

1. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado.

```
int main() {  
    int y, *p, x;  
    y = 0;  
    p = &y;  
    x = *p;  
    x = 4;  
    (*p)++;  
    x--;  
    (*p) += x;  
    printf ("y = %d\n", y);  
    return(0);  
}
```

- a) 0
  - b) 0+4
  - c) 4
  - d) (\*p)++
  - e) (\*&)x
2. Verifique o programa abaixo. Encontre o seu erro e sinalize a opção que corrija-o corretamente, para que seja escrito o número 10 na tela.

```
#include <stdio.h>  
int main() {  
    int x, *p, **q;  
    p = &x;  
    q = &p;  
    x = 10;  
    printf("\n%d\n", &q);  
    return(0);  
}
```

- a) printf("\n%d\n", \*p);
- b) printf("\n%d\n", \*\*q);
- c) printf("\n%d\n", \*&p);
- d) printf("\n%d\n", &p);
- e) printf("\n%d\n", &q++);

3. Considerando as afirmativas abaixo, assinale a alternativa ERRADA.

- O valor de uma variável ou expressão do tipo vetor é o endereço do elemento zero do vetor.
- “a[ ]” é um vetor qualquer, independente de tipo e tamanho.
- “pa” é um ponteiro para o mesmo tipo de “a[ ]”.

a) Após a atribuição `pa=&a[0]`; `pa` e `a` possuem valores idênticos, isto é, apontam para o mesmo endereço.

b) A atribuição `pa=&a[0]`; pode ser escrita como `pa=a`;

c) `a[i]` pode ser escrito como `*(a+i)`

d) `a+i` é o endereço do *i*-ésimo elemento do vetor `a`

e) `a++` é uma operação válida

4. Supor a declaração: `int mat[4], *p, x`; Quais expressões são válidas?

a) `p = mat + 1`;

b) `p = mat++`;

c) `p = ++mat`;

d) `x = (*mat)++`;

e) `x = *p(&mat++)`;

5. Assumindo que **pulo[]** é um vetor do tipo `int`, quais das seguintes expressões referenciam (acessam o conteúdo) da terceira posição do vetor? Considera-se que o `ptr` está inicialmente na posição “0”.

a) `*(pulo + 2)`

b) `*(pulo + 4)`

c) `pulo + 4`

d) `pulo + 2`

e) `*pulo + 3`

6. No programa, há uma função que recebe uma matriz quadrada de inteiros, por parâmetro, e verifica se esta matriz é simétrica, isto é, se cada elemento  $A[i][j]$  é igual a  $A[j][i]$ . O resultado do teste deve ser mostrado na função principal. Assinale a alternativa que corrige o seu mal funcionamento.

```
#include <stdio.h>
#include <stdlib.h>

int matriz_simetrica(int mat[3][3]);

main () {
    int matriz[3][3], i, j, res;
    for (i=0; i<3; i++){
        printf ("\nDIGITE TRES ELEMENTOS DA LINHA %d DA MATRIZ.\n", i);
        for (j=0; j<3; j++){
            scanf ("%d", &matriz[i][j]);
        }
        res=matriz_simetrica(matriz);
        if (res)
            printf ("\nMATRIZ SIMETRICA.\n\n");
        else
            printf ("\nMATRIZ NAO SIMETRICA.\n\n");
        return 0;
        system("pause");
    }

    int matriz_simetrica(mat[i], mat[j]) {
        int i, j, res=1;
        for (i=0; i<3; i++)
            for (j=0; j<3; j++)
                if (mat[i][j]!=mat[j][i])
                    res=0;
        return res;
    }
```

- a) `int matriz_simetrica(mat[i][j]=3; i++ && j++)`
- b) `int matriz_simetrica(int mat[i][j])`
- c) `int matriz_simetrica(mat[3] mat[3])`
- d) `int matriz_simetrica(int mat[i]=3; int mat[j]=3)`
- e) `int matriz_simetrica(int mat[3][3])`

7. Sobre uma função que receba dois números inteiros e retorne a soma dos números inteiros existentes entre eles. Por exemplo: Para os valores de entrada 2 e 8, a soma será  $3+4+5+6+7=25$ . Qual das alternativas corresponde ao resultado esperado?

