

Universidade Federal de Goiás  
Curso de Ciência da Computação  
Introdução à Programação- 2021-2  
Prova P5

Prof. Thierson Couto Rosa

Atenção

A detecção de plágio na solução de alguma questão ou em parte da solução implicará em pontuação zero na questão para todos os envolvidos.

## **Sumário**

<b>1</b>	<b>Valor da Compra - 5,0 pontos</b>	<b>2</b>
<b>2</b>	<b>Premiação Ao Mais Antigo - 5,0 pontos</b>	<b>4</b>
<b>3</b>	<b>Ordenação com Ponteiros - bonus - 4,0 pontos</b>	<b>6</b>

# 1 Valor da Compra - 5,0 pontos

Uma loja possui uma lista com dados das mercadorias que são por ela vendidas. Cada mercadoria possui os campos:

- código da mercadoria (int),
- preço da mercadoria (float).
- nome da mercadoria (cadeia com no máximo 30 caracteres),

Escreva um programa que leia a lista de mercadorias e depois leia registros compras com as seguintes informações:

- nome do cliente (cadeia com no máximo 50 caracteres),
- o código da mercadoria a ser comprada(int), e
- quantidade da mercadoria a ser comprada.

Em seguida, o programa deve computar o valor total da compra e imprimir uma listagem com as seguintes informações: nome do cliente, nome da mercadoria, quantidade da mercadoria a ser comprada e o valor total da compra.

## Entrada

A primeira linha da entrada contém o número  $n$  ( $1 \leq n \leq 20$ ) de mercadorias que formam a lista de mercadorias da loja. Em seguida, para cada mercadoria há 3 linhas contendo as três informações correspondentes aos campos de uma mercadoria. A próxima linha contém um valor inteiro que corresponde ao número  $m$  ( $1 \leq m \leq 100$ ) de clientes da loja. Em seguida, para cada cliente há 3 linhas, com os três dados de um cliente.

## Saída

Para cada cliente a saída possui várias linhas com os formatos mostrados a seguir:

- Linha 1: “Compra do Cliente:  $c$ ”, onde  $c$  é um número inteiro indicando a ordem do cliente na entrada.
- Linha 2: “Nome do cliente:  $nc$ ”, onde  $nc$  é uma string contendo o nome do cliente.
- Linha 3: “Nome da mercadoria:  $nm$ ”, onde  $nm$  é o nome da mercadoria que o cliente está comprando.
- Linha 4: “Quantidade:  $q$ ”, onde  $q$  corresponde à quantidade da mercadoria que o cliente está comprando.
- Linha 5: “Valor da Mercadoria:  $v$ ”, onde  $v$  é o preço da mercadoria.
- Linha 6: “Valor total da compra:  $vt$ ”, onde  $vt$  é o valor total da compra.
- Linha 7: linha em branco - utilizada para separar um pedido de cliente de outro pedido.

## Exemplo

Entrada
4 298 234.5 Cuba Inox 123 150.00 Torneira Inox 452 132.00 Pia para Cozinha Vaza Pouco 341 120.00 Vaso Sanitario Deluxe 3 Joao Luiz Venancio 298 5 Paula Lima de Souza 452 4 Maria Luiza de Assis 123 6
Saída
Pedido de Compra do Cliente: 1 Nome do cliente: Joao Luiz Venancio Nome da mercadoria: Cuba Inox Quantidade: 5 Valor da Mercadoria: 234.50 Valor total da Compra: 1172.50  Pedido de Compra do Cliente: 2 Nome do cliente: Paula Lima de Souza Nome da mercadoria: Pia para Cozinha Vaza Pouco Quantidade: 4 Valor da Mercadoria: 132.00 Valor total da Compra: 528.00  Pedido de Compra do Cliente: 3 Nome do cliente: Maria Luiza de Assis Nome da mercadoria: Torneira Inox Quantidade: 6 Valor da Mercadoria: 150.00 Valor total da Compra: 900.00

## 2 Premiação Ao Mais Antigo - 5,0 pontos

Estamos chegando ao fim do ano, na época de Natal, e uma empresa resolveu prover uma série de premiações aos seus funcionários. Uma delas é a premiação aos três funcionários mais antigos da empresa. A empresa contratou você para escrever um programa que leia os dados dos funcionários da empresa e que os ordene em uma ordem tal que o funcionário mais antigo apareça em primeiro lugar, o segundo funcionário mais antigo apareça em segundo lugar, e assim por diante.

