

Capivaras

Trabalho 1

Algoritmos e Programação II

1 Descrição

Achadas em certas áreas das Américas do Sul e Central, próximo a rios e lagos, a **capivara** (*Hydrochoerus hydrochaeris*), também chamada de carpincho e capincho, é o maior roedor do mundo. Quando a esquadra de Pedro Álvares Cabral chegou ao Brasil em 1500, os indígenas locais já domesticavam este animal. Alimenta-se de capins e ervas, daí, a etimologia de seu nome: capivara procede do termo tupi *kapi'wara*, que significa “comedor de capim”. Já capincho vem do castelhano platino *capincho*. No Rio Grande do Sul, é também conhecida por capinga.



Não é novidade que capivaras adoram o gramado da FCOM¹. Seja por seu gramado sempre verde, pela variedade de ervas ou simplesmente pelo *Wi-Fi*, é comum vê-las se deleitando à frente deste distinto prédio. Sua visita é tão frequente que Nalvo, diretor da FCOM de 2009 a 2016, nomeou-as carinhosamente, uma a uma. Ele inclusive planeja matriculá-las em Algoritmos e Programação I. Preocupado com sua alimentação, prof. Nalvo as tem pesado mensalmente, agrupando registros por famílias. Variações acentuadas no peso, tanto para mais quanto para menos, podem indicar uma alimentação inadequada. Contudo, com o passar do tempo e o crescimento das famílias, a análise desses registros tem se tornado cada vez mais trabalhosa, por isso ele pediu nossa ajuda.

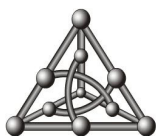
Sua tarefa neste trabalho é escrever um programa que receba o nome de uma família, o nome das capivaras com sua data de nascimento, os registros mensais e exiba uma classificação referente às variações de peso durante o período total registrado.

Ao classificar as capivaras, utilizaremos as seguintes regras, nesta ordem:

1. Maior variação de peso ao longo de todos os meses (para mais ou para menos)²
2. Maior variação de peso em um único mês (para mais ou para menos);
3. Maior idade;
4. Ordem lexicográfica dos nomes das capivaras.

¹Faculdade de Computação/Universidade Federal de Mato Grosso do Sul

²Considere o *valor absoluto* ou *módulo* das variações ao ordenar



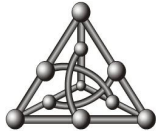
Essas regras gerais estabelecem critérios de classificação que sempre fornecem uma ordem decrescente das variações de peso das capivaras de uma família (daquela que tem a maior variação para aquela que tem a menor), estabelecendo também critérios de desempate. Isso significa que, após uma quantidade de meses, nunca há duas capivaras “empatadas”, mesmo que tenham a mesma variação total de peso.

2 Entrada e saída

A entrada contém vários casos de teste, cada um representando os registros mensais de uma família de capivaras. A primeira linha contém um número inteiro $f > 0$ que se refere ao número de famílias. Após esse número, uma linha em branco é apresentada, e a seguir, são apresentadas as informações das f famílias, em sequência. Cada família tem como entrada primeiramente seu nome, seguido de uma linha em branco. Depois, o número n de capivaras na família (um número inteiro com $1 \leq n \leq 50$) é dado em uma linha. Em cada uma das n linhas seguintes, um nome de uma capivara é fornecido, separado por uma vírgula e um espaço de sua data de nascimento, no formato DD/MM/AAAA. Em seguida, uma linha em branco é dada. Em seguida um número inteiro m , com $2 \leq m \leq 12$, é fornecido, indicando o número total de meses registrados. Para cada mês, um nome descritivo é fornecido em uma linha. Em seguida, o peso registrado naquele mês para cada uma das n capivaras é apresentado linha a linha em uma ordem qualquer, onde cada linha contém o nome de uma capivara separado por uma vírgula e um espaço de seu peso. Um peso é um número inteiro que representa uma massa em hectogramas³. Os conjuntos de registros mensais são dados em ordem cronológica e cada par deles é separado por uma linha em branco. Todos os nomes deste trabalho têm no máximo 50 caracteres.

Para cada um dos casos de teste, seu programa deve imprimir o seguinte. Primeiro, uma linha contendo o nome da família deve ser impressa, separado por espaço, traço e espaço de uma mensagem que descreve as variações de peso após os m meses registrados. Então, imprima n linhas, onde cada uma contém um número inteiro indicando a variação total de peso de uma capivara, seguida do nome da capivara. As linhas devem estar em ordem decrescente, daquela com maior variação de peso ao longo de todos os meses (para mais ou para menos) para aquela com a menor variação. Uma variação de peso deve ser impressa obrigatoriamente com sinal (+ ou -) e ocupar ao menos 4 casas, já incluindo o sinal (vide exemplo de saída). Siga as regras de classificação e desempate descritas na seção 1. Entre duas famílias sempre há uma linha em branco impressa.