a) <pre>int soma(int s1, s2;) {     int x,s=0;     for (s1&lt;x&lt;s2;x++)         s=x;     return s; }</pre>	b) <pre>int soma(int s1, int s2) {     int i,s=0;     for (i=s1+1;i&lt;s2;i++)         s=s+i;     return; }</pre>	c) <pre>int soma(int s1, s2;) {     int i,s=0;     for (s1&lt;x&lt;s2;x++)         s=x;     return; }</pre>
d) <pre>int soma(int s1, s2;) {     int i,s=0;     for (i=s1+1;i&lt;s2;i++)         s=s1+s2(-i);     return; }</pre>		e) <pre>int soma(int s1, int s2) {     int i,s=0;     for (i=s1+1;i&lt;s2;i++)         s=s+i;     return s; }</pre>

8. Qual a saída do programa abaixo?

```
#include <stdio.h>
void main() {
    int vet[] = {4,9,12};
    int i, *ptr;
    ptr = vet;
    for(i = 0; i < 3; i++) {
        printf("%d ",*ptr++);
    }
}
```

- a) 12 9 4
- b) 4 9 12
- c) 4 5 6
- d) 4 4 4

9. Qual a saída do programa abaixo?

```
void main() {
    int vet[] = {4,9,12};
    int i, *ptr;
    ptr = vet;
    for(i = 0; i < 3; i++) {
        printf("%d ",(*ptr)++);
    }
}
```

- a) 4 9 12
- b) 12 9 4
- c) 4 4 4
- d) 4 5 6

10. Qual a saída do seguinte programa?

```
int z;

void f(int x) {
    x = 2;
    z += x;
}

int main() {
    z = 5;
    f(z);
    printf ("z = %d\n", z);

    return 0;
}
```

- a) 5
- b) 7
- c) 10
- d) 12

11. O que a função abaixo calcula?

```
int funcaoX (int m, int n) {
    int i;
    int a = 1;

    for (i=0; i <= n; i++)
        a *= m;

    return a;
}
```

- a) mínimo múltiplo comum entre m e n
- b) fatorial de m
- c) m elevado a n
- d) fatorial de n
- e) n elevado a m

12. Qual a saída do seguinte programa?

```
#include <stdio.h>

void FUNC1() {
    int B;
    B = -100;

    printf("%d ", B);
}

void FUNC2() {
    int B;
    B = -200;
    printf("%d ", B);
}

void main() {
    int B;
    B = 10;
    printf("%d ", B);
    B = 20;
    FUNC1();
    printf("%d ", B);
    B = 30;
    FUNC2();
    printf("%d ", B);
}
```

- a) 10 20 20 30 30
- b) 10 -100 -100 -200 -200
- c) 10 -100 20 -200 30
- d) 10 20 30 -100 -200
- e) 10 -200 20 -100 30

13. O que faz o seguinte programa?

```
#include <stdio.h>

long int funcao(int x, int y) {
    long int res = 0;
    while (y != 0) {
        res += x;
        y--;
    }
    return(res);
}

int main() {
    printf("Resultado: %d", funcao(5, 600));
    return 0;
}
```

- a) Calcula a multiplicação de 5 por 600.
- b) Calcula o fatorial de 5.
- c) Calcula o fatorial de 600.
- d) Calcula 5 elevado a 600.
- d) Calcula a soma de 5 e 600.

14. Qual a saída do seguinte programa?

```
void p(int b, int c) {
    b = 10 * c;
}

int main() {
    int a;
    a = 10;
    p(a, a);
    printf("%d", a);
    return 0;
}
```

- a) 0
- b) 100
- c) 1000
- d) 10
- e) 1

15. O que imprime o programa a seguir?

```
# include <stdio.h>
main() {
    int t, i, M[3][4];
    for (t=0; t<3; ++t)
        for (i=0; i<4; ++i)
            M[t][i] = (t*4)+i+1;
    for (t=0; t<3; ++t) {
        for (i=0; i<4; ++i)
            printf ("%3d ", M[t][i]);
        printf ("\n");
    }
}
```

