Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Estruturas de Dados 2019/1
Prof. Luiz Chaimowicz e Raquel Prates
Monitores: Matheus Nunes e Ronaldo Vieira

Trabalho Prático 1: Acesso ao Ensino Superior em Arendelle

Valor: 10 pontos
Data de entrega: 03 de Maio de 2019

Introdução

O Sistema de Seleção Unificado (SISU) foi criado pelo Ministério da Educação do Brasil em 2010, com o objetivo de unificar o processo seletivo de entrada para as universidades públicas brasileiras. Aberto a participantes do Exame Nacional do Ensino Médio (ENEM) do ano anterior, o SISU recebe milhões de inscrições todo ano, para milhares de vagas em instituições de ensino de nível superior distribuídas pelo país.[1]

Elsa, rainha de Arendelle, ficou maravilhada ao saber da existência do SISU, e gostaria de implementar um sistema parecido em seu reino. A infraestrutura de universidades públicas em Arendelle é muito parecida com a do Brasil, porém o sistema lá ainda é ligeiramente medieval, as inscrições dos alunos às universidades são feitas via pombo-correio, e este método tem se provado extremamente ineficiente. Elsa foi alertada ao fato de que os alunos de Estruturas de Dados da UFMG são programadores extremamente habilidosos, e portanto, resolveu pedir a sua ajuda para modernizar o sistema.

O objetivo deste trabalho é praticar os conceitos de **listas encadeadas e análise de complexidade**. Você deverá simular o processo de candidatura de alunos a vagas em cursos disponíveis, como uma versão simplificada do SISU, chamada de Mini SISU. Uma lista de cursos será fornecida, juntamente com seus respectivos números de vagas. Em seguida, uma lista de alunos será fornecida, juntamente à sua nota e suas duas opções de curso. O seu dever é alocar os alunos para cada curso, de forma que, ao final do processo de inscrição, os alunos tenham sua colocação garantida na primeira chamada ou na lista de espera de algum curso.

Detalhes do problema

Após reunião com Elsa, o Ministério da Educação de Arendelle encaminhou o seguinte e-mail explicando o funcionamento do Mini SISU:

> De: "Ministério da Educação" <comunicacao@mec.gov.ad>
> Assunto: Funcionamento do Mini SISU
>
> Caro(a) aluno(a) de Estrutura de Dados,
>
> Este projeto é de suma importância para democratizar o acesso ao ensino superior em Arendelle. Gostaríamos de construir um sistema em que os alunos, de posse de suas notas na última edição do ENEM, possam se candidatar a dois cursos **distintos** em ordem de preferência, assim como no SISU brasileiro. Em Arendelle as inscrições para o Mini SISU serão feitas em um único dia, então **cada aluno poderá escolher apenas uma vez suas opções de cursos**.
>
> Os cursos possuem números restritos de vagas, mas possuem **infinitas posições na lista de espera**. O aluno classificado em sua primeira opção não deverá ser colocado em nenhuma lista de espera, já um aluno classificado para sua segunda opção **deverá ser colocado**

[1] Mais informações sobre o SISU em `http://www.sisu.mec.gov.br/`

também na lista de espera da primeira opção. Um aluno que não for classificado para nenhuma de suas opções **deverá ser colocado na lista de espera de ambas**. Caso haja empate de nota entre alunos, o desempate deverá ser feito **primeiramente por ordem de opção** (o aluno que escolheu o curso como primeira opção ganhará preferência), e **segundamente por ordem de chegada**, ou seja, o aluno que se cadastrou primeiro no sistema ganhará a vaga (a lista de alunos que enviaremos estará ordenada por horário de cadastro).

Ao término da alocação de alunos aos cursos, gostaríamos de ver a **nota de corte de cada curso** e, é claro, **a lista dos alunos classificados e a lista de espera**, ambos com nome e nota do(a) aluno(a).

Contamos com você!

–
Ministério da Educação
Arendelle, de portas abertas para todos

Dadas as especificações fornecidas acima, sua tarefa é implementar o Mini SISU. A seção a seguir apresenta o formato desejado para a entrada e saída de dados.

Entrada e saída

A partir das observações ressaltadas no e-mail, o resumo do formato de entrada e saída desejados pelo MEC de Arendelle é:

Entrada. A primeira linha da entrada contém 2 inteiros N ($2 \leq N \leq 2^7$) e M ($1 \leq M \leq 2^{10}$), representando respectivamente o número de cursos e número de alunos a serem considerados na edição do Mini SISU.

