

- 1 -) (3,0) Seja  $A = (a_{ik})$  uma matriz  $m \times n$  e  $B = (b_{kj})$  uma matriz  $n \times p$ . Escreva um programa em linguagem de programação C que compute o produto  $C = AB$ , sendo  $C$  uma matriz  $m \times p$ , na qual:

$$C_{ij} = \sum_{k=1}^n a_{ik}b_{kj}$$

Seu programa deve ler da entrada padrão as duas matrizes  $A$  e  $B$ , computar e imprimir na saída padrão a matriz de resposta  $C$ , seguindo a especificação de entrada e saída a seguir. A primeira linha da entrada contém os três inteiros:  $m$ ,  $n$  e  $p$ , separados por um espaço em branco. Cada uma das próximas  $m$  linhas contém  $n$  números, separados por um espaço em branco; o  $k$ -ésimo elemento da  $i$ -ésima linha é o elemento  $a_{ik}$ . Cada uma das próximas  $n$  linhas contém  $p$  números, separados por um espaço em branco; o  $j$ -ésimo elemento da  $k$ -ésima linha é o elemento  $b_{kj}$ . Na saída, seu programa deve gerar  $m$  linhas, cada uma contendo  $p$  números com até 4 casas decimais separados por um espaço em branco; o  $j$ -ésimo elemento da  $i$ -ésima linha é o elemento  $c_{ij}$ . Todos os elementos das matrizes são números flutuantes de precisão dupla conforme a IEEE 754. Considere que  $0 < m, n, p < 1000$ . Para as variáveis definidas acima, utilizar os mesmos nomes para as variáveis relacionadas no seu programa (e.g.,  $A, B$  e  $C$  para as matrizes,  $i, j, k, m, n$  e  $p$  para contadores e tamanho das matrizes).

Resposta:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int m, n, p;
6
7     // Leitura das dimensões
8     scanf("%d %d %d", &m, &n, &p);
9
10    double A[m][n], B[n][p], C[m][p];
11
12    // Leitura Matriz A
13    for (int i = 0; i < m; i++) {
14        for (int k = 0; k < n; k++) {
15            scanf("%lf", &A[i][k]);
16        }
17    }
18
19    // Leitura Matriz B
20    for (int k = 0; k < n; k++) {
21        for (int j = 0; j < p; j++) {
22            scanf("%lf", &B[k][j]);
23        }
24    }
25
26    // Computar C
27    for (int i = 0; i < m; i++) {
28        for (int j = 0; j < p; j++) {
29            C[i][j] = 0;
30            for (int k = 0; k < n; k++) {
31                C[i][j] += A[i][k] * B[k][j];
32            }
33        }
34    }
35
36    // Escrever C
```

```

37     for (int i = 0; i < m; i++) {
38         for (int j = 0; j < p; j++) {
39             if (j==0) {
40                 printf("%.4lf", C[i][j]);
41             } else {
42                 printf(" %.4lf", C[i][j]);
43             }
44         }
45         printf("\n");
46     }
47 }

```

2 -) (3,0) Utilizando a linguagem de programação C, escreva um programa que elabore e imprima uma pirâmide de valores, conforme o exemplo abaixo apresentado:

```

5-5-5 4-4-4 3-3-3 2-2-2 1-1-1
4-4-4 3-3-3 2-2-2 1-1-1
3-3-3 2-2-2 1-1-1
2-2-2 1-1-1
1-1-1

```

Sobre as entradas e saídas, seu programa deve ler dois inteiros da entrada padrão, computar e imprimir o resultado esperado na saída padrão. A entrada contém dois números inteiros:  $m$  e  $n$ , separados por um espaço em branco. O primeiro inteiro ( $m$ ) representa o número inicial (5 no caso do exemplo apresentado), enquanto o segundo inteiro ( $n$ ) indica o número de repetições de cada valor (no exemplo apresentado,  $n = 3$ , ou seja, cada número foi apresentado no formato  $i - i - i$ ). Considere que  $0 < m, n < 1000$ . Para as variáveis definidas acima, utilizar os mesmos nomes para as variáveis relacionadas no seu programa.

Resposta:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5     int m, n;
6
7     // Leitura das dimensões
8     scanf("%d %d", &m, &n);
9
10    for (int i=m; i>0; i--) {
11        for (int j=i; j>0; j--) {
12            for (int k=0; k<n; k++) {
13                if (k==0)
14                    printf("%i", j);
15                else
16                    printf("-%i", j);
17            }
18            if (j>1)
19                printf(" ");
20            else
21                printf("\n");
22        }
23        // printf("\n"); // se remover else \ printf("\n"); acima
24    }
25
26    return 0;
27 }

