

Instituto de Computação da UNICAMP

Disciplina MC102: Algoritmos e Programação de Computadores - Turmas EF

Primeiro Semestre de 2015

Laboratório N° 14. Peso 2.

Prazo de entrega: **29/05/2015 às 23:59:59**

PROFESSOR: Alexandre Xavier Falcão

MONITORES: João do Monte Gomes Duarte

MONITORES: Jadisha Yarif Ramírez Cornejo

MONITORES: Takeo Akabane

MONITORES: Eduardo Spagnol Rossi

MONITORES: Guilherme Augusto Sakai Yoshike

Censo Demográfico Brasileiro (2)

Neste exercício, você estenderá as funcionalidades do programa criado para o IBGE no laboratório 13, desenvolvendo duas novas operações de dados: **atualização** e **validação**. Sendo assim, o novo programa realizará as operações de cadastro, exclusão, listagem, atualização e validação sobre um conjunto de dados organizado nos seguintes campos: uf, município, número de homens, número de mulheres, percentual de homens e percentual de mulheres. A tabela seguinte ilustra como os dados estão estruturados.

UF	Município	População	Número de homens	Número de mulheres	Percentual de homens	Percentual de mulheres
RO	Ariquemes	90354	12670	11752	51.9	48.1
AC	Assis_Brasil	6075	3091	2984	50.9	49.1
SP	Campinas	1080999	521209	559790	48.2	51.8

Note que no campo município não é utilizado o caractere branco, sendo esse substituído pelo caractere "_". Ainda, para os campos percentual de homens e percentual de mulheres, o percentual é dado como um número real com uma casa decimal arredondado para o decimal mais próximo, i.e. o valor 10.666 é arredondado para 10.7, enquanto que 10.644 é arredondado para 10.6.

Você deverá considerar as seguintes especificações para o desenvolvimento das novas funcionalidades deste programa.

Especificações

Os requisitos das operações do laboratório 13 - cadastro, exclusão e listagem - bem como as operações a serem implementadas neste exercício - **atualização e validação** - são detalhados abaixo:

1. Cadastro:

- A primeira operação efetuada pelo programa será o cadastro dos dados de cidades.
- O programa poderá receber um ou mais registros (linhas da tabela) e inseri-los de uma única vez.
- A quantidade de itens que serão cadastrados será fornecida.
- O programa poderá receber outras listas de cadastros enquanto estiver em execução.
- **Entrada:**
 - Primeira linha: caractere '**c**' ou '**C**', para representar uma operação de cadastro, seguido de um espaço em branco e um inteiro **n** sem sinal que indique a quantidade de itens a serem inseridos.
 - Linhas subsequentes: **n** linhas de registros. Os campos de um registro serão separados por um espaço em branco.

2. Exclusão:

- O programa poderá excluir um ou mais registros de uma única vez.
- O programa receberá uma lista de combinações de estados e respectivas cidades (estado/cidade), que permitirão identificar quais registros serão excluídos.
- A quantidade de itens a serem excluídos será fornecida.
- **Entrada:**
 - Primeira linha: caractere '**e**' ou '**E**', para representar uma operação de exclusão, seguido de um espaço em branco e um inteiro **n** sem sinal que indique a quantidade de itens a serem removidos.
 - Linhas subsequentes: **n** linhas de combinações estado/cidade. A combinação será separada por um espaço em branco, assumindo a seguinte forma:
"estado cidade"

3. Listagem:

- A listagem consistirá na identificação e impressão de registros que atendam um determinado critério de consulta.
- O critério estabelecido consiste em identificar os registros que apresentam valores dos campos numéricos população, número de homens ou número de mulheres que estejam dentro de um intervalo determinado.
- O programa deverá considerar apenas um dos campos numéricos por vez. O campo a ser considerado será indicado ao programa através de um dado de entrada.
- O intervalo numérico que delimita a consulta de registros será fornecido. Os valores do campo indicado que forem maiores ou iguais ("**>=**") ao menor valor do intervalo **e** menores ou iguais ("**<=**") ao maior valor do intervalo, terão seus respectivos

registros impressos.

- A listagem deverá ser impressa na ordem original utilizada para os cadastros, ou seja, sempre deve-se cadastrar novos registros após todos os registros já cadastrados.
- **Entrada:**
 - Primeira linha: caractere '**l**' ou '**L**', para representar uma operação de listagem, seguido de um espaço em branco e um caractere que indique o campo numérico a ser considerado na consulta.
 - Ainda na primeira linha, depois do caractere que indica o campo numérico, considere um espaço em branco seguido de dois números inteiros sem sinal, separados por um espaço em branco, que representam o menor e o maior número do intervalo numérico que delimita a consulta.
 - Adote a seguinte convenção de caracteres para os campos numéricos: '**p**' ou '**P**' para população, '**h**' ou '**H**' para quantidade de homens e '**m**' ou '**M**' para quantidade de mulheres.