Os N pares de linhas seguintes contém informações sobre os N cursos. A primeira linha de um par contém uma string S ($1 \leq |S| \leq 100$) representando o nome do curso, sem acentuação. A segunda linha do par contém um inteiro v ($0 \leq v \leq 2^{10}$), representando a quantidade de vagas disponíveis no curso. Os cursos são numerados de 0 a $N - 1$ de acordo com a ordem em que aparecem na entrada.

Após as informações dos cursos, os próximos M pares de linhas trazem as informações sobre os M alunos. A primeira linha de um par contém uma string S' ($1 \leq |S'| \leq 100$) representando o nome do(a) aluno(a), sem acentuação. A segunda linha do par contém um número decimal n ($0.0 \leq n \leq 1000.0$) seguido de dois inteiros p ($0 \leq p \leq N - 1$) e s ($0 \leq s \leq N - 1$), representando respectivamente a nota obtida pelo(a) aluno(a), sua primeira e sua segunda opção de curso. As opções de curso de um(a) aluno(a) serão sempre diferentes. Os alunos são numerados de 0 a $M - 1$ de acordo com a ordem em que aparecem na entrada.

Saída. Para cada curso, na mesma ordem da entrada, deverá ser impresso em uma mesma linha o nome do curso e a nota de corte (com duas casas decimais – **dica:** use '`%.2f`' no lugar de '`%f`' ao imprimir). Na próxima linha deverá ser impresso somente a string "Classificados". Em seguida deverão ser impressas C_i linhas, contendo o nome do aluno e sua nota, na ordem de classificação, sendo C_i o número de alunos classificados para o curso i. Ou seja, **os alunos deverão estar em ordem descendente de nota**, seguindo os critérios de desempate especificados anteriormente. Posteriormente deverá ser impressa a string "Lista de espera"em sua própria linha, seguida de E_i (número de alunos na fila de espera do curso i) linhas contendo os nomes e as notas dos alunos que estão na fila de espera do curso, também em ordem decrescente de nota.

Exemplo. A Tabela 1 apresenta o modelo e um exemplo de entrada e saída.

A entrada deve ser lida da entrada padrão (`stdin`) utilizando `scanf`, por exemplo, e a saída deve ser impressa na saída padrão (`stdout`), utilizando, por exemplo, a função `printf`. Durante a realização de seus testes, você pode querer ler a entrada de um arquivo (por exemplo, *file.in*) e escrever a saída

Entrada	**Entrada**

`(qtd. de cursos) (qtd. de alunos)`
`(nome do curso 0)`
`(qtd. de vagas do curso 0)`
`(nome do curso 1)`
`(qtd. de vagas do curso 1)`
`...`
`(nome do curso` $N-1$`)`
`(qtd. de vagas do curso` $N-1$`)`
`(nome do aluno 0)`
`(nota do aluno 0) (1ª opção) (2ª opção)`
`(nome do aluno 1)`
`(nota do aluno 1) (1ª opção) (2ª opção)`
`...`
`(nome do aluno` $M-1$`)`
`(nota do aluno` $M-1$`) (1ª opção) (2ª opção)`
	2 5
	Mineracao de Gelo
	2
	Engenharia Metalurgica
	1
	Olavo das Neves
	496.00 0 1
	Gothi Gelatto
	490.10 0 1
	Gerda Ferreira
	556.79 1 0
	Hans Westergaard
	418.09 1 0
	Kristoff Bjorgman
	725.66 0 1

Saída	**Saída**

`(nome do curso 0) (nota de corte)`
`Classificados`
`(nome do primeiro aluno) (nota do aluno)`
`(nome do segundo aluno) (nota do aluno)`
`...`
`(nome do último aluno) (nota do aluno)`
`Lista de espera`
`(nome do primeiro aluno) (nota do aluno)`
`(nome do segundo aluno) (nota do aluno)`
`...`
`(nome do último aluno) (nota do aluno)`
 	
...
 	
