

# Contagem de palindromos

**Requested files:** user.c (Download)

**Type of work:** Individual work

**Grade settings:** Maximum grade: 10

**Dissable external file upload, paste and drop external content:** Yes

**Run:** Yes **Evaluate:** Yes

**Automatic grade:** Yes

## Problema:

Um palíndromo é uma palavra que pode ser lida tanto da direita para a esquerda como da esquerda para a direita. Escreva um programa totalmente recursivo que calcule a quantidade de palavras palíndromo que existem entre duas palavras de tamanho " $n$ " escritas em minúscula sem acentuação nem caracteres especiais. Por exemplo, se  $n=3$  há 5 palavras palíndromo entre as palavras "aba" e "afa", as palavras palíndromo para o intervalo são:

"aba", "aca", "ada", "aea" e "afa"

A ordem das palavras são definidas de acordo com seu código ASCII (<http://ascii.cl/>). A seguinte palavra de uma palavra " $x$ " é aquela que difere em seu último caractere em uma unidade a mais quando convertida em um vetor de decimais. Quando seu último valor é "z" o seguinte valor no vetor será incrementado em uma unidade e o valor "z" é convertido em "a". Assim, o seguinte exemplo mostra como é gerada a sequência de palavras para o intervalo "aby" e "aca":

- "aby" em código ASCII como um vetor de decimais é {97,98,121}
- "abz" em código ASCII como um vetor de decimais é {97,98,122}
- "aca" em código ASCII como um vetor de decimais é {97,99,97}

## Entrada e Saída:

A entrada será constituída por duas linhas, a primeira linha com o tamanho " $n$ " das palavras e a segunda linha com as duas palavras " $x$ " e " $y$ " separadas por um espaço em branco. Como saída você deve imprimir na tela do computador a quantidade de palavras palíndromo entre o intervalo fechado da palavra " $x$ " e " $y$ ".

Exemplos de Entrada	Exemplos de Saída
3 aba afa	5
2 ei hl	3
4 msqe nuqe	28

## Dicas:

- Para comparar se duas palavras " $x$ " e " $y$ " são iguais use a função "strcmp":

```
if (strcmp(x,y) == 0) {  
    // as duas palavras são iguais  
}
```

- A seguinte função recursiva gera a próxima palavra da palavra " $x$ " de tamanho " $n$ " atualizando os valores do vetor de caracteres " $x$ "

```
void next_word(char x[], int n) {  
    if (n > 0) {  
        if (x[n-1] == 'z') {  
            x[n-1] = 'a';  
            next_word(x, n-1);  
        } else {  
            x[n-1] = x[n-1]+1;  
        }  
    }  
}
```

## Requested files

### user.c

```
1 #include<stdio.h>  
2 #include<string.h>  
3  
4 int main() {  
5     int n;  
6     scanf("%d", &n);  
7     char x[n];  
8     char y[n];  
9     scanf("%s %s", x, y); // não use outras funciones para ler as palavras  
10    // escreva seu código aqui  
11    return 0;  
12 }  
13
```

## Execution files

### vpl\_run.sh