

NOTA: estão são alguns dos exemplos apresentados na aula teórica
os comentários ao código são reduzidos; foram feitos na aula

//main1.c - DECLARACAO DE ARRAYS + LEITURA DE STRINGS

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

#define NOME_MAX_COMP 50 //evitar "magic numbers"
#define NUM_MESES 12

int main(int argc, char *argv[])
{
    char nome[NOME_MAX_COMP];
    int peso[NUM_MESES];
    int i;

    printf ("Nome ? ");
    scanf ("%s", nome); //tentar c/ "Rui Santos"

    for (i=0; i<NUM_MESES; i++)
    {
        printf ("peso[%d] ? ", i); scanf ("%d", &peso[i]);
    }
    for (i=0; i<NUM_MESES; i++)
    {
        printf ("peso[%d] = %d\n", i, peso[i]);
    }

    // CÓDIGO A COMPLETAR C/ PROCESSAM. DOS DADOS ...

    return 0;
}
```

```

//main2.c - DECLARACAO DE ARRAYS + LEITURAS DO TECLADO (CUIDADOS)

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

#define NOME_MAX_COMP 10 //NOTAR
#define NUM_MESES 4

int main(int argc, char *argv[])
{
    char nome[NOME_MAX_COMP];
    int peso[NUM_MESES];
    int i;

    printf ("Nome ? ");
    fgets(nome,sizeof(nome),stdin); // por que não gets() ?
    // EXPERIMENTAR CORRER COM NOME1 =
    // 1) 123
    // 2) 1234567890
    // 3) 123456789
    // E INTERPRETAR OS RESULTADOS
    // (VERIFICAR QUE EM ALGUNS CASOS 'NOME' JA´ INCLUI O NEWLINE)

    /* TESTAR S/ e C/ ESTE CÓDIGO
    if ((strlen(nome)==(sizeof(nome)-1)) && nome[strlen(nome)-1]!='\n')
        while(getchar()!='\n');
    */

    for (i=0; i<NUM_MESES; i++)
    {
        printf("peso[%d] ? ",i); scanf("%d",&peso[i]);
        // perror("main"); // NAO DA' ERRO; testar valor de retorno de
scanf()
    }

    // mostrar valores lidos
    printf("\n\n%s\n",nome);
    for (i=0; i<NUM_MESES; i++)
    {
        printf("peso[%d] = %d\n",i,peso[i]);
    }

    return 0;
}

```

```

//main3.c - STRINGS DE C

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

#define MAX_LEN 10

int main(int argc, char *argv[])
{
    // Declaração de strings
    char nome1[MAX_LEN] = "Ana Sousa";
    char nome2[MAX_LEN];
    char *nome3; // ATRIBUIDO A CONSTANTE (V. ABAIXO)
    char *nome4;
    char *nome5=NULL;

    printf("%s\n",nome1);
    // nome2 = "Rui Santos"; // TESTAR

    nome3="Pedro Silva";
    printf("%s\n",nome3);

    nome4=(char *) malloc(MAX_LEN*sizeof(char));
    printf("Nome4 ? "); fgets(nome4,MAX_LEN,stdin);
    printf("\nnome4 = %s\n",nome4);

    // QUAL A DIFERENCA ENTRE AS 3 SEQ.S DE INSTRUCOES SEGUINTE ?

    //1) -
    //nome5=nome4;

    //2) -
    // strcpy(nome5,nome4); // sintaxe: strcpy(destination,source)

    //3) - CORRIGE O ERRO DE 2)
    //nome5=(char *) malloc(MAX_LEN*sizeof(char));
    //strcpy(nome5,nome4); //strcpy(destination,source)

    //4)
    nome5 = &nome4[1];

    printf("nome5 = %s\n",nome5); // imprime "SOUSA"

    return 0;
}

```

```
//main4.c -    // RELAÇÃO ENTRE ARRAYS E APONTADORES
```

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

```
#define MAX_LEN 10
```

```
int main(int argc, char *argv[])
{
```

```
    int i;
    int a[MAX_LEN];
    int *b;
    int c[MAX_LEN];
```

```
    for (i=0; i<MAX_LEN; i++)
        a[i]=i;
```

```
    b = a;
    // c = a;  //TESTAR (o que acontece ?)
```

```
    for (i=0; i<MAX_LEN; i++)
        printf ("a[%d]=%d; b[%d]=%d\n",i, a[i], i, b[i]);
```

