**Observações quanto à atividade:**

1- A identificação de cópia implicará na nulidade da(s) questão(ões) para os trabalhos identificados.

2- Dúvidas quanto à interpretação podem ser sanadas com o professor.

3- Para as questões que envolverem “alguma forma de código” será considerada a sintaxe da LP C.

4- Critérios para a avaliação (quando cabível):

\* sintaxe e semântica,

\* nomenclatura (rotinas e variáveis),

\* organização do código (espaçamento horizontal e vertical),

\* tipos adequados para variáveis, parâmetros e retornos das rotinas,

\* representação correta das equações no código,

\* não deverá haver operações de leitura e escrita nas rotinas,

\* mensagens adequadas para a interação com o usuário do programa durante as leituras e as escritas.

5- **Entrega até 30/11/2017 (até 23:55h), via moodle ou email martins@utfpr.edu.br**.

**(1.0) 01)** A empresa XYZ do Brasil S.A. produz **M** diferentes produtos. Todos os produtos são fabricados em **N** pontos de manufatura distintos e distribuídos por **P** centros atacadistas. **Escreva** a rotina com cabeçalho definido abaixo. Deverá haver um laço no qual serão lidos conjuntos de valores (i, j, k, qtde) que representem os códigos dos produtos (*i=[1..M]*), os códigos das fábricas (*j=[1..N]*) e os códigos dos centros distribuidores (*k=[1..P]*), bem como a quantidade produzida (*qtde*).

int LeEstoqueMatrizExplicita(int M, int N, int P, int mEstoque[M][N][P])

**Observações:**

**(a)** deverá haver tratamento para possíveis valores inválidos para os valores de M, N e P;

**(b)** se o mesmo conjunto/trio de códigos (produto, fábrica e centro – i, j, k) for digitado mais de uma vez, as quantidades informadas deverão ser somadas/acumuladas;

**(c)**o**processo de leitura** é encerrado quando o valor -1 (um negativo) for fornecido para, pelo menos, um dos três elementos/códigos anteriores (produto, fábrica e centro – i, j, k); e

**(d)** estas rotinas deverão estar em uma biblioteca chamada EntradaSaidaPadrao (arquivos .h e .c).

**(1.0) 02)** Para auxiliar a empresa XYZ do Brasil S.A. no controle de seu estoque, **implemente** a rotina com cabeçalho definido abaixo. Esta deverá apresentar as diferentes combinações para os conjuntos/trios de códigos (produto, fábrica e centro – i, j, k) identificando a quantidade de itens produzidos para um determinado produto i em uma determinada fábrica j e enviados para um determinado centro k (*vide* exemplo abaixo).

int EscreveEstoqueMatrizExplicita(int M, int N, int P, int mEstoque[M][N][P])

**Observações:**

**(a)** deverá haver tratamento para possíveis valores inválidos para os valores de M, N e P;

**(b)** se a quantidade produzida para uma combinação/conjunto/trio de códigos (produto, fábrica e centro – i, j, k) for 0 (zero), esta combinação não deverá ser escrita;e

**(c)** estas rotinas deverão estar em uma biblioteca chamada EntradaSaidaPadrao (arquivos .h e .c).

**Exemplo da saída final**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Código do Produto** | **Código da Fábrica** | **Código do C. Atacadista** | **Total de Itens** |
| 1 | 1 | 1 | 10 |
| 1 | 1 | 2 | 50 |
| 1 | 1 | 3 | 22 |
| **...** | **...** | **...** | **...** |
| 1000 | 50 | 199 | 29 |
| 1000 | 50 | 200 | 45 |

**(2.0) 03)** Para auxiliar a empresa XYZ do Brasil S.A. no armazenamento dos dados referentes a seu estoque, **implemente** a rotina com cabeçalho definido abaixo. Esta deverá gravar no arquivo *sArquivo* as diferentes combinações para os conjuntos/trios de códigos (produto, fábrica e centro – i, j, k) e a quantidade de itens produzidos para um determinado produto i em uma determinada fábrica j e enviados para um determinado centro k.

