

Prova 2 – Mascarando strings – Valor: 10,0

Nome: _____ Data: __/__/__

Parte 1: _____ Parte 2: _____ **Total:** _____

Prova individual.
Pontuação máxima: 10,0.

Parte 1: escrita a mão.
Total = (Parte1+Parte2)/2

Parte 2: prática.

Parte 1

(3,0) 1 – Mostra Palavra com Mascara. Escreva a função `void showWordWithMask(char * word, int * mask)` para escrever os caracteres de uma string `word` de acordo com uma mascara `mask`. A mascara indica quais caracteres devem aparecer e quais devem estar escondidos. Toda vez que um caracter for escondido pela mascara, escreva um underline. Exemplos:

Mostra "Programacao" com mascara [0,0,0,0,0,0,0,0,0,0,0]: "_____"
Mostra "Programacao" com mascara [1,1,1,1,1,1,1,1,1,1,1]: "Programacao"
Mostra "Programacao" com mascara [0,1,0,1,0,1,0,1,0,1,0]: "_r_g_a_a_a_"
Mostra "PROVA" com mascara [0,1,0,1,0,1,0,1,0,1,0]: "_R_V_"

(3,0) 2 – Atualiza Mascara. Escreva a função `int updateMask(char * word, int * mask, char letter)` para inverter o valor de todos os índices da mascara `mask` toda vez que a letra `letter` for encontrada na string `word`. A função retorna um inteiro com a quantidade de valores alterados na mascara. Exemplos:

Atualiza mascara [0,0,0,0,0,0,0,0,0,0,0] em "Programacao" a partir de 'a':
> Modificou 3 valores. Mascara atualizada: [0,0,0,0,0,1,0,1,0,1,0]

Atualiza mascara [1,1,1,1,1,1,1,1,1,1,1] em "Programacao" a partir de 'r':
> Modificou 2 valores. Mascara atualizada: [1,0,1,1,0,1,1,1,1,1,1]

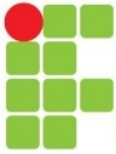
Atualiza mascara [1,1,1,1,1,1,1,1,1,1,1] em "Programacao" a partir de 'X':
> Modificou 0 valores. Mascara atualizada: [1,1,1,1,1,1,1,1,1,1,1]

(2,0) 3 – Conta valores na mascara. Escreva a função `int contValueMask(int * mask, int tam, int value)` para contar na mascara `mask` de tamanho `tam` quantos valores iguais a `value` foram encontrados. Exemplos:

Conta valor 1 na mascara [1,1,1,1,1] da tamanho 5: 5
Conta valor 0 na mascara [1,1,1,1,1] da tamanho 5: 0
Conta valor 1 na mascara [0,0,0,1,1] da tamanho 5: 2

(2,0) 4 – Inicializa mascara Escreva a função `void initMask(int * mask, int tam, int value)` para configurar todos os valores da mascara `mask` de tamanho `tam` com um valor `value`. Exemplos:

Reset mascara [0,1,0,1,0,1,0,1,0,1,0] de tamanho 11 com 1: [1,1,1,1,1,1,1,1,1,1,1]
Reset mascara [0,1,0,1,0,1,0,1,0,1,0] de tamanho 11 com 0: [0,0,0,0,0,0,0,0,0,0,0]
Reset mascara [0,1,0,1,0,1,0,1,0,1,0] de tamanho 11 com 9: [9,9,9,9,9,9,9,9,9,9,9]



Nome: _____ Data: __/__/____

Parte 2 – Jogo da Forca

(10,0) Questão Única. Escreva a função `main()` para desenvolver um jogo da forca em uma versão simplificada:

- Utilize no desenvolvimento todas as 4 funções da parte 1;
- A palavra que o jogador terá que adivinhar é seu nome;
- Você pode ficar lendo letras indefinidamente até o jogador acertar toda a palavra
- Não precisa determinar quando o jogador perde;

Exemplo para a palavra “SEUNOME” e março (mês 03).

```
Diga uma letra
> E

LETRAS CERTAS
  _ E _ _ _ _ E

Word=SEUNOME errado=0 certo=1
contValueMask=2 strlen=7
```

```
Diga uma letra
> S

LETRAS CERTAS
  S E _ _ _ _ E

Word=SEUNOME errado=0 certo=2
contValueMask=3 strlen=7
```

Segunda Chance. Não tirou 10 na parte 1? **Fique com nota 10** ao

- Utilizar a função `void showForca(int erros);`
- Determinar se o jogador perdeu OU ganhou no final;

Pontuação Intermediária. Não vai dar tempo para escrever o jogo da forca na função `main()`? As seguintes demonstrações também valem nota:

- (1,5) i – Escreva a função `main()` demonstrando o uso de `showWordWithMask()`
(1,5) ii – Escreva a função `main()` demonstrando o uso de `updateMask()`
(1,5) iii – Escreva a função `main()` demonstrando o uso de `contValueMask()`
(1,5) iv – Escreva a função `main()` demonstrando o uso de `initMask()`

```

void limpa(){
    getchar();
    system("cls|clear\n");
}

void showForca(int erros){
    switch(erros){
        case 1:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf("    | \n");
            printf("    | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        case 2:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf(" |  | \n");
            printf("    | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        case 3:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf(" /|  | \n");
            printf("    | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        case 4:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf(" /|\ | \n");
            printf("    | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        case 5:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf(" /|\ | \n");
            printf(" /   | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        case 6:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf(" 0  | \n");
            printf(" /|\ | \n");
            printf(" / \ | \n");
            printf("    | \n");
            printf("===== \n");
            break;
        default:
            printf(" +---+ \n");
            printf(" |  | \n");
            printf("    | \n");
            printf("    | \n");
            printf("    | \n");
            printf("    | \n");
            printf("===== \n");
            break;
    }
    printf("\n\n");
}

```