

1) Considerando a matriz  $A = [a_{ij}]_{n \times n}$  abaixo com  $n = 5$ , faça um algoritmo para resolver as questões a seguir, criando uma função para cada uma delas.

$$A = \begin{bmatrix} 0 & 2 & 8 & 1 & 7 \\ 1 & 5 & 8 & 2 & 2 \\ 3 & 4 & 0 & 8 & 1 \\ 9 & 1 & 2 & 7 & 2 \\ 3 & 0 & 1 & 0 & 4 \end{bmatrix}$$

(a)  $S = \sum_{i=1}^n \sum_{j=1}^n a_{ij}$

(b)  $S = \sum_{j=1}^n \sum_{i=1}^n a_{ij}$

(c)  $S = \sum_{i=1}^n \sum_{j=1}^n a_{ij}, i \neq j$

(d)  $S = \sum_{j=1}^n \sum_{i=1}^n a_{ij}, i = j$

(e)  $S = \sum_{i,j=1}^n a_{ij}, i < j$

(f)  $S = \sum_{i,j=1}^n a_{ij}, i > j$

(g)  $S = \sum_{i,j=1}^n a_{ij} \times a_{ji}, i < j$

(h) Considere um número inteiro  $x$  (lido na função main) para a equação a seguir:

$$S = \sum_{i,j=1}^n x \times a_{ij}$$

(i) Crie (na função main) um vetor  $V$  com  $n$  elementos para a equação a seguir:

$$V_i = \sum_{j=1}^n a_{ij}, 1 \leq i \leq n$$

Após o preenchimento do vetor  $V$ , realize a sua impressão na tela (dentro da própria função).

(j) Crie (na função main) um vetor  $P$  com  $n$  elementos para a equação a seguir:

$$P_j = \prod_{i=1}^n a_{ij}, 1 \leq j \leq n$$

Após o preenchimento do vetor  $P$ , realize a sua impressão na tela (dentro da própria função).

2) Considere o vetor  $V$  com 50 números inteiros gerados aleatoriamente de 1 a 100. Faça um algoritmo que modifique  $V$  de modo que:

$$V_i = \sum_{\substack{j=0, \\ i \neq j}}^{49} V_j, 0 \leq i \leq 49$$

Crie e utilize três **procedimentos**: um para preencher o vetor, outro para modificá-lo e o terceiro para realizar a impressão do vetor antes e depois da modificação.

**Obs.:** Não é permitido utilizar qualquer estrutura de dados auxiliar.

3) Considere o vetor  $V$  com  $n$  números inteiros lidos pelo teclado, sendo  $n = 30$ . Faça um algoritmo para ler pelo teclado na função *main* o número inteiro  $k$  ( $1 \leq k \leq 4$ ) e imprimir o que se pede na expressão abaixo:

$$\text{Max}_{0 \leq i \leq n-k} \sum_{j=i}^{i+(k-1)} V_j$$

Crie e utilize um **procedimento** para ler o vetor e uma **função** para retornar o que se pede. A impressão desta informação deve ser realizada na função *main*.

**Obs.:** Não é permitido utilizar qualquer estrutura de dados auxiliar.

4) Considere os vetores  $V^1$  e  $V^2$ , ambos com 40 números inteiros lidos pelo teclado, e dois números inteiros  $x$  ( $x \neq 0$ ) e  $y$  ( $y \neq 0$ ) lidos pelo teclado na função *main*. Faça um algoritmo para verificar se a condição abaixo é verdadeira:

$$\sum_{\substack{i=0, \\ i \text{ ímpar}}}^{39} x \times |V_i^1| \leq \prod_{\substack{j=0, \\ j \text{ par}}}^{39} y \times V_j^2$$

Crie e utilize um **procedimento** para ler os dois vetores e uma **função** para retornar 1, se a condição é verdadeira, ou retornar 0 caso contrário. A impressão da informação (se a condição é verdadeira ou não) deve ser realizada na função *main*.