- a) M[0][0]=0, M[0][1]=0, M[0][2]=0, M[0][3]=0, M[1][0]=0 ... M[2][3]=6
- b) M[0][0]=0, M[0][1]=1, M[0][2]=2, M[0][3]=3, M[1][0]=1 ... M[2][3]=5
- c) M[0][0]=1, M[0][1]=2, M[0][2]=3, M[0][3]=4, M[1][0]=5 ... M[2][3]=12
- d) M[0][0]=0, M[0][1]=2, M[0][2]=4, M[0][3]=6, M[1][0]=20 ... M[2][3]=46
- e) M[0][0]=0, M[0][1]=1, M[0][2]=2, M[0][3]=3, M[1][0]=1 ... M[2][3]=5

16. O que o programa a seguir faz? Qual é o resultado obtido se a string fornecida for:

- a) "Ah! Eu to maluco!"
- b) "5\*4+(3^3)+4\*5"

```
#include <stdio.h>
#include <string.h>
#define TAM 20
int main () {
    char s[TAM];
    int c, i, j;
    for (i=0, j=strlen(s)-1; i<j; i++, j--) {
        c = s[i];
        s[i] = s[j];
        s[j] = c;
    }
    return 0;
}
```

- a) “Eu to maluco! Ah!”, "4\*5+)3^3(+5\*4”.
- b) "!oculam ot uE !hA", "4\*5+(3^3)+5\*4”.
- c) “Eu to maluco! Ah!”, "4\*5+(3^3)+4\*5”.
- d) "!oculam ot uE !hA", "5\*4+)3^3(+4\*5”.
- e) “Eu to maluco! Ah!”, "4\*5+(3^3)+5\*4”.



17. Qual a saída do seguinte programa?

```
void main () {  
    int a[] = {0, 2, 4, 6, 8}, *p = a + 3;  
    printf ("%d %d", *p, *p + 1);  
}
```

- a) 0 2
- b) 3 2
- c) 6 7
- d) 6 8

18. Qual a saída do seguinte programa?

```
#include <stdio.h>  
  
int f(int x) {  
    return x + 2;  
}  
  
int main(void) {  
    int x = 5;  
    printf("%d %d\n", f(x+2), f(f(x+2)));  
    return 0;  
}
```

- a) 7 9
- b) 7 11
- c) 5 5
- d) 5 7
- e) 9 11

19. Qual a saída do seguinte programa?

```
#include <stdio.h>

int confusao(int x, int y) {
    x = 2*x + y;
    return x;
}

int main(void) {
    int x = 2, y = 5;
    y = confusao(y, x);
    x = confusao(y, x);
    printf("%d %d\n", x, y);
    return 0;
}
```

- a) 26 12
- b) 9 9
- c) 9 12
- d) 12 22
- e) 4 5

20. Qual é o valor de x e de y, após a execução do procedimento troca() dos programas A e B?

Programa A	Programa B
<pre>// globais int x = 0; int y = 1; // procedimento troca void troca(int val_x, int val_y){     int temp;     temp = val_x;     val_x = val_y;     val_y = temp; } void main(){     troca(x,y); }</pre>	<pre>// globais int x = 0; int y = 1; // procedimento troca void troca(int * val_x, int *val_y){     int temp;     temp = *val_x;     *val_x = *val_y;     *val_y = temp; } void main(){     troca(&amp;x, &amp;y); }</pre>

- a) programa A: x=0, y=1. Programa B: x=0, y=1.
- b) programa A: x=1, y=0. Programa B: x=1, y=0.
- c) programa A: x=0, y=1. Programa B: x=1, y=0.

d) programa A: x=1, y=0. Programa B: x=0, y=1.

21. Considere as seguintes declarações:

```
void m(int x[][2], int y[][2], int z[][2]) {  
    z[0][0] = x[0][0] * y[0][0] + x[0][1] * y[1][0];  
    z[0][1] = x[0][0] * y[0][1] + x[0][1] * y[1][1];  
    z[1][0] = x[1][0] * y[0][0] + x[1][1] * y[1][0];  
    z[1][1] = x[1][0] * y[0][1] + x[1][1] * y[1][1];  
}
```

Dados:

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 3 \end{pmatrix} \quad e \quad B = \begin{pmatrix} 3 & -1 \\ 1 & 2 \end{pmatrix}$$

