



**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA**



**SISTEMAS EMBARCADOS I (SEMB I)
RESOLUÇÃO LISTA 01**

Aluno: Vítor Carvalho Marx Lima

Matrícula: 11821ECP015

Uberlândia – 11/08/2023

OBS: Os códigos para cada uma das questões se encontram dentro do zip, cada questão tem seu próprio ".c" com a resolução do código, quando necessário, e a resposta em formato de comentário.

Questão 01)

Resposta: O valor da variável "a" no final será de 8. Inicialmente todas as variáveis "a", "b", "c" e "d" foram declaradas como inteiros, sem nenhuma atribuição, tendo os seus respectivos valores iguais aos que se encontram no endereço de memória atribuídos a cada uma delas (lixo de memória, por serem variáveis de escopo local). Depois, foram atribuídos para as variáveis locais "d", "c" e "b" os valores 3, 2 e 4 respectivamente. Em seguida "d" foi atribuído o novo valor de "c+b", igual a "2+4", logo "d" agora possui o valor de 6. Agora "a" foi atribuído o valor de "d+1", igual a "6+1", logo "a" possui o valor 7. E por último, "a" foi atribuído o novo valor de "a+1", igual a "7+1", sendo o valor final da variável "a" igual a 8.

Questão 02)

Resposta:

Erro 1 - Linha 6 -> "float f,g;" ao invés de "float f,g;" separando as variáveis por ponto e vírgula ao invés de usar vírgula.

Erro 2 - Linha 7 -> "char h = 65;" ao invés de "char h[2] = '65';" Iniciando uma variável do tipo char com um inteiro ao invés de usar uma string e sem informar o tamanho do array.

Erro 3 – Linha 9 -> "a, b = 10, 2;" ao invés de "a = 10, b = 2;" ao atribuir valores para as variáveis "a" e "b"

Erro 4 - Linha 11 -> "scanf("%f", g);" ao invés de "scanf("%f", &g);" para armazenar o dado corretamente no endereço de memória da variável "g", e também não há nenhuma mensagem informando que variável que deve ser inserida pelo usuário, nem o tipo de dado que deve ser inserido, sendo que esse é um comando de interação com o usuário.

Questão 03)

Resposta:

```
# include <stdio.h>

int main(){
    int a,b,c;
    a = 14, b = 73;
    c = a - b;
    printf("O resultado da subtracao de %d por %d (%d - %d) e igual a %d", a, b,
a, b, c);
    return 0;
}
```

Questão 04)

Resposta: O valor final da variável "a" no código é de -29. O código declara as variáveis "a", "b" e "h" no escopo local, tendo elas o valor inicial que se encontra no endereço de memória a elas atribuído (lixo de memória). Em seguida são atribuídos os valores 10, -15 e 'A' para "a", "b" e "h" respectivamente, sendo o valor atual de "a" igual a 10. Depois é atribuído o valor de "b" para "a", sendo o novo valor de "a" igual a -15. Adiciona-se uma unidade em "b", tendo agora "b" o valor de -14. Atribui-se a "a" o valor atual de "a" somado ao valor atual de "b", sendo $a = (-15) + (-14)$, sendo o valor final de "a" igual a -29.

Questão 05)

Resposta:

Os valores impressos serão 2 (inteiro), 2.5 (float) e 2.5 (float) respectivamente. Isso ocorre pois, no primeiro print, o resultado é a divisão de dois inteiros "a=5" e "b=2" resultando em um outro inteiro arredondado para baixo (floor) igual a 2. Já no segundo valor impresso, é o resultado da divisão de um inteiro "a=5" e um float "d=2.0", sendo assim, o resultado da divisão é um float, resultando em 2.5. E no último print, está o resultado da divisão entre 2 floats, "c=5.0" e "d=2.0", sendo o resultado da divisão também um float igual a 2.5.

