



**Questão 1** - Considerando que o tipo int, float, char e double ocupam 4, 4, 1 e 8 bytes, respectivamente, indique a quantidade de espaço em memória necessária para armazenar as variáveis:

A. char x;

**R: 1 byte.**

B. int vet[15];

**R:  $4 * 15$  bytes = 60 bytes**

C. double mat[10][3];

**R:  $8 * 10 * 3$  bytes = 240 bytes**

D. float cubo[2][5][4];

**R:  $4 * 2 * 5 * 4$  bytes = 160 bytes**

**Questão 2** - Indique os valores que serão impressos na tela ao término da execução.

```
1.  ...
2.      int a=5, b=6, *ptr;
3.      int **ptr_duplo;
4.      ptr_duplo = &ptr;
5.      ptr = &a;
6.      a = b;
7.      **ptr_duplo = 20;
8.      printf("%i %i", a, b);
9.  ...
```

```
1.  ...
2.      //linha de declaração de variáveis
3.      //linha de declaração de variáveis
4.      //ptr_duplo → |ptr|
5.      //ptr → |a|      ---      logo **ptr_duplo → a
6.      //a = b , logo a = b == 6
7.      /**ptr_duplo → a , logo se **ptr_duplo = 20, a = 20
8.      // a == 20, b == 6
9.  ...
```



**Questão 3** - Indique os valores que serão impressos na tela ao término da execução.

```
1. ...
2.     int a=3, b=4, *p1, *p2, **p_duplo;
3.     p_duplo = &p1;
4.     p1 = &a;
5.     p2 = &b;
6.     **p_duplo = *p1 **p2;
7.     *p_duplo = p2;
8.     **p_duplo = *p1 + **p_duplo;
9.     printf("a = %i\nb = %i\n", a, b);
10. ...
```

```
1. ...
2.     //linha de declaração de variável
3.     //p_duplo → |p1|
4.     //p1 → |a|
5.     //p2 → |b|
6.     /**p_duplo e *p1 → a E *p2 → b, logo a = a * b, ou a = 3 * 4 = 12
7.     /**p_duplo → p2
8.     /**p_duplo e *p2 → b E *p1 → a, logo b = a + b, ou b = 12 +4 = 16
9.     //a == 12 e b == 16
10. ...
```

**Questão 4** - Indique os valores que serão impressos na tela ao término da execução.

```
1. ...
2.     int a=6, b=7, *ptr, **ptr_duplo;
3.     ptr_duplo = &ptr;
4.     ptr = &a;
5.     **ptr_duplo = b + *ptr;
6.     ptr = &b;
7.     **ptr_duplo = a + *ptr * **ptr_duplo;
8.     printf("a=%i \t, b=%i \n", a, b);
9.     ...
```

```
1. ...
2.     //linha de declaração de variável
3.     //ptr_duplo → |ptr|
4.     //ptr → |a|
5.     /**ptr_duplo e *ptr → a, logo, a = b + a == 13
6.     //ptr → |b|
7.     /**ptr_duplo → b pois *ptr → b, logo, b = a(13) + b(7) * a(7)
8.     //a == 13 e b == 62
9.     ...
```



**Questão 5** - Indique os valores que serão impressos na tela ao término da execução.

```
1. void teste_ptr(int *a, int *b){
2.     int *p;
3.     p = a;
4.     *p = 7;
5.     b = p;
6.     *b = 8;
7. }
8.
9. void main( ){
10.     int a=1, b=2;
11.     teste_ptr(&a, &b);
12.     printf("%i - %i", a, b);
13. }
```

```
1. //inicio teste_ptr passando a == 1 e b == 2
2.     //declaração do ponteiro *p
3.     //p → |a|
4.     //*p = 7, ou *p → a = 7, logo a == 7
5.     //b → p, ou b → a
6.     //*b = 8, ou p == 8
7. //fim teste_ptr
8.
9. //inicio main
10.     //declaração das variáveis
11.     //chamada da função teste_ptr
12.     //a == 8 e b == 2
13. //fim main
```



**Questão 6** - Indique os valores que serão impressos na tela ao término da execução.

```
1. void teste_ptr(int *c, int *b, int a){
2.     int *p;
3.     p = &a;
4.     *p = *c + * b;
5.     *c = *p + 2;
6.     *b = a + *c;
7. }
8.
9.
10. int main(){
11.     int A=2, B=3, C=4;
12.     teste_ptr(&A, &B, C);
13.     printf("A = %i \n", A);
14.     printf("B = %i \n", B);
15.     printf("C = %i \n", C);
16. }
```

```
1. //inicio teste_ptr passando *a de C, *b de B e cópia de a de C
2.     //declaração do ponteiro *p
3.     //p → |a|, nesse caso p → cópia de C
4.     //*p → a || cópia de C, logo cópia C = *c(A(2)) + *b(B(3)) = 5
5.     //*c → A, logo A = *p(a || cópia de C(5)) + 2 = 7
6.     //*b → B, logo B = a(5) + *c(A(7)) = 12
7. //fim teste_ptr
8.
9.
10. //inicio main
11.     //declaração de variáveis
12.     //chamada da função teste_ptr
13.     //A == 7
14.     //B == 12
15.     //C == 4
16. //fim main
```



**Questão 7** - Indique os valores que serão impressos na tela ao término da execução.

```
1. void teste_ptr(int *a, int *b){
2.     int *p;
3.     p = a;
4.     *p = 1 + *a + *b;
5.     b = p;
6.     *b = *p + *a + 2;
7. }
8.
9.
10. void main(){
11.     int a=2, b=4;
12.     teste_ptr(&a, &b);
13.     printf("a = %i \nb=%i \n", a, b);
14. }
```

```
1. //inicio teste_ptr passando ponteiro *a, e *b
2. //declaração ponteiro *p
3. //p → |a|
4. //*p → a, logo a = 1 + a(2) + b(4) = 6
5. //b → |p|
6. //*b → a, logo a = a + a + 2, ou 6 + 6 + 2, 14
7. //fim teste_ptr
8.
9.
10. //inicio main
11. //declaração das variáveis
12. //chamada da função teste_ptr
13. //a == 14 e b == 4
14. //fim main
```



**Questão 8** - Seja *vet* um vetor de 4 elementos: TIPO *vet*[4]. Supor que depois da declaração, *vet* esteja armazenado no endereço de memória 4092 (ou seja, o endereço de *vet*[0]). Supor também que na máquina usada uma variável do tipo *char* ocupa 1 byte, do tipo *int* ocupa 2 bytes, do tipo *float* ocupa 4 bytes e do tipo *double* ocupa 8 bytes. Qual o valor de *vet*+1, *vet*+2 e *vet*+3 se

- a) *vet* for declarado como *char*?
- b) *vet* for declarado como *int*?
- c) *vet* for declarado como *float*?
- d) *vet* for declarado como *double*?

Qual o valor de *vet*+1, *vet*+2 e *vet*+3 se:

**&vet[0] == 4092**

A. *vet* for declarado como *char*?

**R:     &vet[1] == 4093**

**&vet[2] == 4094**

**&vet[3] == 4095**

B. *vet* for declarado como *int*?

**R:     &vet[1] == 4094**

**&vet[2] == 4096**

**&vet[3] == 4098**

C. *vet* for declarado como *float*?

**R:     &vet[1] == 4096**

**&vet[2] == 4100**

**&vet[3] == 4104**

D. *vet* for declarado como *double*?

**R:     &vet[1] == 4100**

**&vet[2] == 4108**

**&vet[3] == 4116**