

Guião da Etapa 1

Laboratórios de Algoritmia I
Laboratórios de Informática II

2013/2014

Tarefa 1

Pretende-se que implemente funções de manipulação de strings utilizando código desenvolvido exclusivamente por si. Entre essas funções, deverão existir as seguintes:

1. Calcular o tamanho de uma string s
2. Procurar um caractere c numa string s e devolver o primeiro índice onde ele se encontra
3. Contar o número de ocorrências de um caractere c numa string s
4. Devolver o primeiro índice em que uma string s_1 se encontra dentro de outra string s_2
5. Verificar se uma string s_1 é prefixo de outra string s_2
6. Verificar se uma string s_1 é sufixo de outra string s_2
7. Verificar se duas strings s_1 e s_2 são iguais
8. Verificar se uma string s_1 é menor ¹ do que uma string s_2
9. Devolver -1 , 0 ou 1 se a string s_1 for menor, igual ou maior do que a string s_2 (segundo a ordem lexicográfica)
10. Procura o primeiro índice em s_1 que contém um dos caracteres na string s_2
11. Procura o primeiro índice em s_1 que não contém um dos caracteres na string s_2
12. Copiar para s_1 a string s_2
13. Colar s_2 ao fim da string s_1
14. Mover n bytes da área m_2 para a área m_1 (esta função deverá funcionar mesmo que m_1 e m_2 partilhem memória)

¹encontra-se primeiro no dicionário ou dito de outra maneira, é menor segundo a ordem lexicográfica

Nos casos de funções que devolvam um índice, estas deverão devolver o valor -1 caso não encontrem o que procuram.

Para testar o seu código sugere-se que faça um programa para cada função que leia os argumentos e imprima o resultado. Mas tenha o cuidado de colocar todas as funções implementadas num módulo (num ficheiro chamado `cadeias.c` e que estas não imprimam nem leiam nada.

O que queremos dizer com isto é que as funções que vai implementar devem ter os seguintes protótipos (que ficariam no ficheiro `cadeias.h`):

```
int cad_tamanho(char s[]);
int cad_procurar_car(char c, char s[]);
int cad_ocorrencias(char c, char s[]);
int cad_procurar_str(char s1[], char s2[]);
int cad_e_prefixo(char s1[], char s2[]);
int cad_e_sufixo(char s1[], char s2[]);
int cad_iguais(char s1[], char s2[]);
int cad_primeiro(char s1[], char s2[]);
int cad_comparar(char s1[], char s2[]);
int cad_um_dos(char s1[], char s2[]);
int cad_nenhum_dos(char s1[], char s2[]);
int cad_copiar(char s1[], char s2[]);
int cad_colar(char s1[], char s2[]);
int cad_mover((char m1[], char m2[], int n);
```

E, para testar cada função, escrever-se-ia um programa do género (por exemplo no ficheiro `tproc.c`):

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

int main() {
    char s1[1000];
    char s2[1000];
    scanf("%s", s1);
    scanf("%s", s2);
    printf("%d\n", cad_procurar_str(s1, s2));
    return 0;
}
```

E compiláramos o programa escrevendo por exemplo:

```
gcc -o tproc tproc.c cadeias.c
```

Tarefa 2

Para avaliar a tarefa 2, sugere-se que faça um programa que leia dois inteiros (o n° de linhas nl e o n° de colunas nc da sopa de letras) e seguidamente leia as nl linhas da sopa de letras. Após isso o seu programa deve ler o n° de coordenadas n e seguidamente os n pares de coordenadas sabendo que primeiro se lê o n° da coluna c e depois o n° da linha l .

Segue-se um exemplo:

```
3 4
ABCD
```

EFGH

IJKL

5

1 2 1 3 2 2 3 3 2 4

Devendo neste caso o seu programa imprimir a string BCFKH.