Os dados de cada funcionário fornecidos a você são:

- o nome do funcionário (no máximo 50 caracteres seguidos por uma quebra de linha);
- o dia, o mês e o ano da contratação (três números inteiros separado entre si por um espaço e seguidos por uma quebra de linha);
- uma linha com no máximo 2000 caracteres contendo um breve histórico das ocupações desempenhadas pelo funcionário na empresa;
- o valor do salário atual do funcionário (um valor double seguido por uma quebra de linha);

Escreva um programa que leia os dados de vários funcionários da empresa, ordene os funcionários e imprima os dados dos três funcionários mais antigos por ordem de antiguidade na empresa.

### Entrada

A primeira linha da entrada contém o número  $N$ ,  $0 < N \leq 200$ , de funcionários da empresa. Em seguida, para cada um dos  $N$  funcionários, há quatro linhas, cada uma contendo um dado de um funcionário, conforme descrito acima.

### Saída

A saída contém os dados dos três funcionários, porém mostrados em ordem decrescente de antiguidade (o mais antigo primeiro). Para cada usuário, deve haver as seguintes linhas na saída:

Nome:  $n$

Data de contratacao:  $dd/mm/aa$

Ocupacoes:  $o$

Salario:  $s$

onde  $n$  é o nome do funcionário,  $dd$ ,  $mm$  e  $aa$  são, respectivamente, o dia, o mês e o ano em que o funcionário foi contratado,  $o$  é uma string contendo uma lista de ocupações desempenhadas pelo funcionário na empresa e  $s$  é o salário atual do funcionário. Separando as linhas referentes a um funcionário daquelas referentes ao próximo funcionário deve haver uma linha em branco.

### Sugestão

Primeiramente escreva código somente para ler os dados e para mostra-los na tela para você se certificar que o programa está lendo corretamente antes de escrever código para a ordenação. A leitura da entrada é um pouco complicada pois há duas strings a serem lidas. Lembre-se de consumir os caracteres de quebra de linha que ficam no buffer do teclado antes de ler uma string.

## Exemplo

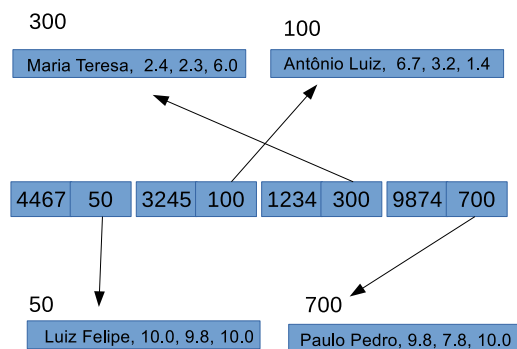
Entrada
5 Antonio Vasconcelos Rogrigues 3 5 1978 Auxliar de escritorio, secretario, gerente administrativo 10000.00 Bruno Oliveira 12 12 2015 Servicos gerais 1000.00 Antonieta Pires 3 4 1968 Secretaria, gerente administrativa 20000.00 Maria Luiza de Souza 3 9 1975 Secretaria 12000.00 Leonardo Taveira 3 4 1980 gerente de RH 20000.00
Saída
Nome: Antonieta Pires Data de contratacao: 3/4/1968 Ocupacoes: Secretaria, gerente administrativa Salario: 20000.00  Nome: Maria Luiza de Souza Data de contratacao: 3/9/1975 Ocupacoes: Secretaria Salario: 12000.00  Nome: Antonio Vasconcelos Rogrigues Data de contratacao: 3/5/1978 Ocupacoes: Auxliar de escritorio, secretario, gerente administrativo Salario: 10000.00

