

# Prova-03

Prof. Dr. Gustavo Teodoro Laureano  
Prof. Dr. Thierson Rosa Couto

## Sumário

<b>1</b>	<b>Estatística Empresarial (++)</b>	<b>2</b>
<b>2</b>	<b>Rede de Contatos no Twitter (+++)</b>	<b>3</b>

# 1 Estatística Empresarial (++)



(++)

Uma empresa deseja obter informações estatísticas sobre seus funcionários. Para isso, irá colher os seguintes dados dos funcionários: matrícula, idade, número de filhos, sexo e salário.

Essas informações precisam ser armazenadas em uma estrutura:

```
1 typedef struct {
2     int matricula;
3     int idade;
4     int numFilhos;
5     char sexo;
6     double salario;
7 } FUNCIONARIO;
```

## Entrada

Na primeira linha ha um inteiro  $n$ ,  $1 < n < 500$ , representando a quantidade de funcionário (fazer alocação dinâmica). A seguir haverá  $n$  linhas com  $n$  os seguintes dados separados por um espaço em branco cada: matrícula, idade, número de filhos, sexo e salário.

## Saída

Deverá imprimir 4 (quatro) respostas:

- Quantidade de funcionários com idade superior a média de idades E salário superior a 3 salários mínimos.
- Quantidade de mulheres que possuem quantidade de filhos acima da média geral.
- Quantidade de homens que possuem quantidade de filhos acima da média geral.
- Quantidade de funcionários maiores de 47 anos com renda per-capita (por pessoa) abaixo de 2 salários mínimos.

\* considere o salário mínimo igual a 1200.00.

## Exemplo

Entrada	Saída
10 101 44 4 M 7001.00 105 56 2 F 2950.00 211 60 2 F 6870.00 221 25 1 F 9200.00 231 38 3 M 4350.00 300 70 4 M 2100.00 545 27 0 F 4500.00 654 65 1 F 2900.00 670 53 2 M 3300.00 888 55 2 F 4100.00	2 0 3 6

## 2 Rede de Contatos no Twitter (+++)



(+++)

Um determinado pesquisador precisa representar em uma tabela informações sobre relacionamentos entre um conjunto  $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$  de usuários do Twitter. Para isso, ele quer guardar para cada par  $(u_1, u_2), u_1, u_2 \in \mathcal{U}$  as seguintes informações, quando existirem:

- O número de *likes* que  $u_1$  fez em *tweets* escritos por  $u_2$ .
- O número de *retweets* que  $u_1$  fez em *tweets* escritos por  $u_2$ .
- O número de *menções* que  $u_1$  fez em *tweets* escritos por  $u_2$ .

Para isso, o pesquisador quer criar uma matriz quadrada  $n \times n$  que permita armazenar para cada par  $u_i, u_j, 1 \leq i, j \leq n$  as informações acima caso elas existam. Ele pensou em criar uma matriz de *structs* onde cada *struct* tem três campos correspondentes aos totais para cada um dos números de interações descritos acima. O problema dessa representação é que o pesquisador sabe que para um grande número de pares de usuários não existe nenhum tipo de interação entre eles, ou seja, os totais para os três tipos de interação seriam iguais a zero. Além disso, há vários casos em que  $u_1$  interage com  $u_2$ , mas  $u_2$  não interage com  $u_1$ . Armazenar todas as  $n \times n$  *structs* na tabela é, portanto, um desperdício de memória e  $n$  pode ser muito grande impossibilitando o programa de funcionar. O pesquisador quer que você faça um programa em que a tabela  $n \times n$  seja uma tabela de ponteiro para as *structs* que possuem os três campos comentados anteriormente e que aloque as *structs* sob demanda, somente quando forem necessárias. O programa deve ler os dados dos relacionamentos entre usuários e imprimir para cada usuário o total de *likes*, o total de *retweets* e o total de *menções* que ele fez a *tweets* de outros usuários. Além disso, é importante também saber a quantidade de *slots* vazios na tabela os valores médios de *likes*, *retweets* e *menções* por usuário.

