

MC102 - Algoritmos e Programação de Computador

Prof. Alexandre Xavier Falcão

14º Aula: Cadeias de Caracteres (cont.)

1 Convertendo cadeias em números, e vice-versa (cont.)

Em algumas situações desejamos que o programa leia uma seqüência de n valores da entrada padrão. Esses valores podem ser passados, um por linha ou separados por espaços em branco (ou vírgulas).

```
#include <stdio.h>

#define N 50

int main()
{
    float v[N];
    int i,n;

    /* Opção 1 */

    fscanf(stdin,"%d",&n);
    for (i=0; i < n; i++)
        fscanf(stdin,"%f",&v[i]);

    for (i=0; i < n; i++)
        fprintf(stdout,"%f\n",v[i]);

    return 0;
}
```

Porém, em alguns casos, vamos ver que esses valores são passados em uma cadeia de caracteres. Nesses casos, o comando **strchr** pode ser usado para procurar a próxima posição da cadeia com um dado caracter (e.g. espaço em branco ou vírgula).

```
#include <stdio.h>

#define N 50

int main()
{
```

```

char  valores[200],*vaux;
float v[N];
int  i,n;

fgets(valores,199,stdin);
vaux = valores;
sscanf(vaux,"%d",&n);
vaux = strchr(vaux,' ') + 1;
for (i=0; i < n; i++) {
    sscanf(vaux,"%f",&v[i]);
    vaux = strchr(vaux,' ') + 1;
}

for (i=0; i < n; i++)
    fprintf(stdout,"%f\n",v[i]);
return 0;
}

```

2 Manipulando cadeias de caracteres

Cadeias de caracteres podem ser manipuladas para diversos fins práticos. O programa abaixo, por exemplo, ilustra algumas operações úteis.

```

#include <stdio.h>
#include <string.h>

int main()
{
    char s1[11], s2[11], s3[21];
    int resultado;

    fscanf(stdin,"%s",s1);
    fscanf(stdin,"%s",s2);

    /* comprimento das cadeias */

    printf("%s tem %d caracteres\n",s1,strlen(s1));
    printf("%s tem %d caracteres\n",s2,strlen(s2));

    resultado = strcmp(s1,s2); /* compara strings */

    if (resultado == 0)
        printf("%s = %s\n",s1,s2);
    else
        if (resultado < 0){

```

```

    printf("%s < %s\n",s1,s2);
    strcat(s3,s1); /* concatena strings */
    strcat(s3,s2);
    printf("%s\n",s3);
}else{
    printf("%s > %s\n",s1,s2);
    strcat(s3,s2);
    strcat(s3,s1);
    printf("%s\n",s3);
}

return 0;
}

```

Veja também as funções **strcpy** e **strncpy** para realizar cópias de caracteres e **strncat** para concatenação de n caracteres no final de uma cadeia.

Um exemplo prático é a busca de padrões em uma cadeia de caracteres. Este problema é muito comum quando pretendemos localizar palavras em um texto e padrões em uma seqüência de caracteres, como ocorre em Bioinformática. O algoritmo abaixo ilustra uma solução $O(nm)$, onde n é o comprimento da cadeia e m é o comprimento do padrão.

```

#include <stdio.h>
#include <string.h>

int main()
{
    char cadeia[101], padrao[11];
    int i,n,m;

    fscanf(stdin,"%s",padrao);
    fscanf(stdin,"%s",cadeia);

    n = strlen(cadeia);
    m = strlen(padrao);

    i = 0;
    while (i <= (n-m)){
        /* compara m caracteres */
        if (strncmp(&cadeia[i],padrao,m)==0)
            printf("Padrão encontrado na posição %d\n",i);
        i++;
    }

return 0;
}

```

3 Exercício

Quando o padrão não é encontrado a partir da posição i , o algoritmo acima repete a busca a partir da posição $i + 1$. Porém, quando os k primeiros caracteres são iguais aos do padrão e a diferença ocorre no caracter da posição $i + k$, a nova busca pode iniciar nesta posição. Note também que toda vez que o padrão for encontrado na posição i , a próxima busca pode iniciar na posição $i + m$. Com essas observações, escreva um algoritmo mais eficiente para buscar padrões.