

Linguagem C - **Soluções****I Arrays de caracteres (strings)**

1. Faça o seguinte programa que pede o primeiro nome, o ultimo nome e a idade.

- (a) Juntar os dois nomes numa string separada por um traço.
- (b) Se o resultado for "ze - maria" deve indicar a mensagem "Ok"
- (c) Verifique que a alínea 1 pode ser resolvida com a função `sprintf`

```
#include <stdio.h>
#include <string.h>
main () {
    char pnome[80], unome[80], nome[80], tmp[100];
    int idade;

    printf("Primeiro nome:");
    //scanf("%s", pnome);
    fgets(pnome, 80, stdin);
    pnome[ strlen(pnome)-1 ] = '\0';

    printf("Ultimo nome:");
    fgets(unome, 80, stdin);
    unome[ strlen(unome)-1 ] = '\0';

    printf("Idade:");
    //scanf("%d", &idade);
    fgets(tmp, 80, stdin);
    idade = atoi ( tmp);

    // produzir a variavel nome
    //strcpy(nome, pnome);
    //strcat(nome, " - ");
    //strcat(nome, unome);
    sprintf(nome, "%s - %s", pnome, unome);

    printf("Idade: %d e nome completo: %s\n", idade, nome);

    if ( strcmp(nome, "ze - maria" ) == 0 )
        printf("OK\n");
}
```

2. Verifique que o programa seguinte escreve os argumentos recebidos.

```
#include <stdio.h>
int main( int argc, char *argv[] ) {
    int i;
    for ( i=0; i<argc; i++ )
        printf("%d-> %s\n", i, argv[i] );
}
```

- (a) altere-o de forma a também indicar o número de argumentos recebidos e o tamanho de cada um deles (em caracteres).

3. Ler uma string, contar os E's e transformar as minúsculas em maiúsculas

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

void maiusculas( char s[] );

main() {
    printf ("Linha: ");
    char s[80];
    fgets ( s, 80, stdin );

    int i, contar = 0;
    for ( i=0; i < strlen(s) ;i++ ) {
        if ( s[i] == 'E' || s[i] == 'e') contar++;
    }
    printf ("Existem %d\n", contar );
    maiusculas( s );
    printf ("Final: %s\n", s );
}

void maiusculas ( char s[] ) {
    int i;
    for ( i=0; i<strlen(s);i++ ) {

        if ( s[i] >= 'a' && s[i] <='z' )
            s[i] = s[i] - 'a' + 'A';
    }
}
```

4. Agora faça:

- (a) Escrever o alfabeto no ecrã
(b) (opcional) ler nome, escrever só as iniciais maiusculas
(c) (opcional) ler uma linha de texto e contar as palavras

```
#include <stdio.h>
#include <string.h>
main() {
    int i;
    char s[80];
    for ( i=0;i<40;i++) {
        printf ("%c", 'A'+ i);
    }

    // experimentar funções

    printf ("Diga: ");
    fgets ( s, 40, stdin );
    printf ("Disse %s\n", s );

    char c, p[40];
    int n;
    printf ("Diga outras coisas: ");
    scanf ("%c", &c);
    scanf ("%d", &n);
    scanf ("%s", p);
    fgets ( s, 80, stdin );

    printf ("c=<%c>, n=<%d>, p=<%s>, s=<%s>\n", c, n, p, s );
}
```

```

//experimental as funções de manipulação de strings

printf ("Diga ainda outras coisas: ");
fgets ( s, 80, stdin );
s [ strlen(s) - 1 ] = 0;    //limpar o enter

int tamanho = strlen(s);
strcpy ( p, s );
strcat ( p, "+" );
strcat ( p, s );

int r = strcmp ( s, "Jac" );

printf ("tamanho=%d, p=<%s>, r=%d\n", tamanho, p, r );
}

```

5. (opcional) Escreva a seguinte função que permite extrair o campo *n* de uma dada *frase*, considerando um determinado carater como *separador*.

```
void obter_campo(char frase[], char separador, int indice, char resultado[])
```

- (a) Teste a função com a frase seguinte para obter os valores “17 anos” e “ISCTE-IUL”

Mariana Silva/17 anos/Lisboa/ISCTE-IUL

II Passagem por valor

1. Considere a seguinte função que tem como objectivo trocar o valor de duas variáveis:

```

void swap (int x, int y) {
    int temp;
    temp = x;
    x = y;
    y = temp;
}

```

- (a) Faça um pequeno programa em que teste a função swap.
(b) O resultado obtido foi o esperado? Em caso negativo, corrija a função.

