# Lista de Exercícios – 5

## *Disciplina Linguagem de Programação - I*

André Rossi Korol – 01810067

**1)**

**a) int s = 1.75;**

1, pois como a variável é declarada como int, a parte decimal é descartada.

**b) int k = a > b;**

1, pois a > b é verdadeiro.

**c) printf("%i", x);**

255, pois o especificador de formatos %i detecta a base (decimal, hexadecimal, ou octal) do valor a ser substituido, e ff (hexadecimal) é 255 na base decimal.

**d) float b = 1 / 2;**

Erro de tipos conflitantes para 'b', pois b já havia sido declarado como int e recebido um valor em 'int b = 90;'.

**2)**

```c
#include <stdio.h>

int main()
{
    int a = 5; // faltando ';' no final
    int y = 0;
    int n = 0;
    scanf("%d", &n); // faltando '&' para referenciar o endereço
                     // da variável 'n' na memória
    while (a <= n)
    // ; ';' desnecessário
    {
        y = a++;
        printf("%x", y);
    }
}
```

Para n = 5, o resultado será 5.

**3)**

```c
#include <stdio.h>

int main()
{
    int num;
    scanf("%d", &num);
    for (int i = 1; i <= 10; i++)
    {
        printf("%d x %d = %d\n", num, i, num * i);
    }

    return 0;
}
```

**4)**

```c
#include <stdio.h>

int main()
{
    int numeros[5];
    int i;

    for (i = 0; i < 5; i++)
    {
        scanf("%d", &numeros[i]);
    }

    for (i = 4; i >= 0; i--)
    {
        printf("%d\n", numeros[i]);
    }

    return 0;
}
```

**5)**

```c
#include <stdio.h>

int main()
{
    unsigned int num;
    scanf("%u", &num);

    if (num & 1)
        printf("%d eh impar\n", num);
    else
        printf("%d eh par\n", num);

    return 0;
}
```

**6)**

```c
#include <stdio.h>

int main()
{
    for (int i = 0; i < 1000; i++)
        printf("%d %d\n", i + 1, 1000 - i);

    return 0;
}
```

**7)** `for (; ;); printf("FEAU\n);`

Não está correto pois há um ';' a mais depois do loop infinito `for (; ;)`.

**8)**

```c
#include <stdio.h>

int main()
{
    int i;
    float valores[3];
    float media;

    for (i = 0; i < 3; i++)
    {
        scanf("%f", &valores[i]);
        media += valores[i];
    }
    media /= 3;

    printf("Media aritmetica = %.2f\n", media);

    return 0;
}
```

**9)**

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

// funcao para comparar dois floats (a > b)
static int compareFloat(const void *a, const void *b)
{
    return (*(float *)a > *(float *)b) - (*(float *)a < *(float *)b);
}

int main()
{
    float primeiroNumero, segundoNumero, terceiroNumero, quartoNumero;
    float amplitudeMedia;

    puts("Entre o primeiro numero:");
    scanf("%f", &primeiroNumero);
    puts("Entre o segundo numero:");
    scanf("%f", &segundoNumero);
    puts("Entre o terceiro numero:");
    scanf("%f", &terceiroNumero);
    puts("Entre o quarto numero:");
    scanf("%f", &quartoNumero);

    float numeros[4] = {primeiroNumero, segundoNumero, terceiroNumero,
quartoNumero};
    // ordena o array de floats
    qsort(numeros, 4, sizeof(float), compareFloat);
    amplitudeMedia = (numeros[3] + numeros[0]) / 2;
    printf("Amplitude media = %.2f\n", amplitudeMedia);

    return 0;
}
```

**10)**

```c
#include <stdio.h>

int main()
{
    int matrixSize = 4;
    int matrix[matrixSize];
    int i, determinant;

    for (i = 0; i < matrixSize; i++)
        scanf("%d", &matrix[i]);

    determinant = matrix[0] * matrix[3] - matrix[1] * matrix[2];
    printf("%d\n", determinant);

    return 0;
}
```

**11)**

```c
#include <stdio.h>
#include <time.h>

int main()
{
    time_t rawtime;
    struct tm *timeinfo;
    int currentYear, birthYear, age;

    time(&rawtime);
    timeinfo = localtime(&rawtime);
    currentYear = timeinfo->tm_year + 1900;
    puts("Entre o ano de nascimento:");
    scanf("%d", &birthYear);
    age = currentYear - birthYear;

    if (age >= 35)
        puts("Pode ser candidato a governador");
    if (age >= 29)
        puts("Pode concorrer ao senado");
    if (age >= 16)
        puts("Pode votar");
    else
        puts("Idade nao permitida para votar ou ser candidato");

    return 0;
}
```

**12)**

```c
// primeiro fluxograma
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    int f = 1;
    do
    {
        f *= n;
        n--;
    } while (n > 1);

    printf("%d\n", f);

    return 0;
}
```

```c
// segundo fluxograma
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    int f = 1;
    while (n > 1)
    {
        f *= n;
        n--;
    }

    printf("%d\n", f);

    return 0;
}
```

O algoritmo do segundo fluxograma funcionaria de forma mais adequada pois se um valor negativo for dado como entrada para n, a saída será sempre consistente, retornando o valor inicial de f, que no caso é 1, uma vez que as expressões dentro da estrutura de repetição não seriam executadas.