### Entrada

A primeira linha da entrada é constituída por um único inteiro positivo  $N (N \leq 1000)$ , o qual corresponde à dimensão da matriz de relacionamentos. A segunda linha contém outro número inteiro  $M, 1 \leq M \leq N^2$  que corresponde ao número de pares não nulos da matriz. Em seguida há  $M$  linhas, cada uma com o seguinte formato:  $u_1 \ u_2 \ \text{num\_likes} \ \text{num\_retweets} \ \text{num\_menções}$ , onde os campos estão separados entre si por um espaço. O primeiro campo  $u_1$  corresponde ao usuário que fez as interações. O segundo campo ( $u_2$ ) corresponde ao usuário que recebe as interações. Os demais três campos correspondem a, respectivamente, o número de likes, o número de retweets e o número de menções que  $u_1$  fez em *tweets* de  $u_2$ .

### Saída

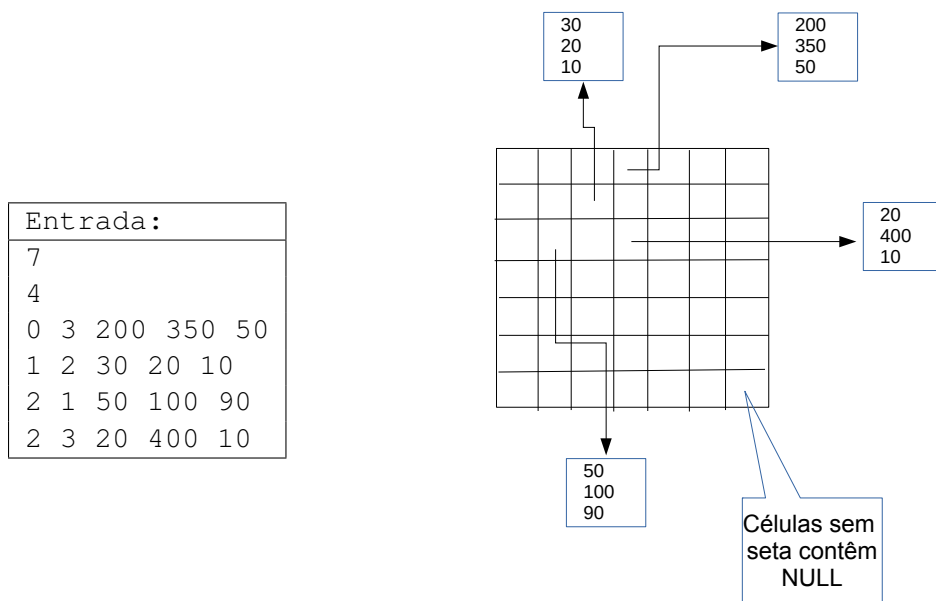
O programa deve imprimir uma linha para cada usuário contendo as seguintes informações: Usuário  $u_i$  - num. likes feitos:  $x$ , num. retweets feitos:  $y$  e num. mencoes feitos:  $z$ . Os valores de  $x, y$  e  $z$  correspondem aos totais de likes, retweets e menções, respectivamente, feitos pelo usuário  $u_i$ . Os usuários devem ser impressos na ordem crescente do seu número (ordem das linhas da tabela).

### Observações

1. Os compiladores da linguagem C não inicializam os campos de uma tabela. Você deve garantir que antes da leitura todos as células da tabela tenham NULL como valor de ponteiro.
2. Ao final do programa, todo espaço alocado dinamicamente deve ser liberado
3. Uso da função `malloc()` :  $x = (\text{Tipo\_do\_x}^*) \text{malloc}(\text{Num\_bytes\_tipo\_x} * \text{quantidade\_elem})$ . A função `malloc()` retorna NULL caso não consiga alocar espaço. Seu programa deve verificar se foi possível alocar espaço e terminar o programa em caso contrário.
4. Deve ser feito `#include<stdlib.h>` para usar a função `malloc()`.

5. A função `exit(1)` encerra o programa.
6. A função `free(x)` libera a área de memória cujo endereço está em `x`.
7. dada uma matriz `mat`, para acessar o campo `num_likes` da struct cujo endereço está em `mat[i][j]`: `(*mat[i][j]).num_likes`. Exemplo `scanf("%d", &((*mat[i][j]).num_likes))`.

### Exemplo



Saída:	
Usuario 0 - num.	likes: 200, num. retweets: 350 e num. mencoes: 50
Usuario 1 - num.	likes: 30, num. retweets: 20 e num. mencoes: 10
Usuario 2 - num.	likes: 70, num. retweets: 500 e num. mencoes: 100
Slots vazios: 45	
Media de likes por usuario: 100.00	
Media de retweets por usuario: 290.00	
Media de mencoes por usuario: 53.33	