**Obs.:** Não é permitido utilizar qualquer estrutura de dados auxiliar.

### IMPORTANTE

- 1) Esta atividade deve ser feita **individualmente**;
- 2) Cada aluno(a) deve enviar as respostas desta atividade até às **11h59** do dia **23/06/2023** para o e-mail:

philippeleal@yahoo.com.br

- 3) Após a hora e a data marcada para o envio das respostas, **NÃO É MAIS PERMITIDO ENVIÁ-LAS**;
- 4) O e-mail considerado para correção será o **ÚLTIMO** recebido pelo Professor **dentro do prazo determinado**;
- 5) Ao enviar o e-mail, coloque como **Assunto** e **Nome do Arquivo**:

**PAA-Encontro02-SeuNome**

- 6) E-mails com o Assunto fora do padrão **NÃO SERÃO ACEITOS**.

# G:\Meu Drive\Classroom\Linguagens de programação\.C\Projeto e Analise de Algoritmos.C\PAA-Encontro-2\Questão 01.c

```
1  #include <stdio.h>
2  #define N 2
3
4  void somatoria1(int matriz[N][N]){
5      int s=0;
6      for(int i=0; i < N; i++){
7          for(int j=0; j < N; j++){
8              s = s + matriz[i][j];
9              printf("%d\n", s);
10         }
11     }
12     printf("O resultado final é: %d\n", s);
13 }
14
15 void somatoria2(int matriz[N][N]){
16     int s=0;
17     for(int i=0; i < N; i++){
18         for(int j=0; j < N; j++){
19             s = s + matriz[j][i];
20             printf("%d\n", s);
21         }
22     }
23     printf("O resultado final é: %d\n", s);
24 }
25
26 void somatoria3(int matriz[N][N]){
27     int s=0;
28     for(int i=0; i < N; i++){
29         for(int j=0; j < N; j++){
30             if(i != j){
31                 s = s + matriz[i][j];
32             }
33             printf("%d\n", s);
34         }
35     }
36     printf("O resultado final é: %d\n", s);
37 }
38
39 void somatoria4(int matriz[N][N]){
40     int s=0;
41     for(int i=0; i < N; i++){
42         for(int j=0; j < N; j++){
43             if(i == j){
44                 s = s + matriz[j][i];
45             }
46             printf("%d\n", s);
47         }
48     }
49     printf("O resultado final é: %d\n", s);
50 }
51
52 void somatoria5(int matriz[N][N]){
53     int s=0;
54     for(int i=0; i < N; i++){
55         for(int j=0; j < N; j++){
56             if(i < j){
57                 s = s + matriz[i][j];
58             }
59             printf("%d\n", s);
```

```

60     }
61 }
62 printf("O resultado final é: %d\n", s);
63 }
64
65 void somatoria6(int matriz[N][N]){
66     int s=0;
67     for(int i=0; i < N; i++){
68         for(int j=0; j < N; j++){
69             if(i > j){
70                 s = s + matriz[i][j];
71             }
72             printf("%d\n", s);
73         }
74     }
75     printf("O resultado final é: %d\n", s);
76 }
77
78 void somatoria7(int matriz[N][N]){
79     int s=0;
80     for(int i=0; i < N; i++){
81         for(int j=0; j < N; j++){
82             if(i < j){
83                 s = s + (matriz[i][j] * matriz[j][i]);
84             }
85             printf("%d\n", s);
86         }
87     }
88     printf("O resultado final é: %d\n", s);
89 }
90
91 void somatoria8(int matriz[N][N], int x){
92     int s=0;
93     for(int i=0; i < N; i++){
94         for(int j=0; j < N; j++){
95             s = s + matriz[i][j];
96             printf("%d\n", s);
97         }
98     }
99     s= s * x;
100    printf("O resultado final é: %d\n", s);
101 }
102
103
104 int main(){
105
106     int matriz[N][N], i, j, x;
107
108     for(i=0; i < N; i++){
109         for(j=0; j < N; j++){
110             printf("Digite um numero: ");
111             scanf("%d",&matriz[i][j]);
112         }
113     }
114     printf("\nDigite um numero para X: ");
115     scanf("%d",&x);
116
117     somatoria8(matriz,x);
118
119 }

