

# Lista de Exercícios – 9

---

## *Disciplina Linguagem de Programação - I*

André Rossi Korol – 01810067

1)

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

int main()
{
    int t = 3;
    double x, N, v = 2.9;

    x = log2(v) / (t * log2(2));
    printf("x = %g\n", x);

    for (; t <= 10; t++)
    {
        v = pow(2, x * t);
        N = 2000 * v;
        printf("N(%d) = %g\n", t, N);
    }

    return 0;
}
```

2)

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

int main()
{
    double x, y, z;

    puts("Entre a altura x:");
    scanf("%lg", &x);
    puts("Entre a distancia y:");
    scanf(" %lf", &y);

    z = sqrt(pow(x, 2) + pow(y, 2));
    printf("Distancia que Carlos tera que percorrer = %g\n", z);

    return 0;
}
```

3)

```
#include <stdio.h>

int main()
{
    int n, raizesReais = 0, raizesImaginas = 0;
    float numReal, temp, raiz;

    puts("Entre a quantidade (n) de numeros reais a ser entrada:");
    scanf("%d", &n);
    while (n-->0)
    {
        puts("Entre um numero real:");
        scanf("%f", &numReal);
        if (numReal < 0)
        {
            printf("Numero imaginario raiz(%.0f)\n", numReal);
            raizesImaginas++;
            numReal *= -1; // converte para positivo
        }
        else
            raizesReais++;

        raiz = numReal / 2;
        temp = 0;
        while (raiz != temp)
        {
            temp = raiz;
            raiz = (numReal / temp + temp) / 2;
        }
        printf("raiz(%.0f) = %.0f\n", numReal, raiz);
    }
    printf("Ocorreram %d raizes reais e %d raizes imaginarias\n",
raizesReais,
raizesImaginas);

    return 0;
}
```

4)

```
#include <stdio.h>

int main()
{
    int m, n;
    scanf("%d", &m);
    scanf(" %d", &n);
    for (int i = 1; i <= m; i++)
        for (int j = 1; j <= n; j++)
            printf("%d, %d\n", i, j);
    return 0;
}
```

5)

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

int main()
{
    float perimetroTotal, area, lado;
    puts("Entre a area do galinheiro (em metros quadrados):");
    scanf("%f", &area);
    lado = sqrt(area);
    perimetroTotal = lado * 4;
    printf("Serao necessarios %.0f metros de tela para cercar o  
galinheiro\n",
           perimetroTotal);

    return 0;
}
```

6)

a) O modelo matemático usado para encontrar o resultado do gráfico é o do logaritmo natural  $\log(x)$ .

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

int main()
{
    for (double x = 0; x <= 10; x += 0.5)
        printf("log(%.1f) = %lf\n", x, log(x));

    return 0;
}
```

7)

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

int main()
{
    float a, b, c, x1, x2;

    puts("Entre o coeficiente a:");
    scanf("%f", &a);
    puts("Entre o coeficiente b:");
    scanf("%f", &b);
    puts("Entre o coeficiente c:");
    scanf("%f", &c);

    x1 = (-b + sqrt(b * b - 4 * a * c)) / (2 * a);
    x2 = (-b - sqrt(b * b - 4 * a * c)) / (2 * a);
    printf("As raizes sao: x1 = %.0f e x2 = %.0f\n", x1, x2);

    return 0;
}
```

8)

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

int main()
{
    float cabo, base, altura;

    puts("Entre a medida C dos cabos:");
    scanf("%f", &cabo);
    puts("Entre a distancia B dos ganchos ate a base da torre:");
    scanf(" %f", &base);

    altura = sqrt(cabo * cabo - base * base);
    printf("A altura da torre eh igual a %.2f metros\n", altura);

    return 0;
}
```

9)

```
#include <stdio.h>

int main()
{
    double x, serie;

    puts("Entre o valor de x:");
    scanf("%lg", &x);

    // Usando expansao da serie de Maclaurin
    serie = 1 / (1 - x);
    printf("O valor da serie geometrica eh igual a %g\n", serie);

    return 0;
}
```

10)

a) A funcao é  $f(x) = x^e$ .

b)

```
#include <stdio.h>
#include <math.h>
#include <complex.h>

#define E 2.71828

int main()
{
    char inflexao = 0;
    double complex z;
    double y;

    puts("Os valores (pontos) da funcao sao:");
    for (int x = -10; x <= 10; x++)
    {
        z = cpow(x, E);
        y = cabs(z);
        if (creal(z) < 0)
            y *= -1;
        if (!inflexao && cimag(z) == 0)
        {
            inflexao++;
            printf("Ponto de inflexao: ");
        }
        printf("(%d, %g)\n", x, y);
    }

    return 0;
}
```

c) O ponto de inflexão é o ponto (0, 0).