```

3 -) (4,0) Considere que os tipos int, short e char possuem tamanho de 4, 2 e 1 bytes, respectivamente. Considere o alinhamento padrão do compilador GCC: membros devem ser alinhados conforme seu tamanho e a estrutura deve ser alinhada conforme o tamanho do seu maior membro. Qual a saída para cada um dos programas abaixo:

(a) Programa 1:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef union t_Ponto2D {
5     int data[2];
6     struct {
7         int X, Y;
8     };
9 } Ponto2D;
10
11 int main() {
12     Ponto2D p1;
13     p1.data[0] = 1;
14     p1.data[1] = 2;
15     p1.X = 3;
16     p1.Y = 4;
17     printf ("%d %d %d %d %d\n", p1.data[0], p1.data[1], p1.X, p1.Y, sizeof(Ponto2D));
18     return 0;
19 }
```

Resposta: 3 4 3 4 8

(b) Programa 2:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct {
5     char a; int b; char c; short d;
6 } t1_t;
7
8 typedef struct {
9     int b; char a; char c; short d;
10 } t2_t;
11
12 typedef struct {
13     int b; char a;
14 } t3_t;
15
16 typedef struct {
17     short a; short b; char c;
18 } t4_t;
19
20
21 int main() {
22     printf ("%d %d %d %d\n", sizeof(t1_t), sizeof(t2_t), sizeof(t3_t), sizeof(t4_t));
23     return 0;
24 }
```

Resposta: 12 8 8 6

(c) Programa 3:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct {
5     char inicial;
6     int idade;
7 } jogador_t;
8
9
10 int main() {
11     jogador_t vetor[100];
12     printf("%d %d\n", sizeof(jogador_t), sizeof vetor);
13     return 0;
14 }
```

Resposta: 8 800

(d) Programa 4:

```
1 #include <stdio.h>
2 #include <stdint.h>
3
4 int main(int argc, char* argv[]) {
5     uint8_t start = 0x01;
6     uint8_t polynomial = 0b00000110;
7     uint8_t lfsr = start;
8     while (1) {
9         uint8_t lsb = lfsr & 0b1; // bitwise AND
10        printf("%c", lsb ? '1' : '0');
11        lfsr = lfsr >> 1;
12        if (lsb)
13            lfsr = lfsr ^ polynomial; // bitwise XOR
14
15        if (lfsr == start)
16            break;
17    }
18    printf("\n");
19    return 0;
20 }
```

Resposta 1011100



# Assinaturas do documento



Código para verificação: **E73Z60MG**

Este documento foi assinado digitalmente pelos seguintes signatários nas datas indicadas:

- ✓ **YURI KASZUBOWSKI LOPES** (CPF: 063.XXX.259-XX) em 20/11/2023 às 09:06:39  
Emitido por: "SGP-e", emitido em 02/06/2021 - 13:17:43 e válido até 02/06/2121 - 13:17:43.  
(Assinatura do sistema)
  
- ✓ **ADRIANO FIORESE** (CPF: 743.XXX.339-XX) em 20/11/2023 às 15:55:57  
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:36:39 e válido até 30/03/2118 - 12:36:39.  
(Assinatura do sistema)
  
- ✓ **GUILHERME PIEGAS KOSLOVSKI** (CPF: 002.XXX.110-XX) em 20/11/2023 às 16:05:07  
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:35:42 e válido até 30/03/2118 - 12:35:42.  
(Assinatura do sistema)
  
- ✓ **ROBERTO SILVIO UBERTINO ROSSO JUNIOR** (CPF: 283.XXX.604-XX) em 20/11/2023 às 17:23:23  
Emitido por: "SGP-e", emitido em 30/03/2018 - 12:37:29 e válido até 30/03/2118 - 12:37:29.  
(Assinatura do sistema)

Para verificar a autenticidade desta cópia, acesse o link <https://portal.sgpe.sea.sc.gov.br/portal-externo/conferencia-documento/VURFU0NfMTIwMjJfMDAwNTE4MTFfNTE4NjFfMjAyM19FNzNaNjBjBNRw==> ou o site <https://portal.sgpe.sea.sc.gov.br/portal-externo> e informe o processo **UDESC 00051811/2023** e o código **E73Z60MG** ou aponte a câmera para o QR Code presente nesta página para realizar a conferência.