```

## Questão 02.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void preencherVetor(int vetor[], int N) {
6      srand(time(NULL));
7      for (int i = 0; i < N; i++) {
8          vetor[i] = (rand() % 100) + 1;
9      }
10 }
11
12 void modificarVetor(int vetor[], int N) {
13     for (int i = 0; i < N; i++) {
14         int soma = 0;
15         for (int j = 0; j < N; j++) {
16             if (i != j) {
17                 soma += vetor[j];
18             }
19         }
20         vetor[i] = soma;
21     }
22 }
23
24 void imprimirVetor(int vetor[], int N) {
25     for (int i = 0; i < N; i++) {
26         printf("|%02d|", vetor[i]);
27     }
28     printf("\n\n");
29 }
30
31 int main() {
32     int N = 50;
33     int vetor[N];
34
35     preencherVetor(vetor, N);
36
37     printf("\n\n Vetor original:\n");
38     imprimirVetor(vetor, N);
39     modificarVetor(vetor, N);
40
41     printf("\n\n Vetor modificado:\n");
42     imprimirVetor(vetor, N);
43
44     return 0;
45 }
46
```

## Questão 03.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <limits.h>
4
5  void lerVetor(int vetor[], int n) {
6      printf("\n\nDigite os %d elementos do vetor:\n", n);
7      for (int i = 0; i < n; i++) {
8          scanf("%d", &vetor[i]);
9      }
10 }
11
12 int calcularExpressao(int vetor[], int n, int k) {
13     int max = INT_MIN;
14     for (int i = 0; i <= n - k; i++) {
15         int soma = 0;
16         for (int j = i; j <= (i + k - 1); j++) {
17             soma += vetor[j];
18         }
19         if (soma > max) {
20             max = soma;
21         }
22     }
23     return max;
24 }
25
26 int main() {
27
28     int n = 30;
29     int vetor[n];
30     lerVetor(vetor, n);
31
32     int k;
33     printf("\n\nDigite o valor de k (1 a 4): ");
34     scanf("%d", &k);
35
36     int resultado = calcularExpressao(vetor, n, k);
37     printf("\n\nResultado da expressao: %d \n\n", resultado);
38
39     return 0;
40 }
41
```

## Questão 04.c

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  void lerVetores(int vetor1[], int vetor2[], int N) {
5      printf("\n\nDigite os %d elementos do vetor V1:\n", N);
6      for (int i = 0; i < N; i++) {
7          scanf("%d", &vetor1[i]);
8      }
9
10     printf("\n\nDigite os %d elementos do vetor V2:\n", N);
11     for (int i = 0; i < N; i++) {
12         scanf("%d", &vetor2[i]);
13     }
14 }
15
16 int verificarCondicao(int vetor1[], int vetor2[], int N, int x, int y) {
17     int somatorio = 0, produtorio = 1;
18
19     for (int i = 1; i < N; i += 2) {
20         somatorio += x * abs(vetor1[i]);
21     }
22
23     for (int j = 0; j < N; j += 2) {
24         produtorio *= y * vetor2[j];
25     }
26
27     if (somatorio <= produtorio)
28         return 1; // Condição é verdadeira
29     else
30         return 0; // Condição não é verdadeira
31 }
32
33 int main() {
34     int N = 40;
35     int vetor1[N];
36     int vetor2[N];
37     lerVetores(vetor1, vetor2, N);
38
39     int x, y;
40     printf("\n\nDigite o valor de x: ");
41     scanf("%d", &x);
42     printf("\n\nDigite o valor de y: ");
43     scanf("%d", &y);
44
45     int resultado = verificarCondicao(vetor1, vetor2, N, x, y);
46
47     if (resultado == 1) {
48         printf("\n\nCondicao verdadeira.\n\n");
49     } else {
50         printf("\n\nCondicao falsa.\n\n");
51     }
52
53     return 0;
54 }
55
```