



**DISCIPLINA :** 09038/09039 – Introdução à Computação

**PROFESSOR:** Equipe SE/9

**VISTO:** \_\_\_\_\_

**ANO / CURSO:** 1º Básico (CFG) / 2º Ano Básico (CG)

**DATA DA PROVA:** 16/06/2020

**TIPO DE PROVA:** VE

**DURAÇÃO:** 90 min

**INÍCIO:** 08:30 h

**CONSULTA:**

X

LIVRE

RESTRITA

PROIBIDA

## INSTRUÇÕES

- As questões estarão entregues quando o aluno fizer o *upload* do código-fonte no Portal EAD. **O ALUNO SÓ DEVE APERTAR O BOTÃO DE SUBMISSÃO FINAL A COMANDO DO PROFESSOR. NENHUM ALUNO ESTÁ AUTORIZADO A SUBMETER A PROVA SEM A CIÊNCIA E ORIENTAÇÃO DO PROFESSOR.**
- O Portal EAD só aceitará submissões de arquivos de extensão .c (código-fonte em C).
- O arquivo do código-fonte de ter o seguinte formato: questão, letra que identifica a turma a qual o aluno pertence e o nome de guerra, separados por “\_”. Exemplos: Questão 1 do Aluno Fulano da Turma A: Q1\_A\_FULANO.c / Questão 2 do Aluno BELTRANO da Turma B: Q2\_B\_BELTRANO.c
- O aluno deve indicar, como comentário no início de cada arquivo de código-fonte, o sistema operacional em que programou, ou seja, se foi Windows, Linux ou Mac.
- O programa deve ser escrito de maneira portátil, portanto não serão aceitas extensões não padronizadas da linguagem.
- Lembre-se que o padrão de especificação de C adotado na disciplina é o C99. Logo, a compilação deve ser feita usando o parâmetro -std=c99
- Toda a declaração de função, variável ou parâmetro deve ter o tipo explicitamente declarado.
- Os professores tirarão dúvidas pelo Google Meet nos primeiros 15 minutos da prova e permanecerão online durante toda a prova para auxiliá-los na submissão.

### 1ª Questão (5,0 pontos)

Uma brincadeira infantil muito conhecida é a comunicação na língua do P, acrescentando o fonema *pê* antes de cada sílaba, como uma forma de código para dificultar que outras pessoas entendam a conversa. Assim, a frase “vamos ao cinema” seria dita da seguinte forma: “pê-va pê-mos pê-no pê-ci pê-ne pê-ma”

Em uma adaptação da **língua do P para mensagens eletrônicas**, podemos acrescentar a letra ‘p’ (p minúsculo) antes de cada caractere alfabético (letra) minúsculo ou maiúsculo de uma mensagem. Os exemplos abaixo mostram duas mensagem codificadas e a respectiva mensagem decodificada utilizando esta versão da língua do P. Note que o acréscimo da letra ‘p’ não se aplica aos espaços em branco e aos caracteres de pontuação.

Mensagem Codificada	Mensagem Decodificada (Original)
pVpapmpops papo pcpipnpepmpa?	Vamos ao cinema?
pFpupi pbpepm pnpa ppprpopvpa.	Fui bem na prova.

O programa ao fim do enunciado desta questão, recebe uma mensagem de até 500 caracteres do usuário, a codifica e decodifica usando a língua do P para mensagens eletrônicas, conforme o exemplo acima.

Copie e cole o código do programa em sua IDE (Integrated Development Environment) e implemente o corpo das seguintes funções:

void codifica(char msg\_original[501], char msg\_codificada[1001]); - recebe a mensagem original e gera a mensagem codificada.

void decodifica(char msg\_codificada[1001], char msg\_decodificada[501]); - recebe a mensagem codificada e gera a mensagem decodificada.

CÓDIGO (copie e cole na sua IDE):

```
//Sistema operacional: INSIRA O SISTEMA OPERACIONAL UTILIZADO AQUI
```

```
#include<stdio.h>
```

```
//PROTOTIPOS
```

```
void codifica(char msg_original[501], char msg_codificada[1001]);
```

```
void decodifica(char msg_codificada[1001], char msg_decodificada[501]);
```

```
//MAIN
```

```
int main() {
```

```
    //Declaracao de variaveis
```

```
    char msg_original[501], msg_codificada[1001], msg_decodificada[501];
```

```
    //Leitura da mensagem
```

```
    printf("Digite uma mensagem de ate 500 caracteres:\n");
```

```
    scanf("%[^\\n]", msg_original);
```

```
    //Codificacao da mensagem
```

```
    codifica(msg_original, msg_codificada);
```

```
    //Impressao da mensagem codificada
```

```
    printf("\nMENSAGEM CODIFICADA:\n%s", msg_codificada);
```

```
    //Decodificacao da mensagem
```

```
    decodifica(msg_codificada, msg_decodificada);
```

```
    //Impressao da mensagem decodificada
```

```
    printf("\n\nMENSAGEM DECODIFICADA:\n%s", msg_decodificada);
```

```
    return 0;
```

```
}
```

```
//OUTRAS FUNCOES
void codifica(char msg_original[501], char msg_codificada[1001]) {
    //IMPLEMENTE A CODIFICACAO DA MENSAGEM AQUI
}

void decodifica(char msg_codificada[1001], char msg_decodificada[501]) {
    //IMPLEMENTE A DECODIFICACAO DA MENSAGEM AQUI
}
```

## 2ª Questão (5,0 pontos)

Considere a seguinte série:

$$T_1 = T_2 = T_3 = 1;$$

$$T_n = T_{n-1} + 2 * T_{n-2} + 3 * T_{n-3}, \text{ para } n \geq 4$$

O programa ao fim do enunciado desta questão, recebe um número natural do usuário e imprime o valor de  $T_n$  calculado pelas abordagens recursiva e iterativa.

Copie e cole o código do programa em sua IDE (Integrated Development Environment) e implemente o corpo das seguintes funções:

int calcula\_recursivo(int n); - calcula e retorna o valor de  $T_n$  pela abordagem recursiva.

int calcula\_iterativo(int n); - calcula e retorna o valor de  $T_n$  pela abordagem iterativa.

CÓDIGO (copie e cole na sua IDE):

```
//Sistema operacional: INSIRA O SISTEMA OPERACIONAL UTILIZADO AQUI
#include<stdio.h>
```

```
//PROTOTIPOS
```

```
int calcula_recursivo(int n);
```

```
int calcula_iterativo(int n);
```

```
//MAIN
```

```
int main() {
```

```
    //Declaracao de variaveis
```

```
    int n;
```

```
    //Leitura de n
```

```
    printf("Insira um numero natural: ");
```

```
    scanf("%d", &n);
```

```
    //Impressao dos valores de Tn
```

```
    printf("\nT(%d) = %d (RECURSIVO)", n, calcula_recursivo(n));
```

```
    printf("\nT(%d) = %d (ITERATIVO)", n, calcula_iterativo(n));
```

```
    return 0;
```

```
}
```

```
//OUTRAS FUNCOES
```

```
int calcula_recursivo(int n) {
```

```
    //IMPLEMENTE O CALCULO RECURSIVO AQUI
```

```
}
```

```
int calcula_iterativo(int n) {
```

```
    //IMPLEMENTE O CALCULO ITERATIVO AQUI
```

```
}
```