	Mineracao de Gelo 496.00
	Classificados
	Kristoff Bjorgman 725.66
	Olavo das Neves 496.00
	Lista de espera
	Gothi Gelatto 490.10
	Hans Westergaard 418.09
	
	Engenharia Metalurgica 556.79
	Classificados
	Gerda Ferreira 556.79
	Lista de espera
	Gothi Gelatto 490.10
	Hans Westergaard 418.09

Tabela 1: À esquerda, um modelo genérico de entrada e saída do problema. À direita, um exemplo concreto, com dois cursos e cinco alunos.

em um arquivo (por exemplo *file.out*). Nesse caso, você pode utilizar uma linha de comando como a seguir, que usa o arquivo *file.in* como a entrada padrão em vez do teclado, e usa o arquivo *file.out* como a saída padrão, em vez da tela:

```
./executavel < file.in > file.out
```

Entregáveis

Código-fonte. A implementação poderá ser feita utilizando as linguagens C ou C++. Não será permitido o uso da *Standard Library* do C++ ou de bibliotecas externas que implementem as estruturas de dados. **A implementação das estruturas utilizadas neste trabalho deve ser sua**. Os códigos devem ser executáveis em um computador com *Linux*. Caso você não possua um computador com este sistema, pedimos que teste a funcionalidade do seu trabalho em um dos computadores do laboratório de graduação do CRC[2]. Códigos redigidos como projetos do `CodeBlocks`[3] também serão aceitos.

[2] `https://crc.dcc.ufmg.br/infraestrutura/laboratorios/linux`
[3] `http://www.codeblocks.org/`

O código deve ser dividido em **pelo menos** três arquivos: ***main.c(.cpp)***, ***lista.c(.cpp)*** e ***lista.h***. O arquivo ***main.c(.cpp)*** deve conter as chamadas para as principais funções do seu programa. Os arquivos ***lista.c(.cpp)*** e ***lista.h*** devem conter, respectivamente, as implementações e as declarações das estruturas de dados utilizadas no trabalho.

A utilização de *Makefile*[4] é recomendada para este trabalho. Junto a esta especificação, serão postados exemplos de *Makefile* genéricos para *C* e *C++*.

Documentação. A documentação de seu programa deverá estar em formato PDF e ser **sucinta**. Ela deverá contemplar os seguintes tópicos:

- Introdução. Apresentação do problema abordado e visão geral sobre o funcionamento do programa.
- Implementação. Descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, o formato de entrada e saída de dados, compilador utilizado, bem como decisões tomadas relativas aos casos e detalhes que porventura estejam omissos no enunciado.
- Análise de complexidade. Estudo da complexidade de tempo e espaço do algoritmo desenvolvido utilizando o formalismo da notação assintótica.
- Conclusão. Resumo do trabalho realizado e eventuais dificuldades ou considerações sobre seu desenvolvimento.
- Bibliografia. Fontes consultadas para realização do trabalho, incluindo sites da Internet, se for o caso.

O código-fonte e a documentação devem ser submetidos em um único arquivo **'Nome_Sobrenome.zip'** no Moodle da disciplina até as **23:59** do dia **03 de Maio de 2019**.

Considerações Finais

Algumas considerações finais importantes são:

- O que será avaliado no trabalho:

  Boas práticas de programação: se o está código bem organizado e indentado, com comentários explicativos, possui variáveis com nomes intuitivos, modularizado, etc.

  Uso correto das estruturas de dados: se as estruturas de dados escolhidas são adequadas para o propósito do algoritmo e foram implementadas de forma correta. O uso correto de listas encadeadas será avaliado.

  Corretude do algoritmo: se o algoritmo resolve o problema corretamente.

  Conteúdo da documentação: se todo o conteúdo necessário está presente, reflete o que foi implementado e está escrito de forma coerente e coesa.

- Após submeter no Moodle seu arquivo **'.zip'**, faça o download dele e certifique-se que não está corrompido. Não será dada segunda chance de submissão para arquivos corrompidos.
- Em caso de dúvidas, pergunte no Fórum de Discussão no Moodle ou procure os monitores da disciplina.

[4] `https://opensource.com/article/18/8/what-how-makefile`

- **PLÁGIO É CRIME**: caso utilize códigos disponíveis online ou em livros, **referencie** (inclua comentários no código fonte e descreva a situação na documentação). Trabalhos onde o plágio for identificado serão devidamente penalizados: o aluno terá seu trabalho anulado e as devidas providências administrativas serão tomadas. **Não copie código do colega, isto também é plágio e as regras acima também se aplicam**.
- Comece o trabalho o mais cedo possível. A data de entrega está tão longe quanto poderia estar.

Bom trabalho!