

UEPG
Algoritmos e Programação
Exercícios

1) Qual a saída do algoritmo considerando que foi digitado em x o valor 378?

```

início
  inteiro x, v[6];
  leia(x);
  se (x < 0)
    então x ← x*(-1);
  fimse
  para i de 0 até 5
    faça v[i] ← 0;
  fimpara.
  enquanto (x>0)
    faça se (x ≥ 100)
      então v[0] ← v[0]+1 ;
      x ← x-100 ;
      senão se (x ≥ 50)
        então v[1] ← v[1]+1;
        x ← x-50 ;
        senão se (x ≥ 10)
          então v[2] ← v[2]+1;
          x ← x-10 ;
          senão se (x ≥ 5)
            então v[3] ← v[3]+1;
            x ← x-5 ;
            senão v[4] ← v[4]+1;
            x ← x-1;
          fimse
        fimse
      fimse
    fimse
    v[5] ← v[5]+1 ;
  fimenquanto ;
  para i de 0 até 5
    faça escreva (v[i]) ;
  fimpara
fim
  
```

x	i	v					
		0	1	2	3	4	5
378	-						
378	0	3	1	2	1	3	10
378	1						
378	2						
378	3						
378	4						
378	5						
278							
178							
78							
28							
18							
8							
3							
2							
1							

Resposta : v =	3	1	2	1	3	10
-------------------	---	---	---	---	---	----

2) Dado o vetor CRR de caracteres abaixo.

CRR

I	U	O	T	R	E	C	A
0	1	2	3	4	5	6	7

Qual será a sua configuração depois de executados os comandos:

para I de 0 até 3 passo 1

faça $AUX \leftarrow CRR[I];$

$CRR[I] \leftarrow CRR[6-I+1];$

$CRR[6-I+1] \leftarrow AUX;$

fimpara;

i	aux	crr							
0	i	0	1	2	3	4	5	6	7
1	u	a	c	e	r	t	o	u	i
2	o								
3	t								

Resposta: a,c,e,r,t,o,u,i

3) Considerando um vetor em C, o qual foi definido como $\text{int } V[10]$, assinalar como verdadeiro (V) ou falso (F) as afirmações abaixo:

- (V) $V[0]$ contém o primeiro elemento de V;
- (F) $V[10]$ contém o último elemento de V;
- (F) a instrução $V=0$, inicializa todos os elementos de V com zeros;
- (F) a última posição de V contém um `'\0'`;
- (F) `scanf("%d", &V)` permite ler todos os elementos de V de uma só vez.

4) Considere o pseudocódigo a seguir. Qual seria o valor de saída para a matriz A de tamanho 4x4 (ou seja, $N=4$) e toda ela inicializada com o valor 3?

$t \leftarrow -10;$

para i de 0 até N-1 passo 1 faça

para j de 0 até N-1 passo 1 faça

se ($j \leq i$)

então $t \leftarrow t + A[i][j]$

fimse

fimpara

fimpara

 escreva (t) ;

fim

n	i	j	t	A				
					0	1	2	3
4	0	0	-10	0	3	3	3	3
	1	1	-7	1	3	3	3	3
	0	2	-4	2	3	3	3	3
	1	3	-1	3	3	3	3	3
	2	4	2					
	3	0	5					
	4	1	8					
		2	11					
		3	14					
		4	17					
		0	20					
		1						
		2						
		3						
		4						
		0						
		1						
		2						
		3						
		4						
		0						
		1						
		2						
		3						
		4						

Resposta: 20

5) Considere o trecho de código C sobre multiplicação de matrizes. Complete os espaços para o funcionamento adequado. As matrizes foram definidas como: A[r][s], B[s][t] e C[r][t], todas tipo float.

```
for (lin = 0; lin < r; lin++) {
    for (col = 0; col < t; col++) {
        soma = 0;
        for (k = 0; k < s; k++)
            soma = soma + A[lin][k] * B[k][col];
        C[lin][col] = soma;
    }
}
```

6) Implemente em C o programa que resolve o algoritmo da questão 4. Crie a matriz 4x4, preencha com 3 cada posição e execute o restante do algoritmo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  #define N 4
5
6  int main()
7  {
8      int A[N][N], i, j, t;
9      setlocale (LC_ALL, "Portuguese");
10
11     t = -10;
12     for(i=0; i<=N-1; i++) {
13         for(j=0; j<=N-1; j++) {
14             .....
15             A[i][j] = 3;
16         }
17     }
18
19     for(i=0; i<=N-1; i++) {
20         for(j=0; j<=N-1; j++) {
21             if(j <= i) {
22                 .....
23                 t = t + A[i][j];
24             }
25         }
26     }
27     printf("%d", t);
28 }
```

7) Faça um programa em C que leia uma matriz quadrada (NxN) e calcule a soma dos elementos da diagonal principal.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  #include <time.h>
5
6  int main(){
7      int matriz[5][5];
8      int i, j;
9      int soma=0;
10     setlocale (LC_ALL, "Portuguese");
11     srand(time(NULL));
12
13     printf( "Matriz:\n" );
14
15     for( i = 0; i < 5; i++ ) {
16         for( j = 0; j < 5; j++ ) {
17             matriz[i][j] = 5 + rand() % 5;
18             printf( "%3d", matriz[i][j] );
19
20             if( i==j ) {
21                 soma += matriz[i][j];
22             }
23         }
24         printf( "\n" );
25     }
26
27     printf("A soma da diagonal principal é: %d", soma );
28     return 0;
29 }
30
```

8) Faça um programa em C que leia uma matriz A de tamanho NxM e inverta criando uma matriz B de tamanho MxN.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  #include <time.h>
5
6  int main() {
7      int tamL = 5, tamC = 4;
8      int l, c, mat[tamL][tamC], trans[tamC][tamL];
9      setlocale (LC_ALL, "Portuguese");
10     srand(time(NULL));
11
12     for(l = 0; l < tamL; l++){
13         for(c = 0; c < tamC; c++){
14             mat[l][c] = rand() % 500;
15         }
16     }
17
18     printf("\nMatriz original:\n");
19     for(l = 0; l < tamL; l++){
20         for(c = 0; c < tamC; c++){
21             printf("%3d ", mat[l][c]);
22         }
23         printf("\n");
24     }
25
26     for(l = 0; l < tamL; l++){
27         for(c = 0; c < tamC; c++){
28             trans[c][l] = mat[l][c];
29         }
30     }
31
32     printf("\nMatriz transposta:\n");
33     for(l = 0; l < tamC; l++){
34         for(c = 0; c < tamL; c++){
35             printf("%3d ", trans[l][c]);
36         }
37         printf("\n");
38     }
39 }
```