UNIVERSIDADE FEDERAL DE MINAS GERAIS
Departamento de Ciência da Computação

Disciplina	Turmas
Algoritmos e Estruturas de Dados II	M1, M2, N, W e F

Professores
Danilo J. S. Oliveira, Gisele L. Pappa, Ítalo Cunha, Loïc Cerf e William Schwartz

Execício de Programação em Linguagem C

Uma determinada empresa tem em sua linha de produção a confecção de dez produtos que são produzidos e vendidos mensalmente durante o ano todo. Existe um sistema computacional que gera dois arquivos textos ao longo do ano. Um dos arquivos refere-se à produção do 1^o semestre e o outro à produção do 2^o semestre. Estes arquivos possuem dez linhas com seis números inteiros separados por espaços em branco (cada coluna corresponde a um mês). As linhas seguem a ordem dos seguintes produtos:

1. Sola inteira
2. Meia sola
3. Quarto de sola
4. Entre sola
5. Palmilha
6. Meia Palmilha
7. Salto alto
8. Salto médio
9. Salto curto
10. Salto baixo

Um exemplo de um arquivo texto com informações do 1^o semestre seria:

10	12	20	12	19	23
23	32	12	90	10	87
32	43	14	87	11	78
54	56	41	65	23	79
45	57	51	77	32	97
12	58	15	12	42	76
55	23	76	99	24	65
64	21	87	87	56	45
52	67	55	67	65	12
12	34	75	90	77	12

Um exemplo de um arquivo texto com informações do 2^o semestre seria:

```
30  40  50  50  10  10
98  53  65  12  13  41
97  11  22  56  53  43
97  22  32  56  21  33
97  33  64  77  54  43
99  44  74  67  64  32
90  55  84  56  75  92
23  66  88  47  87  82
43  77  77  47  52  82
23  88  99  37  82  72
```

Os arquivos textos possuem em seus nomes informações que se referem a produção, o ano e o semestre. Exemplos dos nomes dos arquivos seriam: prod-2010-1.txt, prod-2010-2.txt.

Todo o mês de janeiro do ano corrente deve ser gerado um arquivo com as informações obtidas dos arquivos de produções semestrais. As informações geradas seriam: produção anual total em unidades por produto, produção anual total em reais por produto, produção total mensal em unidades de produtos, produção total mensal em reais por produto e os respectivos totais.

O exemplo de arquivo gerado é apresentado a seguir. O arquivo deve ter o nome prod-2011-2010.txt sempre o ano e o ano dos semestres envolvidos.

```
-----------------------------------------------
Produção anual total em unidades por produto
-----------------------------------------------
Sola inteira     286--
Meia sola        536
Quarto de sola   547
Entre sola       579
Palmilha         727
Meia palmilha    595
Salto alto       794++
Salto médio      753
Salto curto      696
Salto baixo      701
Total:          6214
-----------------------------------------------
Produção anual total em reais por produto
-----------------------------------------------
Sola inteira      357,50
Meia sola         455,60
Quarto de sola    355,55--
Entre sola       1997,55
Palmilha         2268,24
Meia palmilha    1636,25
Salto alto       3191,88++
Salto médio      2597,85
Salto curto      2074,08
Salto baixo      1549,21
Total:          16483,71
-----------------------------------------------
Produção total mensal em unidades de produtos
-----------------------------------------------
Janeiro    359--
Fevereiro  403
Março      446
Abril      686
Maio       359--
Junho      574
```

```
Julho       697++
Agosto      489
Setembro    655
Outubro     505
Novembro    511
Dezembro    530
Total:     6214
-------------------------------------------------------
Produção total mensal em reais por produto
-------------------------------------------------------
Janeiro     1035,93
Fevereiro   1040,40
Março       1321,44
Abril       1742,23++
Maio         987,64--
Junho       1416,42
Julho       1713,51
Agosto      1274,80
Setembro    1735,16
Outubro     1335,89
Novembro    1412,76
Dezembro    1467,53
Total:     16483,71
-------------------------------------------------------
```

Observe que no arquivo de saída os caracteres $--$ e $++$ indicam os menores e os maiores valores obtidos.

O que deve ser feito

1. Elaborar um programa capaz de gerar o arquivo de saída proposto a partir dos arquivos de entradas das produções semestrais.

2. O programa deve ter a função `void LerArquivoProducaoSemestral(char* arquivo, int *M)` que lê um arquivo com nome armazenado na variável *arquivo* de informações da produção semestral. Esta função recebe como parâmetro um endereço de uma matriz 10×6.

3. O programa deve ler via entrada padrão, por intermédio da função `void LerValoresUnitarios(double *V)` que lê os valores unitários de cada um dos 10 produtos em questão e os coloca em um vetor. A função `LerValoresUnitarios` recebe como parâmetro o endereço para um vetor de `double`.

4. O programa deve gerar uma matriz de dimensões 12×14 por intermédio da função `GerarMatrizFinal(double *ProdSem1, double *ProdSem2, double *ValorUnitario, double *M)`, onde *ProdSem1* contém o endereço da matriz de produção do primeiro semestre, *ProdSem2* contém o endereço da matriz de produção do segundo semestre, *ValorUnitario* contém o endereço do vetor com valores unitários e M contém o endereço de uma variável de uma matriz 12×14. Onde nas linhas de 0 até 9 e colunas 0 até 11 deve ter as informações das matrizes semestrais de produção. As linhas 10 e 11 terão as informações dos totais produzidos por mês e o montante em reais por mês. As colunas 12 e 13 terão as informações dos totais dos produtos produzidos anualmente e o montante em reais anuais por produto.

5. O programa deve ter uma função chamada `void GerarArquivoDeSaida(char *nome_arquivo, double *M)` que recebe o nome do arquivo de saída e a matriz com todas as informações. A função `void GerarArquivoDeSaida` deve chamar outras funções que seriam: `double MaiorValorDaLinha(double *M, int nlin)`, `double MaiorValorDaColuna(double *M, int ncol)`, `MenorValorDaLinha(double *M, int nlin)` e `double MenorValorDaColuna(double *M, int ncol)`. Estas funções retornam o menor ou maior valor de uma linha ou de uma coluna de uma matriz e recebem como parâmetro o endereço de uma matriz do tipo `double` e o número da linha ou da coluna, conforme a função que

esteja sendo utilizada, retornam o menor ou maior um valor e devem ser chamadas dentro da função `GerarArquivoDeSaida`.

6. Todas as funções implementadas devem possuir um cabeçalho conforme o exemplo a seguir:

```
/*-------------------------------------------------------------------
Protótipo: GerarArquivoDeSaida(char *nome_arquivo, double *M)
Função: Gerar o arquivo de saída proposto.
Entrada: Nome do arquivo e a matriz de dados.
Saída: Nenhuma
-------------------------------------------------------------------*/
```

7. O programa deve ser em C padrão, ou seja, o código fonte do programa deve ser compilável no Linux, não devendo ser utilizada nenhuma biblioteca que seja específica do sistema operacional MS-WINDOWS.

8. Deve ser enviado (submetido) **APENAS** o código fonte, em um arquivo ".zip"contendo um arquivo .h (com as declarações das funções) e um arquivo .c (com a implementação das funções).

9. Valor 3 pontos para o programa que compilar, estiver documentado (linhas de comentários ao longo do código fonte, cabeçalhos das funções) e executar corretamente, produzindo o resultado solicitado.

10. Deve ser entregue via *moodle* até as 23:45 horas do dia 26/03/2012. **Os trabalhos não serão aceitos, em hipótese nenhuma, fora do prazo.**