em um vetor alocado dinamicamente (e posteriormente liberado) para cada caso de teste.

Os dados de um time devem ser armazenados em um tipo abstrato de dados criado especificamente para isso, contendo campos *lider*, *entregador* e *piloto* para os nomes, além de campos *idadeLider*, *idadeEntregador* e *idadePiloto* para as idades. Os dados dos times devem ser armazenados em um vetor alocado dinamicamente (e posteriormente liberado) para cada caso de teste.

Note que não se deve utilizar algum algoritmo de ordenação para resolver este problema. Basta ir selecionando qual o próximo duende mais velho e seguir escalando os times.

Diretivas de Compilação

```
$ gcc duende.c -c
$ gcc principal.c -c
$ gcc duende.o principal.o -o programa
```