Simule a seguinte chamada da função acima: m(A, B, A);

a) 7 -7

0 6

d) 7 -5

0 6

b) 7 0

0 7

e) 5 0

-4 6

c) 7 0

-4 6

22. Qual a saída do programa abaixo?

```
#include <stdio.h>  
  
int i;  
  
void p1(int x) {  
    i = i + 1;  
    x = x + 2;  
    printf("%d ", x);  
}  
  
void p2(int *x) {
```

```

    i  = i + 1;
    *x = *x + 2;
    printf("%d ", *x);
}

```

```

int main() {
    int a[2];

    a[0] = 10;
    a[1] = 20;
    printf("%d %d ", a[0], a[1]);
    i = 0;
    p1(a[i]);
    printf("%d %d ", a[0], a[1]);
    a[0] = 10;
    a[1] = 20;
    i = 0;
    p2(&a[i]);
    printf("%d %d", a[0], a[1]);
    return 0;
}

```

- a) 10 20 12 12 20 12 12 20
- b) 10 20 12 10 20 12 10 20
- c) 10 20 12 10 20 22 10 22
- d) 10 20 22 10 22 10 22 22
- e) 10 20 12 10 20 12 12 20

23. O que é impresso pelo seguinte código?

```

int a = 1, b = 2, c = 3;
a += b += ++c;
printf("%5d%5d%5d\n", a, b, c);
{
    float b = 4.0;
    int c;
    a += c = 5 * b;
    printf("%5d%5.1f%5d\n", a, b, c);
}
printf("%5d%5d%5d\n", a, b, c);

```

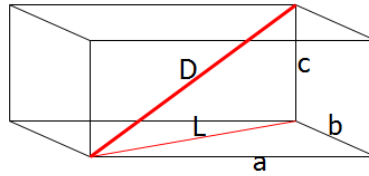
- |    |    |     |    |    |    |     |    |    |    |     |    |
|----|----|-----|----|----|----|-----|----|----|----|-----|----|
| a) | 7  | 6   | 4  | c) | 3  | 6   | 4  | e) | 3  | 6   | 4  |
|    | 27 | 4.0 | 20 |    | 23 | 4.0 | 20 |    | 23 | 4.0 | 20 |
|    | 7  | 6   | 4  |    | 23 | 6   | 4  |    | 23 | 4   | 20 |
- 
- |    |    |     |    |    |    |     |    |
|----|----|-----|----|----|----|-----|----|
| b) | 7  | 6   | 4  | d) | 7  | 6   | 4  |
|    | 27 | 4.0 | 20 |    | 27 | 4.0 | 20 |
|    | 27 | 6   | 4  |    | 27 | 4   | 20 |

24. O que será impresso pelo código abaixo?

```
int main( ) {
    char str1[ ] = { 'H', 'e', 'l', 'l', 'o' };
    char str2[ ] = "Hello";
    printf ("\n%s", str1);
    printf ("\n%s", str2);
    return 0;
}
```

- a) O programa não compila.
  - b) A string “Hello” duas vezes.
  - c) A string “Hello” + lixo de memória e a string “Hello”.
  - d) No primeiro caso nada é impresso e no segundo caso é impressa a string “Hello”.
  - e) Nada é impresso.
25. Faça uma função em C que multiplique duas matrizes (bidimensionais) de inteiros, de dimensões quaisquer e:
- a) retorne um ponteiro;
  - b) não retorne nada, a matriz resultado é passada (vazia) como parâmetro.
26. Declare protótipos de funções para:
- Escrever uma mensagem na tela e retornar nada.
  - Somar dois inteiros e retornar o resultado.
  - Ler uma string e retornar o caracter mais freqüente.
27. Escreva um programa que utilize uma função para ler três variáveis inteiras, a, b e c quaisquer e colocá-las em ordem crescente tal que o conteúdo de a < conteúdo de b < conteúdo de c. Não utilize variáveis globais. Os valores das variáveis devem ser impressos na função main().

28. Escreva um programa que utilize uma função para calcular a hipotenusa de um triângulo.
29. Faça um programa que utilize a função hipotenusa, do exercício anterior, no cálculo da diagonal  $D$  de um paralelepípedo. Imprimir o valor da diagonal via programa principal. Somente os valores das arestas  $a$ ,  $b$  e  $c$  devem ser fornecidos pelo usuário.



**Fim da lista.**