4. Atualização:

- O programa receberá uma lista de registros atualizados e a quantidade de itens à serem modificados.
- Cada registro fornecido será precedido pela combinação estado/cidade. Esta combinação permitirá a identificação do item correto, mesmo nos casos em que os campos estado ou cidade sejam atualizados.
- **Entrada:**
 - Primeira linha: caractere '**a**' ou '**A**', para representar uma operação de atualização, seguido de um espaço em branco e um inteiro **n** sem sinal que indique a quantidade de itens a serem modificados.
 - Linhas subsequentes: **2n** linhas, sendo que a cada 2 linhas, uma combinação estado/cidade (1ª linha) e o respectivo registro completo (2ª linha) são apresentados.
 - Os campos da combinação estado/cidade (1ª linha) e do respectivo registro (2ª linha) serão separados por um espaço em branco.

5. Validação:

- O programa deverá validar todos os registros afim de verificar valores inconsistentes para os campos percentual de homens e percentual de mulheres.
- A validação consiste em calcular a porcentagem de homens e mulheres de cada registro - baseado nos campos população, quantidade de homens e quantidade de mulheres - e verificar se as porcentagens apresentadas encontram-se corretas.
- Lembre-se que as porcentagens dadas são sempre números reais com uma casa decimal arredondado **para o decimal mais próximo**.
- Caso um registro contenha valores errados para pelo menos um dos campos de porcentagem, o programa deverá imprimir o respectivo registro.
- Os valores de porcentagens considerados inconsistentes são aqueles que possuem

variação de, pelo menos, **0.1** para **menos** ou para **mais** com relação ao valor correto.

- **Entrada:**

- Linha única: caractere 'v' ou 'V', para representar uma operação de validação.

A **saída** do programa se dará através das operações de listagem e validação e será caracterizada pela impressão de todos campos de registros. O programa poderá imprimir um ou mais registros, sendo que cada registro ocupará uma linha na lista impressa. Os campos de registros deverão ser separados por um espaço em branco. Imprima uma **linha vazia** após cada impressão de lista, de modo que as listas subsequentes (caso ocorram) não fiquem juntas e sejam distinguíveis.

O seu programa deve funcionar recebendo comandos e executando as respectivas funções, até que seja recebido como entrada o caractere 'f' ou 'F'. Caso o usuário entre com algum comando inválido, o programa deve imprimir a mensagem "Comando invalido. Entre com outro comando, ou 'F' para terminar.". Todas as mensagens impressas devem ser acompanhadas de uma quebra de linha.

Implementação

Para esse laboratório, você obrigatoriamente **deve** definir uma **estrutura** que encapsule todos os campos de dados que designam o censo: estado, município, população, número de homens, número de mulheres, porcentagem de homens e porcentagem de mulheres.

Você **deverá** criar uma **função** para cada operação de dados especificada acima: cadastro, exclusão, listagem, atualização e validação.

Observações e Dicas

- O programa poderá receber, no máximo, **1000** registros. Utilize um vetor para armazenar as estruturas.
- Os campos uf e municipio podem ser armazenados em vetores de **3** e **100** posições, respectivamente.
- Utilize uma variável contadora para o número de registros mantidos em memória. Estruturas novas, cadastradas durante a execução do programa, deverão ser inseridas a partir do último registro cadastrado.
- Não haverá espaço em branco em nomes compostos de municípios.
- Cada combinação "estado cidade" é estritamente **única**. Não haverá combinações repetidas.
- As porcentagens devem ser impressas com apenas **um dígito** na parte decimal.
- Na operação de validação, **evite** estabelecer comparações diretas entre variáveis de tipo

float, utilizadas para o armazenamento de porcentagens. As porcentagens carregadas do arquivo e as porcentagens calculadas pelo programa provavelmente terão partes decimais **diferentes**, fazendo com que os valores sejam sempre diferentes, mesmo nos casos em que as porcentagens estejam muito próximas.

- Na operação de exclusão e atualização, você pode supor que as chaves fornecidas ("estado cidade") sempre existirão na base de dados.
- As funções implementadas no laboratório 13 serão igualmente **utilizadas** neste laboratório. Você deve garantir que estas funções estejam adequadamente implementadas.
- Você **deve** quebrar a linha após imprimir o resultado.
- Ao usar a função scanf para ler um caractere (com a máscara %c), deve-se acrescentar um espaço antes de %c para garantir que somente um caractere válido seja lido (ou seja, para ignorar espaços e quebras de linhas).
 - Por exemplo, para um char x, o comando scanf("%c",&x); assinalaria para x o caractere de espaço ' ' para a entrada " A " (note o espaço na entrada!) enquanto que scanf(" %c",&x); (note o espaço antes de %c) assinalaria 'A' para a variável x.

