MC102: Algoritmos e Programação de Computadores
Lista de Exercícios sobre Cadeias de Caracteres

Implemente as operações a seguir com cadeias de caracteres (*strings*). Lembre-se que *strings* em C possuem um marcador no fim para indicar seu tamanho.

1. Escreva uma função que inverte uma dada string1 e guarda o resultado em string2:

`void InverteString(char string1[], char string2[], int n);`

```
void InverteString(char string1[], char string2[])
{
  int tamanho = strlen(string1);
  int i=0;

  for (i=0; i<tamanho; i++) {
    string2[(tamanho-1)-i] = string1[i];
  }
}
```

2. Implemente uma função que receba uma *string* como parâmetro e retorne como resultado o número de vogais encontradas.

`int conta_vogais(char* str);`

Atenção: devem ser contadas vogais em minúsculo ou maiúsculo.

```
int conta_vogais(char* str)
{
  int i=0, conta=0;

  while (str[i] != 0) {
    if (str[i] == 'a' ||
        str[i] == 'e' ||
        str[i] == 'i' ||
        str[i] == 'o' ||
        str[i] == 'u') {
      conta++;
    }
    i++;
  }

  return conta;
}
```

3. Implemente uma função que receba como parâmetro uma string e um caractere, e retorne como resultado o número de ocorrências desse caractere na string passada como parâmetro.

`int conta_char (char* str, char letra);`

```
int conta_char (char* str, char letra)
{
  int i=0, conta=0;

  while (str[i] != 0) {
    if (str[i] == letra) {
      conta++;
    }
    i++;
  }

  return conta;
}
```

4. Implemente uma função que receba uma string como parâmetro e altere nessa string as ocorrências de caracteres maiúsculos para minúsculos e vice-versa.

```
void inverte_letra (char* str);
```

```
void inverte_letra (char* str)
{
  int i=0;
  int a_para_A = 'A' - 'a';

  while (str[i] != 0) {
    if (str[i] >= 'a' && str[i] <= 'z') {
      str[i] = str[i] + a_para_A;
    }
    else if (str[i] >= 'A' && str[i] <= 'Z') {
      str[i] = str[i] - a_para_A;
    }
    i++;
  }
}
```

5. Implemente uma função que receba como parâmetro uma string e dois caracteres (original e novo), e substitua nessa string todas as ocorrências do caractere original pelo caractere novo. Por exemplo, se essa função receber como parâmetro a string "Estruturas" e os caracteres 't' e 'd', após a sua chamada a string deve passar a conter a seqüência de caracteres "Esdruduras".

```
void troca_letra (char* str, char original, char novo);
```

```
void troca_letra (char* str, char original, char novo)
{
  int i=0;

  while (str[i] != 0) {
    if (str[i] == original) {
      str[i] == novo;
    }
    i++;
  }
}
```

6. Implemente uma função que receba uma string como parâmetro e substitua todas as letras por suas sucessoras no alfabeto. Por exemplo, a string "Casa" seria alterada para "Dbtb".

```
void shift_string (char* str);
```

Obs.: A letra 'z' deve ser substituída pela letra 'a' (e 'Z' por 'A'). Caracteres que não forem letras devem permanecer inalterados.

```
void shift_string (char* str)
{
  int i=0;

  while (str[i] != 0) {
    if      (str[i] == 'z') str[i] = 'a';
    else if (str[i] == 'Z') str[i] = 'a';
    else if (str[i] >= 'a' && str[i] < 'z') str[i]++;
    else if (str[i] >= 'A' && str[i] < 'Z') str[i]++;
    i++;
  }
}
```

7. Implemente uma função que receba uma string como parâmetro e substitua as ocorrências de uma letra pelo seu oposto no alfabeto, isto e, 'a' para 'z', 'b' para 'y', 'c' para 'x', etc. Caracteres que nao forem letras devem permanecer inalterados.

```
void string_oposta (char* str);
```

```
void string_oposta (char* str)
{
  int i=0;

  while (str[i] != 0) {
    if (str[i] >= 'a' && str[i] <= 'z') {
      str[i] = 'z' - str[i] + 'a';
    }
    else if (str[i] >= 'A' && str[i] <= 'Z') {
      str[i] = 'Z' - str[i] + 'A';
    }
    i++;
  }
}
```

8. Implemente uma função que receba uma string como parâmetro e desloque os seus caracteres uma posição para a direita. Por exemplo, a string "casa" seria alterada para "acas". Repare que o último caractere vai para o ínicio da string.

`void roda_string (char* str);`

```
void roda_string (char* str)
{
  int i=1;

  while (str[i] != 0) {
    str[i] = str[i-1];
    i++;
  }

  str[0] = str[i-1];
}
```

9. Implemente uma função que receba duas strings como parâmetros e procure a primeira ocorrência da segunda string na primeira, retornando a posição da ocorrência, ou -1 caso não haja nenhuma ocorrência. Por exemplo: str1 = "banana", str2 = "ana" retorna 1. str1 = "Estruturas", str2 = "tura" retorna 5. str1 = "banana", str2 = "ite" retorna -1. Considere que caracteres maiúsculos e minúsculos são diferentes.

`int procura_string (char* str1, char* str2);`

```
int procura_string (char* str1, char* str2)
{
  int i=0, j;

  while (str1[i] != 0) {

    /* acha primeira letra */
    while (str1[i] != 0  &&  str1[i] != str2[0]) {
      i++;
    }

    /* acha proximas letras */
    j = 1;
    while (str1[i+j] != 0  &&  str2[j] != 0  &&  str1[i+j] == str2[j]) {
      j++;
    }

    if (str2[j] == 0) return i;

    i++;
  }

  return -1;
}
```

10. Escreva uma função que identifique se uma dada string A é ou não um palíndromo (ex: "SOCORRAM ME EM MARROCOS" é um palíndromo).

`void Palindromo(char string[], int n);`

```
void Palindromo(char str[], int n)
{
  int palind, i, ultimo, meio;

  /* indice do ultimo caractere em str */
  ultimo = strlen(str) - 1;

  /* indice do caractere do meio em str */
  meio = strlen(str) / 2;

  /* assumo que str eh palindromo */
  palind = 1;

  /* verifico se realmente eh palindromo */
  i = 0;
  while (palind && i <= meio) {
    palind = (str[i] == str[ultimo-i]);
    i++;
  }

  return palind;
}
```

11. Escreva uma função que concatene duas strings dadas A e B numa nova string C (ex: se A = “Perdi as chaves ” e B = “do carro”, então C = “Perdi as chaves do carro”):

`void ConcatenaString(char stringA[], char stringB[], char stringC[]);`

```
void ConcatenaString(char strA[], char strB[], char strC[])
{
  int i, j;

  i = 0;
  while (strA[i] != 0) {
    strC[i] = strA[i];
    i++;
  }

  j = 0;
  while (strB[j] != 0) {
    strC[i+j] = strB[j];
    j++;
  }
}
```

12. Escreva uma função que conte quantas letras maiúsculas existem numa string:

`void ContaMaiusculas(char string[]);`

```
void ContaMaiusculas(char str[])
{
  int i=0, conta=0;

  while (str[i] != 0) {
    if (str[i] >= 'A' && str[i] <= 'Z') {
      conta++;
    }
    i++;
  }

  return conta;
}
```