³1 hectograma = 0.1 ou 10^{-1} quilograma, utilizamos pois é difícil comparar com exatidão números reais (*float*)



3 Exemplo de entrada

```
2

Capitubbies

4
Tinky Winky, 04/10/2004
Dipsy, 02/12/2008
Laa-Laa, 16/03/2005
Po, 07/03/2001

2

Primeiro mes
Tinky Winky, 513
Laa-Laa, 650
Po, 300
Dipsy, 630

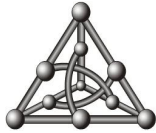
Segundo mes
Po, 325
Laa-Laa, 625
Tinky Winky, 501
Dipsy, 684

Caixao

8
Toto, 12/08/2001
Grampola, 12/08/2001
Salim, 19/08/2001
Cem Gramas, 31/12/2002
Bussunda, 20/07/2009
MacGyver, 24/07/2010
Goku, 09/10/2012
Astroncio, 12/08/2001

3

Fevereiro
Salim, 551
Toto, 772
Grampola, 457
Bussunda, 320
Cem Gramas, 801
MacGyver, 331
Astroncio, 520
Goku, 95
continua na próxima página...
```



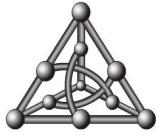
```
Marco
Astroncio, 520
Bussunda, 320
Goku, 120
Salim, 572
Toto, 772
Cem Gramas, 802
MacGyver, 321
Grampola, 460
```

```
Abril
Bussunda, 311
Salim, 563
Toto, 772
Goku, 135
Cem Gramas, 803
Astroncio, 520
Grampola, 455
MacGyver, 223
```

4 Exemplo de saída

```
Familia Capitubbies - variacoes apos 2 meses
+054, Dipsy
+025, Po
-025, Laa-Laa
-012, Tinky Winky

Familia Caixao - variacoes apos 3 meses
-108, MacGyver
+040, Goku
+012, Salim
-009, Bussunda
-002, Grampola
+002, Cem Gramas
+000, Astroncio
+000, Toto
```



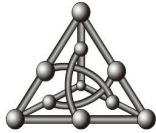
5 Exigências

Você **DEVE** usar a seguinte estrutura de dados em seu trabalho:

```
/* Armazena informacoes de uma capivara */
typedef struct {
    char nome[MAX_NOME+1];
    int ultimo_peso;          /* Ultimo peso registrado */
    int var_total;           /* Variacao total de peso */
    int max_var_mensal;      /* Maior variacao mensal encontrada */
    char nascimento[11];    /* Data de nascimento */
} capivara;

/* Armazena informacoes de uma familia de capivaras */
typedef struct {
    char nome[MAX_NOME+1];
    int num_capis;           /* Quantidade de capivaras na familia */
    int num_meses;          /* Meses registrados */
    capivara capis[MAX_CAPIVARAS]; /* Armazena as capivaras */
} familia;
```

Além disso, como você pôde perceber pelo enunciado do trabalho, em alguns pontos há necessidade de rearranjar um conjunto de dados em ordem crescente ou decrescente. Você **DEVE** usar um método de ordenação não-elementar para realizar essas tarefas, tais como ordenação por intercalação ou ordenação por separação.



6 Entrega

Instruções para entrega do seu trabalho:

1. Cabeçalho

Seu trabalho deve ter um cabeçalho com o seguinte formato:

```
/******  
*  
* Nome do(a) estudante  
* Trabalho 1  
* Professor(a): Nome do(a) professor(a)  
*  
*/
```

2. Compilador

Os(as) professores(as) usam o compilador da linguagem C da coleção de compiladores GNU `gcc`, com as opções de compilação `-Wall -std=c99 -pedantic` para corrigir os programas. Se você usar algum outro compilador para desenvolver seu programa, antes de entregá-lo verifique se o seu programa tem extensão `.c`, compila sem mensagens de alerta e executa corretamente.

3. Forma de entrega

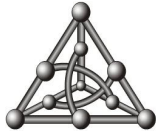
A entrega será realizada diretamente na plataforma de ensino à distância ([EAD/Moodle](#)), na disciplina de Algoritmos e Programação II. Um fórum de discussão deste trabalho já se encontra aberto. Para entrega do trabalho, você deve estar cadastrado na página <http://ead.facom.ufms.br> na disciplina Algoritmos e Programação II da turma que você está matriculado. Após abrir uma sessão digitando seu *login* e sua senha, vá até o tópico “Trabalhos”, e escolha “T1 - Entrega”. Você pode entregar o trabalho quantas vezes quiser até às **06 horas e 00 minutos** do dia **6 de fevereiro de 2017**. A última versão entregue é aquela que será corrigida. Encerrado o prazo, não serão mais aceitos trabalhos.

4. Atrasos

Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

5. Erros

Trabalhos com erros de compilação receberão nota **ZERO**. Faça todos os testes necessários para garantir que seu programa está livre de erros de compilação.



6. O que entregar?

Você deve entregar um único arquivo contendo **APENAS** o seu programa fonte com o mesmo nome de seu login no moodle, como por exemplo, `fulano_silva.c`. **NÃO** entregue qualquer outro arquivo, tal como o programa executável, já compilado.

7. Verificação dos dados de entrada

Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10 e o usuário digita um número negativo, uma letra, um cifrão, etc, o seu programa pode fazer qualquer coisa, como travar o computador ou encerrar a sua execução abruptamente com respostas erradas.

8. Arquivo com o programa fonte

Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso. Não esqueça da documentação de seu programa e de suas funções.

Dê o nome do seu usuário do moodle para seu programa e adicione a extensão `.c` a este arquivo. Por exemplo, `fulano_silva.c` é um nome válido.

9. Conduta Ética

O trabalho deve ser feito **INDIVIDUALMENTE**. Cada estudante tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho em grupo e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir idéias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas **NÃO** copie o programa!

Trabalhos considerados plagiados terão nota **ZERO**.