```

void swap (int *x, int *y) {
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

main () {
    int a=5, b=7;
    swap(a, b);
    printf("A=%d, b=%b\n", a, b);
}

```

III Ficheiros

1. Teste o seguinte programa que permite escrever num ficheiro

```

#include <stdio.h>
main() {
    FILE *fic;
    fic = fopen("batata.txt", "w");
    fprintf(fic, "Eu sou o %d\n", getpid() );
    fprintf(fic, "Adeus\n");
}

```

```

        fclose(fic);
    }

```

2. Teste o seguinte programa que permite ler todas as linha de um ficheiro.

```

#include <stdio.h>
main() {
    char linha[100];
    FILE *fic;
    fic = fopen("batata.txt", "r");
    int i = 0;
    while ( fgets(linha, 100, fic ) != NULL )
        printf("Linha %d: %s", i++, linha);
    fclose(fic);
}

```

3. Pretende-se criar um editor para edição de ficheiros de texto. Por agora, suponha que o nome do ficheiro de texto é fixo: teste.txt. Faça um programa que apresente um menu com as opções: mostrar, acrescentar, apagar, duplicar e sair. A opção “mostrar” deverá mostrar o conteúdo do ficheiro. A opção acrescentar pede uma nova linha e acrescenta-a ao ficheiro. A opção apagar apaga uma linha, com base no número de linha introduzido pelo utilizador.

```

#include <stdio.h>
#include <stdlib.h>

main() {
    int opcao;
    char s[0];

    while ( 1 ) {
        printf ( "1. Mostrar\n" );
        printf ( "2. Acrescentar\n" );
        printf ( "3. Apagar\n" );
        printf ( "4. Duplicar\n" );
        printf ( "0. Sair\n" );

        printf ( ">>>> Opção: " );
        fgets ( s, 80, stdin );
        opcao = atoi(s);

        if ( opcao == 0 ) exit(0);

        if ( opcao == 1 ) mostrar();
        if ( opcao == 2 ) acrescentar();
        if ( opcao == 3 ) apagar();
        if ( opcao == 4 ) duplicar();
    }
}

int duplicar() {
    char s[80];
    printf ( "Nº da linha a duplicar: " );
    fgets ( s, 80, stdin );
    int n = atoi(s);

    int i;
    FILE *fpr, *fpw;

    // 1ª passo copiar para um aux todas as linhas,
    // acrescentando a nova

    fpr = fopen ( "teste.txt", "r" );
    fpw = fopen ( "aux.txt", "w" );

```

```

    for ( i=1; fgets( s, 80, fpr ); i++ ) {
        fprintf ( fpw, "%s", s );
        if ( i == n ) fprintf ( fpw, "%s", s );
    }
    fclose(fpr);
    fclose(fpw);

    // ! aux.txt já tem a linha nova
    // copiar aux -> teste.txt

    fpr = fopen ( "aux.txt", "r" );
    fpw = fopen ( "teste.txt", "w" );
    for ( i=1; fgets( s, 80, fpr ); i++ )
        fprintf ( fpw, "%s", s );
    fclose(fpr);
    fclose(fpw);
}

int apagar() {
    char s[80];
    printf ("Nº da linha: ");
    fgets ( s, 80, stdin);
    int n = atoi(s);

    int i;
    FILE *fpr, *fpw;

    // 1ª passo copiar para um aux todas as linhas menos a que
    // queremos apagar

    fpr = fopen ( "teste.txt", "r" );
    fpw = fopen ( "aux.txt", "w" );
    for ( i=1; fgets( s, 80, fpr ); i++ )
        if ( i != n ) fprintf ( fpw, "%s", s );
    fclose(fpr);
    fclose(fpw);

    // ! aux.txt já tem a linha apagada
    // copiar aux -> teste.txt

    fpr = fopen ( "aux.txt", "r" );
    fpw = fopen ( "teste.txt", "w" );
    for ( i=1; fgets( s, 80, fpr ); i++ )
        fprintf ( fpw, "%s", s );
    fclose(fpr);
    fclose(fpw);
}

int acrescentar () {
    char s[80];
    printf ("Nova linha: ");
    fgets ( s, 80, stdin);

    FILE *fp = fopen ("teste.txt", "a");
    if ( fp == NULL ) printf ("Erro\n" );
    fprintf ( fp, "%s", s );
    fclose(fp);
}

int mostrar ( ) {
    int i;
    char s[80];
    FILE *fp;
    printf ("-----\n");

```

```

        fp = fopen ("teste.txt", "r" );
        for ( i=1; fgets ( s, 80, fp) != NULL; i++ )
            printf ( "%d: %s", i, s );
        fclose(fp);
        printf ("----- FIM -----\\n");
    }

```

4. Melhore o exercício anterior de forma a:

- dar o nome do ficheiro no inicio (deixar de ser um nome de ficheiro fixo)
- acrescentar operações (ex: substituir o conteúdo de uma linha)
- acrescentar possibilidades (ex: mostrar o ficheiro pagina a página, procurar texto, etc)