Exemplos de execução

Exemplo 1:

```
c 10
BA Itapetinga 68314 33480 34834 51.0 49.0
RO Espigao_D'Oeste 28741 14633 14108 50.9 49.1
MS Santa_Rita_do_Pardo 7254 3805 3449 52.5 47.5
MT Cuiaba 551350 269397 281953 48.9 51.1
GO Anapolis 335032 163467 171565 48.8 51.2
DF Brasilia 2562963 1225237 1337726 45.8 54.2
RS Porto_Alegre 1409939 654022 755917 46.4 53.6
SC Sao_Martinho 3211 1661 1550 51.7 48.3
SP Sao_Paulo 11244369 5323385 5920984 47.3 52.7
RJ Rio_de_Janeiro 6323037 2960954 3362083 53.2 46.8
v
a 4
BA Itapetinga
BA Itapetinga 68314 33480 34834 49.0 51.0
RO Espigao_D'Oeste
RO Espigao_D'Oeste 28741 14633 14108 50.9 49.1
DF Brasilia
DF Brasilia 2562963 1225237 1337726 47.8 52.2
RJ Rio_de_Janeiro
RJ Rio_de_Janeiro 6323037 2960954 3362083 46.8 53.2
```

l p 28000 15000000

f

BA Itapetinga 68314 33480 34834 51.0 49.0

DF Brasilia 2562963 1225237 1337726 45.8 54.2

RJ Rio_de_Janeiro 6323037 2960954 3362083 53.2 46.8

BA Itapetinga 68314 33480 34834 49.0 51.0

RO Espigao_D_Oeste 28741 14633 14108 50.9 49.1

MT Cuiaba 551350 269397 281953 48.9 51.1

GO Anapolis 335032 163467 171565 48.8 51.2

DF Brasilia 2562963 1225237 1337726 47.8 52.2

RS Porto_Alegre 1409939 654022 755917 46.4 53.6

SP Sao_Paulo 11244369 5323385 5920984 47.3 52.7

RJ Rio_de_Janeiro 6323037 2960954 3362083 46.8 53.2

Exemplo 2:

c 5

PR Paranaipoema 2791 1383 1408 49.6 50.4

RO Alta_Floresta_D'Oeste 24422 12670 11752 55.9 48.1

MG Fama 2348 1179 1169 50.2 49.8

SP Buritama 15418 7732 7686 50.1 49.9

BA Sao_Goncalo_dos_Campos 33289 16032 17257 48.2 51.8

l p 10 100000

c 3

AC Tarauaca 35526 18317 17209 51.6 48.4

AM Anama 10193 5434 4759 53.3 46.7

PA Sapucaia 5047 2710 2337 53.0 46.0

v

a 2

RO Alta_Floresta_D'Oeste

RO Alta_Floresta_D'Oeste 24422 12670 11752 51.9 48.1

PA Sapucaia

PA Sapucaia 5047 2710 2337 53.7 46.3

l h 10 50000

e 1

AC Tarauaca

l m 10 50000

PR Paranapoema 2791 1383 1408 49.6 50.4
RO Alta_Floresta_D'Oeste 24422 12670 11752 55.9 48.1
MG Fama 2348 1179 1169 50.2 49.8
SP Buritama 15418 7732 7686 50.1 49.9
BA Sao_Goncalo_dos_Campos 33289 16032 17257 48.2 51.8

RO Alta_Floresta_D'Oeste 24422 12670 11752 55.9 48.1
PA Sapucaia 5047 2710 2337 53.0 46.0

PR Paranapoema 2791 1383 1408 49.6 50.4
RO Alta_Floresta_D'Oeste 24422 12670 11752 51.9 48.1
MG Fama 2348 1179 1169 50.2 49.8
SP Buritama 15418 7732 7686 50.1 49.9
BA Sao_Goncalo_dos_Campos 33289 16032 17257 48.2 51.8
AC Tarauaca 35526 18317 17209 51.6 48.4
AM Anama 10193 5434 4759 53.3 46.7
PA Sapucaia 5047 2710 2337 53.7 46.3

PR Paranapoema 2791 1383 1408 49.6 50.4
RO Alta_Floresta_D'Oeste 24422 12670 11752 51.9 48.1
MG Fama 2348 1179 1169 50.2 49.8
SP Buritama 15418 7732 7686 50.1 49.9
BA Sao_Goncalo_dos_Campos 33289 16032 17257 48.2 51.8
AM Anama 10193 5434 4759 53.3 46.7
PA Sapucaia 5047 2710 2337 53.7 46.3

Notas: Textos em azul denotam dados de entrada do programa.

Textos em vermelho denotam dados de saída do programa.

Observações gerais:

- O número máximo de submissões é 15;
- O seu programa deve estar completamente contido em um único arquivo denominado censo2.c;
- Para a realização dos testes automáticos, a compilação se dará da seguinte forma:
gcc censo2.c -o censo2 -Wall -std=c99 -pedantic;
- Não se esqueça de incluir no início do programa uma breve descrição dos objetivos, da entrada, da saída, seu nome e RA;

- Após cada submissão, você deve aguardar um minuto até poder submeter seu trabalho novamente.
-

Critérios importantes:

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará nota zero nesta tarefa de laboratório.

- Você deve utilizar obrigatoriamente o header `stdio.h`. Adicionalmente, é permitido o uso do header `string.h`, caso julgue necessário.
- Você deve necessariamente especificar e implementar as funções e estruturas requisitadas na seção **Implementação**.