

**MAC-115 Introdução à Computação para Ciências Exatas e Tecnologia**  
**IO – Terceiro Exercício-Programa – Entregar até 09/11/2007**

**O Problema da Mochila**

São dados um inteiro positivo  $n$  e dois vetores  $quant$  e  $custot$ , ambos com  $n$  números reais. O vetor  $quant$  contém as quantidades disponíveis, em quilos, de  $n$  produtos, e o vetor  $custot$  representa o custo  $total$  de cada produto. É dado também um número real  $m$  representando a capacidade total, em quilos, de uma mochila. Deseja-se encher a mochila de modo a obter o maior preço total possível para o seu conteúdo.

Para resolver este problema, deve-se colocar os produtos na mochila em ordem decrescente de preço por quilo, ou seja, inicialmente coloca-se na mochila toda a quantidade disponível do produto que tenha o maior preço por quilo, depois o produto que tenha o segundo maior preço por quilo, e assim por diante, até encher a mochila ou não tiver mais produtos. Note que podem ocorrer situações em que alguns produtos não serão colocados na mochila, e também situações em que será colocada na mochila só uma parte do último produto, e não todo ele.

Escreva um **programa** na linguagem C (padrão ANSI), que resolva o problema da mochila, para vários conjuntos de produtos e mochilas.

Para cada problema a ser resolvido, seu programa deve ler um número inteiro  $n$  ( $1 \leq n \leq 50$ ) e um número real  $m$ , e para cada um dos  $n$  produtos, ler a sua quantidade e o seu custo total. O programa deve imprimir os valores de  $n$  e  $m$ , e para cada um dos  $n$  produtos, imprimir o seu número, a sua quantidade, e os seus custos total e por quilo. O produto de número 1 é o primeiro cujos dados são fornecidos, o produto de número 2 é o segundo cujos dados são fornecidos, etc. O programa deve definir um vetor contendo o preço por quilo de cada produto, e ordenar esse vetor para determinar os produtos a serem colocados na mochila. O programa deve fornecer como resposta o número de produtos colocados na mochila, e o número de cada produto colocado na mochila e a respectiva quantidade, na ordem em que os mesmos foram colocados. Escreve também o custo total dos produtos colocados na mochila.

**Entrada e saída**

Os dados de entrada devem ser lidos de um arquivo, cujo nome deve ser fornecido pelo usuário. Este arquivo deve conter números de acordo com o seguinte formato:

1. A primeira linha contém somente os valores de  $n$  e  $m$  para o primeiro problema a ser resolvido.
2. As linhas seguintes contém pares de números reais, indicando quantidade em quilos e o preço total do produto.
3. Uma vez fornecidos os  $n$  pares, ou a próxima linha contém exclusivamente valores de  $n$  e  $m$  para um novo problema, ou o arquivo de dados termina.

A saída também deverá ser escrita num arquivo, cujo nome deve ser fornecido pelo usuário. O arquivo de saída deve conter um cabeçalho com pelo menos o seu nome e número USP.

**Exemplo de entrada e de saída**

Suponha que um arquivo de dados tenha o conteúdo a seguir.

```

10      70.5
4.5    1462.50
5.0    1500.00
7.5    2775.00
6.5    1982.50
12.5   4275.00
10.5   4620.00
8.0    2000.00
7.5    2152.50
15.0   5325.00
22.5   6862.50
7       350
10     2300
12.7   3456
38.4   2900.43
42.6   9870.00
12.3   1234.56
45.9   2987.99
34.9   7777.77

```

A saída do seu programa, para o arquivo de dados descrito acima, deve ser algo da seguinte forma:

```

*****
* ..... *
* Nome do aluno ...      Numero USP ... *
* ..... *
*****

```

Problema 1

Numero de produtos : 10  
Capacidade da mochila (em quilos) : 70.50

Produto (numero)	Quantidade (quilos)	Preco total	Preco (quilo)
1	4.50	1462.50	325.00
2	5.00	1500.00	300.00
3	7.50	2775.00	370.00
4	6.50	1982.50	305.00
5	12.50	4275.00	342.00
6	10.50	4620.00	440.00
7	8.00	2000.00	250.00
8	7.50	2152.50	287.00
9	15.00	5325.00	355.00
10	22.50	6862.50	305.00

Foram colocados 6 produtos na mochila, na seguinte ordem:

Produto (numero)	Quantidade (quilos)
6	10.50
3	7.50
9	15.00
5	12.50
1	4.50
10	20.50

Peso total dos produtos na mochila : 70.50 (quilos)  
Custo total dos produtos na mochila : 24710.00

-----

## Problema 2

Numero de produtos : 7

Capacidade da mochila (em quilos) : 350.00

Produto (numero)	Quantidade (quilos)	Preco total	Preco (quilo)
1	10.00	2300.00	230.00
2	12.70	3456.00	272.13
3	38.40	2900.43	75.53
4	42.60	9870.00	231.69
5	12.30	1234.56	100.37
6	45.90	2987.99	65.10
7	34.90	7777.77	222.86

Foram colocados 7 produtos na mochila, na seguinte ordem:

Produto (numero)	Quantidade (quilos)
2	12.70
4	42.60
1	10.00
7	34.90
5	12.30
3	38.40
6	45.90

Peso total dos produtos na mochila : 196.80 (quilos)

Custo total dos produtos na mochila : 30526.75

---

## Funções a serem implementadas

O seu programa deve conter, obrigatoriamente, pelo menos as seguintes funções.

