

# Lista de Exercícios – 7

---

## *Disciplina Linguagem de Programação - I*

André Rossi Korol – 01810067

1)

```
#include <stdio.h>

int main()
{
    int primeiro, segundo, i;

    puts("Entre o primeiro valor:");
    scanf("%d", &primeiro);
    puts("Entre o segundo valor:");
    scanf("%d", &segundo);

    if (primeiro < segundo)
        for (i = primeiro; i <= segundo; i++)
            (i != segundo) ? printf("%d, ", i) : printf("%d\n", i);
    else if (primeiro > segundo)
        for (i = primeiro; i >= segundo; i--)
            (i != segundo) ? printf("%d, ", i) : printf("%d\n", i);
    else
        puts("valores iguais");

    return 0;
}
```

3)

```
#include <stdio.h>

int main()
{
    float valor1, valor2, resultado;
    char operador;

    puts("Entre o primeiro valor:");
    scanf("%f", &valor1);
    puts("Entre um operador(+, -, *, /):");
    getchar(); // Consome o '\n' do input anterior
    operador = getchar();
    puts("Entre o segundo valor:");
    scanf("%f", &valor2);

    switch (operador)
    {
        case '+':
            resultado = valor1 + valor2;
            break;

        case '-':
            resultado = valor1 - valor2;
            break;

        case '*':
            resultado = valor1 * valor2;
            break;

        case '/':
            resultado = valor1 / valor2;
            break;

        default:
            printf("Operador '%c' invalido.\n", operador);
            return -1;
    }

    printf("%.2f %c %.2f = %.2f\n", valor1, operador, valor2, resultado);

    return 0;
}
```

4)

```
#include <stdio.h>

int main()
{
    int primeiroValor, segundoValor;

    puts("Entre o primeiro valor:");
    scanf("%d", &primeiroValor);
    puts("Entre o segundo valor:");
    scanf("%d", &segundoValor);

    for (int i = 1; i <= primeiroValor; i++)
        for (int j = 1; j <= segundoValor; j++)
            printf("%d,%d\n", i, j);

    return 0;
}
```

5)

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

int isPrime(int number)
{
    if (((!(number & 1)) && number != 2) || (number < 2) || (!(number % 3)
&& number != 3))
        return 0;

    for (int k = 1; 36 * k * k - 12 * k < number; ++k)
        if ((!(number % (6 * k + 1))) || (!(number % (6 * k - 1))))
            return 0;

    return 1;
}

int main()
{
    int n, currentNumber, largestPrime, primeNumbers = 0;
    int *numbers = (int *)malloc(sizeof(int));

    puts("Entre a quantidade (n) de numeros a serem entrados:");
    scanf("%d", &n);

    for (int i = 1; i <= n; i++)
    {
        printf("Entre o %d° numero:\n", i);
        scanf("%d", &currentNumber);
        *(numbers + i - 1) = currentNumber;
        numbers = (int *)realloc(numbers, sizeof(int) * i + 1);
    }

    for (int j = 0; j < n; j++)
    {
        currentNumber = *(numbers + j);
        if (isPrime(currentNumber))
        {
            primeNumbers++;
            largestPrime = (!j || currentNumber > largestPrime) ?
currentNumber : largestPrime;
        }
    }

    if (primeNumbers > 0)
    {
        printf("%d dos numeros entrados ", primeNumbers);
        (primeNumbers == 1) ? printf("eh primo\n") : printf("sao
primos\n");
        printf("O maior primo entrado eh o %d\n", largestPrime);
    }
}
```

```

    }
    else if (n > 0)
        printf("Nenhum dos numeros entrados eh primo\n");

    free(numbers);

    return 0;
}

```

6)

```

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

int main()
{
    unsigned int m, n, i;
    float c, a = 0, b = 0;
    float *array = (float *)malloc(sizeof(float));

    puts("Entre um numero real para n:");
    scanf("%u", &n);
    m = 0;
    printf("Entre um numero real, menor que %u, para m:\n", n);
    scanf("%u", &m);
    if (m >= n)
    {
        printf("%d nao eh menor que %d!\n", m, n);
        return -1;
    }

    puts("Entre um numero real para a constante c:");
    scanf("%f", &c);

    for (i = 1; i <= n - m + 1; i++)
    {
        printf("Entre o %d° numero do vetor a:\n", i);
        scanf("%f", (array + i - 1));
        array = (float *)realloc(array, sizeof(float) * i + 1);
    }

    for (i = 0; i < n - m + 1; i++)
    {
        a += c * (*(array + i));
        b += *(array + i);
    }
    b *= c;

    printf("Σ(cai) = %.2f\ncΣai = %.2f\n", a, b);

    free(array);
}

```

```
    return 0;
}
```

7)

```
#include <stdio.h>

int main()
{
    // The Fibonacci sequence : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
    int n, tmp, a = 0, b = 1;

    scanf("%d", &n);
    for (int i = 2; i < n; i++)
    {
        tmp = a + b;
        a = b;
        b = tmp;
    }
    printf("%d\n", b);

    return 0;
}
```

9)

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

typedef struct Candidata
{
    int nota;
    char *nome;
} Candidata;

int main()
{
    int candidatasCount = 0, maiorNota = 0;
    Candidata *concurso = (Candidata *)malloc(sizeof(Candidata));
    char ans = 'Y';
    Candidata *candidata, *vencedora;

    while (tolower(ans) != 'n')
    {
        candidata = (concurso + candidatasCount);
        candidata->nome = malloc(sizeof(char) * 32);

        printf("Entre o nome da candidata:\n");
        scanf("%s", candidata->nome);

        printf("Entre a nota da candidata %s:\n", candidata->nome);
        scanf("%d", &candidata->nota);
        if (candidata->nota < 0)
            candidata->nota = 0;
        else if (candidata->nota > 100)
            candidata->nota = 100;

        concurso = (Candidata *)realloc(concurso, sizeof(Candidata) *
1));

        printf("Deseja continuar adicionando candidatas? [Y/n]\n");
        scanf(" %c", &ans);
    }

    for (int i = 0; i < candidatasCount; i++)
    {
        candidata = (concurso + i);
        if (candidata->nota >= maiorNota)
        {
            vencedora = candidata;
            maiorNota = vencedora->nota;
        }
    }
}
```

```

printf("A candidata vencedora foi a %s, com uma nota igual a %d!\n",
      vencedora->nome, vencedora->nota);

free(concurso);

return 0;
}

```

10)

```

#include <stdio.h>

int main()
{
    int n, x;
    int riddleNumbers[30];
    riddleNumbers[0] = 1;
    riddleNumbers[1] = 3;

    for (int i = 2; i < 30; i += 2)
    {
        riddleNumbers[i] = riddleNumbers[i - 1] + 5;
        riddleNumbers[i + 1] = riddleNumbers[i] * 3;
    }

    scanf("%d", &n);
    while (n--)
    {
        scanf(" %d", &x);
        printf("%d\n", riddleNumbers[x - 1]);
    }

    return 0;
}

```