```
    printf("\n");
```

```
    for (i=0; i<MAX_LEN; i++)
        printf ("b[%d]=%d\n",i,*(b+i));
```

```
    return 0;
```

```
}
```

```
//main5a.c - // ARRAYS DE STRINGS
```

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

```
#define MAX_LEN 10
```

```
void mostra(int n, char nms[][MAX_LEN])
{
    int i;
    for (i=0; i<n; i++)
        printf("%s\n", nms[i]);
}
```

```
int main(int argc, char *argv[])
{
    char nomes[3][MAX_LEN];

    strcpy(nomes[0], "Ana Sousa");
    strcpy(nomes[1], "Rui Silva");
    mostra(2, nomes);

    return 0;
}
```

```
-----
//main5b.c - // ARRAYS DE STRINGS
```

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

```
#define MAX_LEN 10
```

```
// POR QUE DA' ERRO DE COMPILACAO ?
```

```
void mostra(int n, char *nms[])
{
    int i;
    for (i=0; i<n; i++)
        printf("%s\n", nms[i]);
}
```

```
int main(int argc, char *argv[])
{
    char nomes[3][MAX_LEN];

    strcpy(nomes[0], "Ana Sousa");
    strcpy(nomes[1], "Rui Silva");
    mostra(2, nomes);

    system("PAUSE");
    return 0;
}
```

```
-----
```

```

//main6a.c - ARRAYS DE STRINGS

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

#define MAX_LEN 10
#define MAX_NOMES 3 // >= 2

void mostra(int n, char *nms[])
// ALTERNATIVA: void mostra(int n, char **nms)
{
    int i;
    for (i=0; i<n; i++)
        printf("%s\n", nms[i]);
}

int main(int argc, char *argv[])
{
    char *nomes[MAX_NOMES];

    int i;

    // reserva dinamicamente a memoria
    for (i=0; i<MAX_NOMES; i++)
        nomes[i]=(char *) malloc(MAX_LEN*sizeof(char));

    // atualiza array de nomes
    strcpy(nomes[0], "Ana Sousa");
    strcpy(nomes[1], "Rui Silva");
    mostra(2, nomes);

    // liberta a memoria reservada dinamicamente
    for (i=0; i<MAX_NOMES; i++)
        free(nomes[i]);

    // mostra(2, nomes); // CUIDADO!, a memória já foi libertada

    return 0;
}

```

```

//main6b.c - ARRAYS DE STRINGS

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

#define MAX_LEN 10

void lerNomes(int n, char **nms)
{
    int i;

    for (i = 0; i < n; i++)
    {
        printf("Nome [%d] ? ", i);
        fgets(nms[i],MAX_LEN,stdin);
    }
}

void mostrarNomes(int n,char **nms)
{
    int i;
    for (i=0;i<n;i++)
        printf("%s",nms[i]);
}

int main(int argc, char *argv[])
{
    char **nomes;

    int i, n;

    printf("Quantos nomes ? ");
    scanf("%d",&n);
    while (getchar() != '\n'); //fflush(stdin);

    // reserva dinamicamente a memoria
    nomes = (char **) malloc(n*sizeof(char *));
    for (i = 0; i < n; i++)
        nomes[i] = (char *) malloc(MAX_LEN*sizeof(char));

    // le^ os nomes
    lerNomes(n,nomes);

    // mostra os nomes
    mostrarNomes(n,nomes);

    // liberta a memoria reservada dinamicamente
    for (i = 0; i < n; i++)
        free(nomes[i]);
    free(nomes);

    return 0;
}

```

```
//main7.c - FUNCOES E APONTADORES
```

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

```
#define MAX_LEN 10
```

```
void teste1(void)
{
    printf("Hello\n");
}
```

```
void teste2(int n)
{
    int i;
    for (i=0; i<n; i++)
        printf("Hello no. %d\n",i);

    printf("Hello\n");
}
```

```
int main(int argc, char *argv[])
{
    void (*func1) (void);
    void (*func2) (int);

    func1 = teste1;
    func1();

    func2 = teste2;
    func2(3);

    return 0;
}
```