Questão 06)

Resposta:

```
# include <stdio.h>

int main(){

    int num_lst[3];
    int mult_res = 1, sum_res = 0;
    char lst[][20] = { "primeiro", "segundo", "terceiro" };

    int i;
    for(i=0;i<3;i++){
        printf("Por favor, insira um valor inteiro para o %s numero:", lst[i]);
        scanf("%d", &num_lst[i]);
        sum_res += num_lst[i];
        mult_res *= num_lst[i];
    }

    printf("Os numeros informados foram %d, %d e %d\nO resultado da soma desses numeros e igual a %d\nE o resultado da multiplicacao desses numeros e igual a %d", num_lst[0], num_lst[1], num_lst[2], sum_res, mult_res);

    return 0;
}
```

Questão 07)

Resposta:

```
# include <stdio.h>

int main(){

    int n=0, quadrado=0, cubo=0;

    printf("Por favor, insira um valor inteiro para que seja calculado seu quadrado e seu cubo:");
    scanf("%d", &n);

    quadrado = n*n;
    cubo = n*n*n;

    printf("O numero informado foi %d\nO quadrado deste numero e %d\nE o cubo deste numero e %d", n, quadrado, cubo);

    return 0;
}
```

Questão 08)

Resposta: Os valores impressos serão 7, 6 e 6 respectivamente. O programa declara as variáveis "a", "b" e "c", inicializando "a" com o valor 5. Em seguida, "b" é atribuído o valor de "++a", ou seja, aqui a operação de adicionar uma unidade na variável "a", ocorre antes da atribuição do valor de "a" para "b", fazendo com que "a" que possuía o valor 5, receba o valor 6, e então "b" receba esse novo valor de 6. Depois "c" recebe a atribuição de "a++", ou seja, "c" recebe o valor atual de "a" antes da incrementação de uma unidade em "a", logo "c" recebe o valor 6 e então "a" recebe uma unidade, sendo o valor final de "a" igual a 7.

Questão 09)

Resposta: Os valores impressos serão 1, 0 e 1 respectivamente. Nos prints do programa, o resultado é um inteiro igual a 0 ou 1, dependendo do resultado da álgebra booleana que foi inserida para checagem. No primeiro caso, foi inserido "9 > 3", o que resulta em verdadeiro, pois 9 é maior que 3, logo o resultado dessa comparação é o inteiro 1. No segundo caso, foi inserida a comparação "(3*4)/2 != (2*3)", ao desenvolver a conta, temos "6!=6", o que é uma comparação falsa, pois 6 é igual a 6, e não diferente de 6, logo, o resultado é falso, imprimindo o inteiro 0. E por último, a comparação é entre duas variáveis "a", com o valor inicial de 1, e "b", com o valor inicial de -1. A comparação foi "a!=b", ou seja, comparando se "a" é diferente de "b", e como "a=1" e "b=-1", é uma verdade, pois 1 é diferente de -1, imprimindo o valor de 1.

Questão 10)

Resposta:

Primeira maneira (usando uma terceira variável auxiliar):

```
# include <stdio.h>

int main(){
    int x, y, aux;
    x = 10, y = 15, aux = 0;
    printf("Os valores das variaveis x e y antes de serem trocadas -> x = %d e y = %d\n", x, y);

    aux = x;
    x = y;
    y = aux;

    printf("Os valores das variaveis x e y depois de serem trocadas -> x = %d e y = %d", x, y);

    return 0;
}
```

Segunda maneira (sem usar uma terceira variável auxiliar):

```
# include <stdio.h>

int main(){
    int x, y, aux;
    x = 10, y = 15;
    printf("Os valores das variaveis x e y antes de serem trocadas -> x = %d e y = %d\n", x, y);

    (x ^= y), (y ^= x), (x ^= y);

    printf("Os valores das variaveis x e y depois de serem trocadas -> x = %d e y = %d", x, y);

    return 0;
}
```