

## Lista de Exercícios 2:

Observações gerais: utilize funções com variáveis locais sempre que possível, documente e faça indentação de código.

1. Faça um programa que verifique se um determinado número é primo. O número a ser testado deve ser fornecido pelo usuário do programa. O programa deve imprimir a mensagem “NUMERO PRIMO” caso o número seja primo, e “NUMERO NAO PRIMO” caso contrário.

2. Faça um programa que calcule e imprima a soma dos  $n$  primeiros números naturais ímpares. O usuário do programa deve fornecer quantos números devem ser somados.

3. Faça um programa que imprima os  $n$  primeiros termos da série de Fibonacci, lembrando que esta série é definida da seguinte forma:

$$\text{termo}_1 = 1$$

$$\text{termo}_2 = 1$$

$$\text{termo}_n = \text{termo}_{n-1} + \text{termo}_{n-2}$$

O usuário do programa deve fornecer o número de termos, e estes devem ser impressos separados por um espaço. Por exemplo, se o usuário pedir os 6 primeiros termos da série de Fibonacci, o programa deve ter a seguinte saída:

1 1 2 3 5 8

4. Escreva um programa que sorteie uma letra do alfabeto de ‘a’ a ‘z’ e o usuário é perguntado qual é essa letra. O programa deve informar se a letra é maior ou menor e quando o usuário a acertou.

5. Escreva um programa em C para ler um valor X e um valor Z (se Z for menor que X deve ser lido um novo valor para Z). Contar quantos números inteiros devemos somar em sequência (a partir do X inclusive) para que a soma ultrapasse a Z o mínimo possível. Escrever o valor final da contagem.

Exemplo:

X	Z	Resposta
3	20	5 (3+4+5+6+7=25)
2	10	4 (2+3+4+5=14)
30	40	2 (30+31=61)

6. Fazer um programa que sorteie um número de 0 a 100 e que permita que o usuário (sem conhecer o número sorteado) tente acertar. Caso não acerte, o programa deve imprimir uma mensagem informando se o número sorteado é maior ou menor que a tentativa feita. Ao acertar o número, o programa deve imprimir a quantidade de tentativas feitas.

7. Escreva um programa que calcule o salário semanal de um trabalhador. As entradas são o número de horas trabalhadas na semana e o valor da hora. Até 40 h/semana não se acrescenta nenhum adicional. Acima de 40h e até 60h há um bônus de 50% para essas horas. Acima de 60h há um bônus de 100% para essas horas.

**8.** O número 3025 possui a seguinte característica:  $30 + 25 = 55 \rightarrow 55 * 55 = 3025$ . Fazer um programa para obter todos os números de 4 algarismos com a mesma característica do número 3025.

**9.** Implementar a função INVERTE que recebe um número unsigned int como parâmetro e retorna este número escrito ao contrário. Ex: 431 <-> 134.

**10.** O maior divisor comum (MDC) de dois números inteiros é o maior número inteiro que faz uma divisão exata (com resto zero) desses dois números. Por exemplo, o MDC de 32 e 40 é 8, o MDC de 9 e 27 é 9 e o MDC de 13 e 25 é 1. Implemente uma função que receba como parâmetros dois números inteiros x e y e calcule o MDC desses números. Essa função deve ter o seguinte protótipo:

```
int mdc (int x, int y);
```

**11.** Faça um algoritmo para ler dois números inteiros e calcular e mostrar o valor resultante se elevarmos a base representada pelo primeiro à potência representada pelo segundo. Dica: busque qual a função que faz a potenciação.