Fundamentos de Programação

Prof. Ítalo Mendes da Silva Ribeiro

Lista 3

1 – Escreva os números de 100 até 250.

```c
int main()
{
    int i;

    for(i = 100; i <= 250; i++){

        printf("%i, ", i);
    }

    return 0;
}
```

2 – Escreva os números de 3190 até 3310.

3 – Escreva os números pares de 200 até 351.

4 – Escreva os números divisíveis por 5 de 550 até 690.

5 – Escreva os números pares de 175 até 51.

6 – Desenvolva um programa que leia um número inteiro e positivo n. Apresente os números inteiros de 1 a n.

---

7 – Mostre a soma dos números ímpares de 410 até 551.

```c
int main()
{
    int i, soma = 0;

    for(i = 411; i <= 551; i += 2){

        soma += i;
    }

    printf("soma = %i \n", soma);   // soma = 34151

    return 0;
}
```

8 – Informa a soma dos números divisíveis por 5 de 55 até 123.

9 – Escreva a soma dos números múltiplos de 3 e por 5 entre 356 e 455.

10 – Informe o produto dos números pares de 43 até 77.

11 – Leia um número e mostre o seu fatorial. Exemplo, fatorial de 5! $= 5 \times 4 \times 3 \times 2 \times 1 = 120$

12 – Faça um programa que peça dois números, base e expoente. Mostre o primeiro número elevado ao segundo número. Não utilize a função de potência da linguagem.

13 – Apresente a média dos números ímpares de 50 até 85.

14 – Exiba a média dos números divisíveis por 3 e por 4 menores que 175.

---

15 – Exiba o valor de S dado por:

$$S = \frac{1}{5} + \frac{3}{10} + \frac{5}{15} + \frac{7}{20} + \ldots + \frac{17}{45}$$

```c
int main()
{
    int i, j = 5;
    float s = 0;

    for(i = 1; i <= 17; i += 2){

        s += (float)i / j;
        j += 5;
    }

    printf("S = %f \n", s);   // S = 3.034206

    return 0;
}
```

16 – Informe o valor de R dado por:

$$R = \frac{1}{4} + \frac{2}{5} + \frac{3}{6} + \frac{4}{7} + \ldots + \frac{20}{23}$$

17 – Exiba o valor de V dado por:

$$V = \frac{17}{7} + \frac{15}{9} + \frac{13}{11} + \frac{11}{13} + \frac{9}{15} + \frac{7}{17}$$

18 – Receba um valor n > 0 e mostre o valor do somatório:

$$-1 + 2 - 3 + 4 - 5 + 6 - \ldots + n$$

19 – Exiba o valor de F dado por:

$$F = 7 + \frac{20}{121} + \frac{25}{114} + \frac{30}{107} + \frac{35}{100} + \ldots + \frac{80}{37}$$

20 – Mostre o resultado da soma:

$$1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \ldots + \frac{1}{999} - \frac{1}{1000}$$

---

21 – Crie um algoritmo que receba vários números inteiros e positivos. Imprima o produto dos números ímpares digitados e a soma dos pares. O algoritmo encerra quando um número menor ou igual a zero é recebido pelo programa.

```c
int main()
{
    int n, produtoImpares = 1, somaPares = 0;

    while(1){

        printf("Escreva um numero: \n");
        scanf("%i", &n);

        if(n <= 0){
            break;
        }

        if(n % 2 == 0){
            // soma numeros pares
            somaPares += n;
        }else{
            // multiplica numeros impares
            produtoImpares *= n;
        }
    }

    printf("Soma dos pares: %i \n", somaPares);
    printf("Produto dos impares: %i \n", produtoImpares);

    return 0;
}
```

22 – Desenvolva um programa que tenha o seguinte menu e realize os cálculos necessários de acordo com a opção do menu escolhida. O programa encerrará apenas quando o usuário escolher a opção de Sair.

1 - triplo de um número
2 - 15% de um número
3 - Cubo de um número
4 - Sair.

23 – Crie um algoritmo que atualiza a senha. O usuário digitará a nova senha, depois deverá repetir a nova senha. Caso a nova senha repetida não seja igual à primeira senha, mostre que as senhas não conferem e peça que o usuário digite a nova senha mais uma vez, repetindo o processo até que o usuário digite a nova senha duas vezes iguais.

24 – Faça um programa que leia 5 números e informe o maior número.

25 – Escreva um algoritmo que leia um número não determinado de pares de valores [m,n], onde m e n são inteiro e positivos. Deve-se ler um par de cada vez. Mostre a soma de todos os números inteiros de m e n (inclusive m e n). A entrada de pares terminará quando m for igual a n.

26 – Implemente um programa que leia n números inteiros e positivos, até que um número negativo ou nulo seja informado. Mostre o menor número par digitado pelo usuário.

27 – Os preços e os produtos de uma loja estão na tabela seguinte. Faça um programa que simule um caixa. O programa lê os itens comprados pelo cliente, dentre aqueles existentes na loja e informa o preço da compra. A entrada de itens comprados, deve encerrar pela escolha de alguma opção do programa. Depois de ler os itens comprados, o programa deve receber o pagamento do cliente e informar o troco caso exista.

