

UNIFEI	Universidade Federal de Itajubá Instituto de Engenharia de Sistemas e Tecnologias da Informação-IESTI
ECOP11A – Técnicas de Programação 10º Exercício Profª.Drª.Thatyana de Faria Piola Seraphim	

1) Faça um programa, usando a linguagem C, onde o usuário informa um substantivo e em seguida é gerado seu plural. Abaixo segue uma simplificação das regras:

- Acrescenta-se (es) aos terminados em (r, n, s e z): pares (par), Hífenes (hífen), ingleses (inglês), pazes (paz);
- Troca-se por (is) aos terminados por (l): funerais (funeral), pastéis (pastel);
- Troca-se por (ns) aos terminados em (m): tons (tom), afins (afim);
- Troca-se por (ces) aos terminados por (x): Cálices (cálx), apêndices (apêndix);
- Troca-se por (oes) aos terminados em (ao): corações(coração), eleições (eleição);
- Demais situações acrescenta (s): gatos (gato), aves (ave);

Observação: não considere acento no caso (ão), para não ter que se preocupar com qual tabela de codificação que está sendo usada no compilador C e qual está no sistema operacional.

Abaixo segue um trecho da implementação usando switch e as funções de string.h (**strlen**, **strcpy**, **strncpy**, **strcat**) para primeiro e segundo caso, sendo que o quarto e o quinto caso é tratado na instrução default.

```
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[]){
    char sub[50], plural[50];
    int ult;
    printf("Digite palavra: ");
    scanf("%s", sub);
    ult=strlen(sub)-1;
    switch(sub[ult]){
        case 'r':
        case 'n':
        case 's':
        case 'z':
            strcpy(plural, sub);
            strcat(plural, "es");
            break;
        case 'l':
            strncpy(plural, sub, ult);
            plural[ult]='\0';
            strcat(plural, "is");
            break;
        default:
            if(sub[ult-1]=='a'&&sub[ult]=='o'){
            }else{
            }
            break;
    }
    printf("Substantivo=%s\nPlural=%s\n", sub, plural);
    return 0;
} //end main
```

2) Faça um programa, usando a linguagem C, que realiza a leitura de uma frase com no máximo 50 caracteres em maiúsculo. Em seguida, essa frase será criptografada, ou seja, cada caractere da frase deverá ser substituído pelo caractere que está três posições à sua frente na tabela ASCII. Os três últimos caracteres do alfabeto (X, Y, Z) serão substituídos pelos três primeiros caracteres (A, B, C), sendo que qualquer outro símbolo não é substituído. Imprima na tela a criptografia resultante. **OBS:** utilize a função **strlen()** para saber o tamanho da frase digitada; utilize o comando **gets** para fazer a leitura da frase; se for usar o comando **scanf** para leitura da frase, é necessário usar o seguinte: **scanf("%[^\\n]", frase);** onde **[^\\n]** indica que a string será lida até

que seja encontrado o terminador de nova linha (enter). Abaixo segue um exemplo:

frase: Tecnicas de Programacao

cripto: Whfqlfdv gh Surjudpdfdr

3) Faça um programa, usando a linguagem C, que declara uma matriz de dimensão 10x10 de caracteres chamada **matriz**. Essa matriz deve ser inicializada utilizando a matriz exemplo abaixo. Em seguida, declare um vetor de caracteres de dimensão 4 chamado **animal** e o inicialize com as palavras {"gato", "vaca", "bode", "anta"}. O programa deve procurar em qual linha e coluna começa e qual linha e coluna termina cada palavra do vetor animal. A ocorrência da palavra na matriz pode estar na horizontal, horizontal invertida, vertical ou vertical invertida. É necessário fazer a verificação se nenhum teste é realizado fora dos limites da matriz. Após a verificação de limites, imprima as seguintes informações: nome do animal, linha e coluna de início e linha e coluna de término. Abaixo segue um exemplo da matriz 10x10 de caracteres que procura as palavras gato, vaca, bode e anta.

	0	1	2	3	4	5	6	7	8	9
0	Q	F	H	Q	P	L	P	W	S	Y
1	A	B	N	W	A	Z	O	Q	A	T
2	Z	O	U	E	S	O	T	A	G	R
3	W	D	J	R	D	X	I	L	M	E
4	S	E	M	T	F	C	U	K	N	W
5	X	V	I	V	A	C	A	J	B	A
6	E	T	K	Y	G	V	Y	H	V	T
7	D	G	O	U	H	B	T	G	C	N
8	C	B	L	I	J	N	R	F	X	A
9	R	Y	P	O	K	M	E	D	Z	Q

GATO INICIO 2,8 FIM 2,5

VACA INICIO 5,3 FIM 5,6

BODE INICIO 1,1 FIM 4,1

ANTA INICIO 8,9 FIM 5,9

Abaixo segue a inicialização dos vetores e matriz de caracteres.

```
#include<stdio.h>
#include<string.h>
int main(int argc, char *argv[]){
    char m[10][10]={
        {'Q','F','H','Q','P','L','P','W','S','Y'},
        {'A','B','N','W','A','Z','O','Q','A','T'},
        {'Z','O','U','E','S','O','T','A','G','R'},
        {'W','D','J','R','D','X','I','L','M','E'},
        {'S','E','M','T','F','C','U','K','N','W'},
        {'X','V','I','V','A','C','A','J','B','A'},
        {'E','T','K','Y','G','V','Y','H','V','T'},
        {'D','G','O','U','H','B','T','G','C','N'},
        {'C','B','L','I','J','N','R','F','X','A'},
        {'R','Y','P','O','K','M','E','D','Z','Q'};
    char pal[4][5]={"GATO","VACA","BODE","ANTA"};
    return 0;
}
```