

1. Leia uma cadeia de caracteres no formato "DD/MM/AAAA" e copie o dia, mês e ano para 3 variáveis inteiras. Antes disso, verifique se as barras estão no lugar certo, e se DD, MM AAAA são numéricos.
2. O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, 'A' seria substituído por 'D', 'B' se tornaria 'E', e assim por diante. Implemente um programa que faça uso desse Código de César (3 posições), entre com uma string e retorne a string codificada. Exemplo: String: a ligeira raposa marrom saltou sobre o cachorro cansado Nova string: D OLJHLUD UDSRVD PDUURP VDOWRX VREUH R FDFKRUUR FDQVDGR.
3. Faça um programa que leia uma palavra (máximo de 50 letras) e some 1 no valor ASCII ' de cada caractere da palavra. Imprima a string resultante.
4. Crie um programa que compara duas strings (não use a função strcmp).
5. Faça um programa que conte quantas vezes um dado caractere aparece na string S1. Esse caractere desse ser informado pelo usuário;
6. Faça um jogo da forca, sendo permitido seguir este algoritmo.

### **Algoritmo: Jogo da Forca**

1. **Definição da Palavra a ser Adivinhada:**
  - Armazene a palavra a ser adivinhada em uma variável (**palavra**) com o valor a ser definido por você.
  - Determine o número de letras da palavra usando e armazene esse valor em **tamanho**.
2. **Inicialização das Variáveis:**
  - **letra**: armazenará a letra que o jogador vai chutar.
  - **palavra\_acertada**: um array de caracteres que armazena as letras que o jogador acertou.
  - **num\_tentativas**: define o número de tentativas restantes para o jogador (inicialmente zero).
  - **qtd\_ch\_404**: conta quantas vezes a letra chutada pelo jogador não é encontrada na palavra.
3. **Apresentação do Jogo:**
  - Exibe uma mensagem de boas-vindas ao jogo da forca e informa ao jogador o número de letras da palavra.
4. **Desenho Inicial do Campo de Letras:**
  - Exibe uma linha de traços ( \_ ) equivalente ao número de letras da palavra, representando as letras desconhecidas.
5. **Laço Principal de Tentativas:**

- O laço `while` continua até que o jogador erre 6 vezes (`num_tentativas < 6`) ou adivinhe a palavra.
  - **Entrada de Letra:**
    - Solicite ao jogador que digite uma letra (`letra`).
  - **Comparação e Atualização:**
    - Itere pela palavra para verificar se a letra chutada está presente em alguma posição.
      - Se a letra estiver presente, armazene-a na posição correspondente em `palavra_acertada`.
      - Se a letra não estiver na posição atual, incremente `qtd_ch_404` (indicando que não houve coincidência).
  - **Avaliação do Chute:**
    - Se `qtd_ch_404` for igual ao `tamanho` da palavra, significa que a letra chutada não está na palavra:
      - Incrementa `num_tentativas` e exibe o número de tentativas restantes.
    - Restaure `qtd_ch_404` para zero, para a próxima letra.
  - **Exibição do Progresso Atual:**
    - Exiba a palavra com letras acertadas ou `_` nas posições ainda não descobertas.
6. **Condição de Vitória:**
- Se o comprimento da `palavra_acertada` atingir o tamanho da palavra original, o jogador venceu o jogo, e a mensagem "VOCÊ VENCEU A FORÇA" é exibida.
7. **Finalização:**
- Exiba o total de tentativas feitas (`num_tentativas`).