- `void custo_quilo (int n, float q[], float ct[], float cq[]);`  
Recebe um inteiro  $n$  e dois vetores reais  $q$  e  $ct$  contendo, respectivamente, a quantidade e o custo total de cada um dos  $n$  produtos.  
Determina o vetor real  $cq$ , armazenando o custo por quilo de cada um dos  $n$  produtos.
- `void ordena (int n, int num[], float cq[]);`  
Recebe um inteiro  $n$ , um vetor inteiro  $num$  com os números dos  $n$  produtos, e um vetor real  $cq$  contendo o custo por quilo de cada um dos  $n$  produtos.  
Coloca os  $n$  números no vetor  $cq$  em ordem decrescente, alterando, respectivamente, o vetor  $num$ .
- `int enche_mochila (float capm, int n, float q[], int num[], float *qultp, float *ptm);`  
Recebe um número real  $capm$  e um inteiro  $n$ , que representam, respectivamente, a capacidade total (em quilos) de uma mochila e o número total de produtos. Recebe também o vetor  $q$ , contendo a quantidade (em quilos) de cada produto; e o vetor  $num$ , contendo os números dos produtos, ordenados de acordo com o custo por quilo (ou seja, o primeiro número no vetor  $num$  é o número de um produto com maior custo por quilo).  
A função deve devolver em  $*qultp$  a quantidade (em quilos) do último produto colocado na mochila, pois pode não ter colocado toda a quantidade; e em  $*ptm$  o peso total dos produtos na mochila.  
A função deve determinar e devolver o número total de produtos colocados na mochila.

4. float custo\_mochila (int np, float custo\_ult, float ct[], int num[]);

Recebe um inteiro *np* que representa o número total de produtos na mochila; e um número real *custo\_ult* com o custo total do último produto colocado na mochila. Recebe também o vetor *ct*, contendo o custo total de cada produto; e o vetor *num*, contendo os números dos produtos, ordenados de acordo com o custo por quilo (em ordem decrescente).

A função deve determinar e devolver o custo total dos produtos na mochila.

## Uso de arquivos para entrada e saída

Em seguida, descrevemos alguns trechos para o terceiro programa, destacando partes da entrada e da saída.

```
#include <stdio.h>
#define COMPNOME 31
/* mais definicoes de constantes ... */

/* declaracoes das funcoes ..... */

int main ()
{
    FILE *arqentra, *arqsai;
    int n, i;
    float m;
    char nomearq[COMPNOME];
    /* declaracoes de outras variaveis ..... */

    printf ("Digite o nome (maximo %d caracteres) do arquivo de entrada :\n", COMPNOME-1);
    scanf ("%s", nomearq);
    arqentra = fopen (nomearq, "r");
    if (arqentra == NULL) {
        printf ("ERRO: nao foi possivel abrir arquivo %s\n", nomearq);
        return 1;
    }

    printf ("Digite o nome (maximo %d caracteres) do arquivo de saida :\n", COMPNOME-1);
    scanf ("%s", nomearq);
    arqsai = fopen (nomearq, "w");
    if (arqsai == NULL) {
        printf ("ERRO: nao foi possivel criar arquivo %s\n", nomearq);
        return 2;
    }

    /* ... */

    fprintf (arqsai, "*****\n");
    fprintf (arqsai, "* ..... *\n");
    fprintf (arqsai, "* Nome do aluno ... Numero USP ... *\n");
    fprintf (arqsai, "* ..... *\n");
    fprintf (arqsai, "*****\n");

    /* ... */
```

```

while (!feof (arqentra)) {
    fscanf (arqentra, "%d %f ", &n, &m);
    /* ... */
    for (i = 1; i <= n; i++) {
        fscanf (arqentra, "%f %f ", &???, &??);
        /* ... */
    }
    /* ... */
}
fclose (arqentra);
fclose (arqsai);
return 0;
}

/* definicoes das funcoes ..... */

```

### Observações gerais

1. Para testar o seu programa, crie vários arquivos de entrada.
2. Só utilize os recursos da linguagem C que forem vistos em aula até a data de entrega.  
Não utilize variáveis globais.
3. Defina constantes para os tamanhos máximos dos vetores utilizados no seu programa.
4. Não altere os nomes das variáveis, os nomes das funções, e os nomes e a ordem dos parâmetros das funções, que aparecem neste enunciado.
5. Continuam valendo as observações do primeiro exercício-programa.