### 3 Ordenação com Ponteiros - bonus - 4,0 pontos

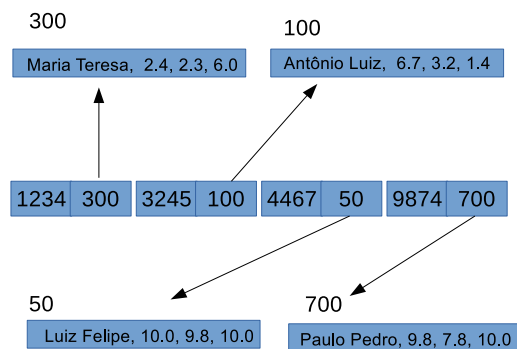
O problema de ordenar uma sequência de registros de dados (structs) segundo um determinado campo desse registro (denominado *chave de ordenação* ou simplesmente *chave*) é comum em várias aplicações. Entretanto, os algoritmos de ordenação podem envolver trocas de registros que estão em posições diferentes na sequência original, com o objetivo de ordená-los na sequência correta (crescente ou decrescente). Se o tamanho do registro é muito grande, a troca de registros entre uma posição e outra na memória pode tornar o processo de ordenação lento, pois registros envolvendo vários bytes devem ser copiados de uma região para outra na memória. Uma alternativa é separar a chave do restante do registro e criar um novo registro **par** que contenha apenas dois campos: a chave e um ponteiro para o restante dos dados. Cria-se um vetor cujos elementos são *pares*. A ordenação é feita somente com os pares. Como cada par tem um endereço para o restante dos dados originais, a ordenação envolve a troca apenas dos pares e os demais dados mantêm suas posições na memória. A ilustração abaixo mostra essa representação para os dados de entrada.

#### Ilustração

Considere os seguintes dados sobre alunos: matrícula, nome, três notas. Considere que temos quatro alunos e que o campo chave usado na ordenação será a matrícula. Suponha ainda, os seguintes dados na entrada: {44677, Luiz Felipe, 10.0, 9.8, 10.0}, {3245, Antônio Luiz, 6.7, 3.2, 1.4}, {1234, Maria Teresa, 2.4, 2.3, 6.0} {9867, Paulo Pedro, 9.8, 7.8, 10.0}. Esses dados poderiam ser representados como mostra a figura abaixo.



Após a ordenação, a configuração passa a ser:



Você deve escrever um programa que utilize estruturas de dados que permitam a representação ilustrada nas figuras acima. O programa deve ter uma função para ler um caso de teste, uma função para ordenar os pares em ordem crescente por matrícula e uma função para trocar os pares de posição entre si, durante a execução da função de ordenação. A função de ordenação deve ordenar os dados de forma que apenas os structs no vetor sejam trocados entre si durante a ordenação em **ordem crescente**.

### Entrada

A entrada é um arquivo binário de nome "entrada.dat"(aspas não inclusas). O primeiro dado no arquivo é um número inteiro que indica o número de casos de teste. Cada caso de teste contém um número inteiro  $n$  indicando o tamanho do conjunto de dados. A seguir, aparecem três dados para cada elemento conjunto: o primeiro dado corresponde à matrícula de um aluno (int), o segundo, corresponde ao nome do aluno (no máximo 200 caracteres) e o terceira contém três valores float separados entre si por um espaço que correspondem às notas de um aluno.

### Saída

Para cada caso de teste o programa deve imprimir o conjunto em ordem crescente de matrícula. Uma linha deve ser impressa para cada elemento do conjunto, contendo: a matrícula seguida pelo nome do aluno, seguida pelas três notas do aluno. Após cada caso de teste o programa deve imprimir uma linha em branco para separar a saída de um caso de teste de outro. Veja o exemplo.

## Exemplo

Entrada				
2				
4				
4467				
Luiz Felipe				
2.0 3.9 7.8				
3245				
Antonio Luiz				
8.8 7.9 10.0				
1234				
Maria Tereza				
8.9 10.0 10.0				
9874				
Paulo Pedro				
7.0 8.0 9.0				
3				
8712				
Roberto Carlos				
8.9 9.8 10.0				
3456				
ErasmO Carlos				
3.4 8.7 9.0				
8912				
Michele Pfifer				
7.9 9.0 10.0				
Saída				
1234	Maria Tereza	8.90	10.00	10.00
3245	Antonio Luiz	8.80	7.90	10.00
4467	Luiz Felipe	2.00	3.90	7.80
9874	Paulo Pedro	7.00	8.00	9.00
3456	ErasmO Carlos	3.40	8.70	9.00
8712	Roberto Carlos	8.90	9.80	10.00
8912	Michele Pfifer	7.90	9.00	10.00