int GravaEstoqueMatrizExplicita(char \*sArquivo, int M, int N, int P, int mEstoque[M][N][P])

**Observações:**

**(a)** deverá haver tratamento para possíveis valores inválidos para os valores de sArquivo, M, N e P; e

**(b)** estas rotinas deverão estar em uma biblioteca chamada EntradaSaidaHD (arquivos .h e .c).

**(2.0) 04)** Complementando a funcionalidade do exercício anterior, **implemente** a rotina com cabeçalho definido abaixo. Esta deveráler o arquivo *sArquivo*que contém as diferentes combinações para os conjuntos/trios de códigos (produto, fábrica e centro – i, j, k) e a quantidade de itens produzidos para um determinado produto i em uma determinada fábrica j e enviados para um determinado centro k.

int RecuperaEstoqueMatrizExplicita(char \*sArquivo, int M, int N, int P, int mEstoque[M][N][P])

**Observações:**

**(a)** deverá haver tratamento para possíveis valores inválidos para os valores de sArquivo, M, N e P; e

**(b)** estas rotinas deverão estar em uma biblioteca chamada EntradaSaidaHD (arquivos .h e .c).

**(2.0) 05)**Escreva a rotina *main*.

Ao iniciar o programa, deverá ser verificado se existe algum arquivo de dados. Se houver, os dados deste arquivo deverão ser carregados para memória principal.

Na sequência, o usuário deverá escolher uma das funcionalidades (rotinas) anteriores, isto é, 1(c) ou 2(b).

Também poderão ser alterados valores específicos da matriz por meio das rotinas

int AlteraMatrizExplicita(int M, int N, int P, int mEstoque[M][N][P], int i, int j, int k, int valor)

Ao final, os dados constantes na matriz deverão ser escritos por meio da rotina 3(a).

Para cada possibilidade desta questão, os dados necessários deverão ser lidos e fornecidos à rotina selecionada, a mesma deverá ser chamada e o programa deverá apresentar os resultados retornados ou mensagens equivalentes (que descrevam o que ocorreu na rotina).

**(valor 2.0) 05)**Cada uma das rotinas anteriores deverá conter documentação considerando os 2 modelos a seguir.

**Exemplo de Documentação 01 - Bhaskara**

/\*\*

Rotina: CalculaRaizesBhaskara

Objetivo: calcular as raizes de uma equacao de 2o. grau por meio de Bhaskara.

Parametros:

Entrada:

coeficientes da equacao Ax2 + Bx + C

a, b, c do tipo real/float.

Saida:

pX1 e pX2: raizes da equacao, caso existam.

Entrada/Saida

nao ha.

Retorno:

+1: raizes calculadas com sucesso e retornadas por meio de pX1 e pX2.

Pseudo-codigo:

Calcula delta

Calcula raizes

\*/

int CalculaRaizesBaskhara(float a, float b, float c, float \*pX1, float \*pX2)

{

float delta;

// calcula o valor de delta

delta = pow(b,2) - 4\*a\*c;

// calcula raizes

\*pX1 = (-b + sqrt(delta))/(2\*a);

\*pX2 = (-b - sqrt(delta))/(2\*a);

return +1;

}

**Exemplo de Documentação 02 - Troca de valores**

/\*\*

Rotina: TrocaValores

Objetivo: troca os valores recebidos por meio dos parametros.

Parametros:

Entrada:

nao ha.

Saida:

nao ha.

Entrada/Saida

pA e pB do tipo real/float.

Retorno:

nao ha.

Pseudo-codigo:

Salva valor referenciado por pA

Copia valor referenciado por pB para pA

Copia valor anteriormente referenciado por pA e salvo para pB

\*/

void TrocaValores(int \*pA, int \*pB)

{

int temp;

// salva valor referenciado por pA em temp

temp = \*pA;

// copia valor referenciado por pB para pA

\*pA = \*pB;

// copia valor anteriormente referenciado por pA e salvo para pB

\*pB = temp;

}