

Lista 2

1. Faça um programa que imprima um menu de 4 pratos na tela e uma quinta opção para sair do programa. O programa deve imprimir o prato solicitado. O programa deve terminar quando for escolhido a quinta opção.
2. Faça um programa que lê dois números inteiros positivos a e b . Utilizando laços, o seu programa deve calcular e imprimir o valor a^b .
3. Faça um programa que lê um número n e que compute e imprima o valor

$$\sum_{i=1}^n i.$$

OBS: Não use fórmulas como a da soma de uma P.A.

4. Faça um programa que lê um número n e imprima os valores entre 2 e n , que são divisores de n .
5. Faça um programa que lê um número n e imprima os valores

$$\sum_{i=1}^j i$$

para j de 1 até n , um valor por linha.

6. No exemplo dos números primos visto em aula, não precisamos testar todos os números entre $2, \dots, (n-1)$, para verificar se dividem ou não n . Basta testarmos até $n/2$. Por que? Qual o maior divisor possível de n ? Na verdade basta testarmos os números $2, \dots, \sqrt{n}$. Por que?
7. Considere o programa para determinar se uma sequência de n números digitados pelo usuário está ordenado ou não. Faça o programa usando uma variável contadora.
8. Faça um programa em C que calcule o máximo divisor comum de dois números m, n . Você deve utilizar a seguinte regra do cálculo do mdc com $m \geq n$

$$\text{mdc}(m, n) = m \text{ se } n = 0$$

$$\text{mdc}(m, n) = \text{mdc}(n, m \% n) \text{ se } n > 0$$

9. Escreva um programa que lê um número n , e então imprime o menor número primo que é maior ou igual n , e imprime o maior primo que é menor ou igual a n .
10. O que será impresso pelo programa abaixo? Assuma que o valor de D na declaração de x é o valor do último dígito do seu RA.

```

int main() {
    int x = 5+D, y = 0;
    do {
        y = (x % 2) + 10 * y;
        x = x / 2;
        printf("x = %d, y = %d\n", x, y );
    } while (x != 0);
    while (y != 0) {
        x = y % 100;
        y = y / 10;
        printf("x = %d, y = %d\n", x, y );
    }
}

```

11. Escreva um programa para ler n de números do tipo `float` e imprimir quantos deles estão nos seguintes intervalos: $[0 \dots 25]$, $[26 \dots 50]$, $[51 \dots 75]$ e $[76 \dots 100]$. Por exemplo, para $n = 10$ e os seguintes dez números 2.0, 61.5, -1.0, 0.0, 88.7, 94.5, 55.0, 3.1415, 25.5, 75.0, seu programa deve imprimir:

```

Intervalo [0..25]: 3
Intervalo [26..50]: 0
Intervalo [51..75]: 3
Intervalo [76..100]: 2

```

12. Elabore um programa em C para computar a raiz quadrada de um número positivo. Use a idéia abaixo, baseada no método de aproximações sucessivas de Newton. O programa deverá imprimir o valor da vigésima aproximação.

Seja Y um número, sua raiz quadrada é raiz da equação

$$f(x) = x^2 - Y.$$

A primeira aproximação é $x_1 = Y/2$. A $(n + 1)$ -ésima aproximação é

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

13. Aponte os erros de implementação existentes no código em C, a seguir, desenvolvido com o intuito de calcular e imprimir o *fatorial* de um número inteiro não-negativo.

```
#include <stdio.h>

int main(){
    int valor;
    scanf("%d", &valor);

    int n = valor;
    float fatorial = 1;

    if (n > -1)
        while (n > 0)
        {
            fatorial *= n;
            n--;
        }
    print("O fatorial de %d e igual a %d\n", valor, fatorial);
else
    print("Nao existe fatorial de %d\n", n);
return 0;
}
```

14. Implemente um programa que compute todas as soluções de equações do tipo

$$x_1 + x_2 + x_3 + x_4 = C$$

. Melhore o seu programa com as seguintes idéias.

- Fixado x_1 , os valores possíveis para x_2 são $0, \dots, C - x_1$. Fixado x_1 e x_2 , os valores possíveis para x_3 são $0, \dots, C - x_1 - x_2$. Fixados x_1, x_2 , e x_3 , então x_4 é unicamente determinado.

15. Na transformação decimal para binário, modifique o programa para que este obtenha o valor binário em uma variável inteira, ao invés de imprimir os dígitos um por linha na tela. **Dica:** Suponha $n = 7$ (111 em binário), e você já computou $x = 11$, para "inserir" o último dígito 1 em x você deve fazer $x = x + 100$. Ou seja, você precisa de uma variável acumuladora que armazena as potências de 10: 1, 10, 100, 1000 etc.

16. Faça um programa que leia um inteiro n (no máximo 50) e imprima uma saída da forma:

```
1
 2
   3
    .
     .
      .
       n
```

17. Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se $n = 6$):

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
```

18. Faça um programa que leia um número n e imprima n linhas na tela com o seguinte formato (exemplo se $n = 6$):

```
+ * * * * *
* + * * * *
* * + * * *
* * * + * *
* * * * + *
* * * * * +
```

19. Um jogador da Mega-Sena é supersticioso, e só faz jogos em que o primeiro número do jogo é par, o segundo é ímpar, o terceiro é par, o quarto é ímpar, o quinto é par e o sexto é ímpar. Faça um programa que imprima todas as possibilidades de jogos que este jogador supersticioso pode jogar.

20. Um automóvel possui um medidor que informa de tempos em tempos a um velocímetro digital a distância percorrida (em **metros**) e o tempo gasto (em **segundos**). O velocímetro acumula algumas medições e eventualmente realiza cálculos para atualizar o valor da velocidade (em **km/h**) mostrada no display do automóvel. Desenvolva um programa que:

- Leia uma sequência de pares de valores na forma: distância percorrida<espaço>tempo gasto<enter>;
- Acumule os valores e quando receber um par de zeros, imprime na saída a velocidade média em **km/h**, zerando os acumuladores;
- O programa é encerrado quando recebe um par de -1;

```
20 0.5
50 0.5
0 0
Saída: 252.00
20 0.5
120.5 2.0
0 0
Saída: 202.32
-1 -1
Encerra programa
```