Produto	Valor
Camisa	R$ 70
Calça	R$ 90
Chinela	R$ 15
Cinto	R$ 35

---

28 – Supondo que a população de um país A seja de 80.000 habitantes com uma taxa anual de crescimento de 3% e que a população de um país B seja de 200.000 habitantes, com uma taxa de crescimento de 1.5%. Faça um programa que escreva o número de anos necessários, para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.

```c
int main()
{
    int a = 80000;
    int b = 200000;
    int tempo = 0;

    while(a < b){

        a += a * 0.03;
        b += b * 0.015;
        tempo++;
    }

    printf("Tempo foi de %i anos \n", tempo); // 63 anos

    return 0;
}
```

29 – Dois ciclistas A e B estão andando em uma pista de ciclismo com 2 Km de comprimento com velocidades de 10 m/s e 15 m/s, respectivamente. Escreva um algoritmo que determine iterativamente, o tempo que levará para que esses dois ciclistas A e B, se encontrem em um mesmo ponto, sabendo que eles partiram em uma mesma direção, porém em sentido contrário. O algoritmo também mostrará a distância que cada um percorreu.

30 – Chico tem 1,50m e cresce 2 centímetros por ano, enquanto Juca tem 1,10m e cresce 3 centímetros por ano. Construa um algoritmo que imprima quantos anos serão necessários para que Juca seja maior que Chico.

---

31 – Crie uma tabela que mostra os números múltiplos de 3, como exibido abaixo:

3	6	9	12	15
18	21	24	27	30
33	36	39	42	45

```c
int main()
{
    int n = 3, linhas, colunas;

    for(linhas = 1; linhas <= 3; linhas++){

        for(colunas = 1; colunas <= 5; colunas++){

            printf("%i \t", n);
            n += 3;
        }

        printf("\n");
    }

    return 0;
}
```

32 – Crie um algoritmo que receba um inteiro n e imprima a imagem como mostrado abaixo, usando estrutura de repetição, escrevendo apenas um asterisco por vez. (OBSERVAÇÃO: não existem espaços em branco à esquerda dos asteriscos. Existe um espaço em branco entre os asteriscos de uma mesma linha.)

Exemplo de saída do programa para n = 5, ou seja, 5 linhas e 10 colunas de asteriscos.

```
**********
**********
**********
**********
**********
```

33 – Crie um algoritmo que receba um inteiro n e imprima a imagem como mostrado abaixo, usando estrutura de repetição, escrevendo apenas um asterisco por vez. (OBSERVAÇÃO: não existem espaços em branco à esquerda dos asteriscos. Existe um espaço em branco entre os asteriscos de uma mesma linha.)

Exemplo de saída do programa para n = 5, ou seja, 5 linhas e 10 colunas alternadas com e sem asteriscos.

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

34 – Implemente um programa que imprima a imagem como abaixo, usando estrutura de repetição, escrevendo apenas um asterisco ou espaço em branco por vez. (OBSERVAÇÃO: não existem espaços em branco à esquerda dos asteriscos da primeira coluna.)

** ****
*** ***
**** **
***** *

35 – Mostre a imagem como mostrado abaixo, usando estrutura de repetição, escrevendo apenas um sinal de mais por vez. (OBSERVAÇÃO: não existem espaços em branco à esquerda dos sinais de mais e não escreve o conteúdo dentro dos parênteses.)

+++
(linha em branco)
++
(linha em branco)
+

36 – Mostre na tela, os dias de um mês como abaixo, usando estrutura de repetição.

Janeiro

D	S	T	Q	Q	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

---

37 – A série de Fibonacci é uma seqüência de termos que tem como os 2 primeiros termos, respectivamente, os números 0 e 1. A partir daí, os demais termos são formados seguindo uma certa regra. A série de Fibonacci pode ser vista a seguir:

0 1 1 2 3 5 8 13 21...

Descubra a regra que gera a seqüência da série de Fibonacci e escreva um algoritmo que gere os N (solicitados pelo usuário) primeiros termos desta série. Além disso, escreva a soma destes termos.

38 – Construa um algoritmo que receba um número e verifique se ele é um número triangular. Um número é triangular quando é resultado do produto de três números consecutivos. Exemplos de números triangulares:
Exemplo 1: 24 = 2 x 3 x 4.
Exemplo 2: 210 = 5 x 6 x 7.

39 – Faça um programa que avalie o desempenho de um piloto de corrida automobilística. O programa deve receber a posição de chegada e o tempo (em minutos) de 5 corridas, e mostrar as seguintes informações:

a) A pontuação do piloto após as cinco corridas, onde a quantidade de pontos recebidos por cada corrida, segue a tabela seguinte:

Posição de Chegada	Pontos Ganhos
1º	10
2º	8
3º	5
Outras posições	0

b) A quantidade de corridas em que o piloto chegou em posição igual ou maior que 2^{o} e o tempo de corrida foi menor que 120 minutos;

c) A pior posição de corrida;

d) O maior tempo de corrida.