Strings

Adaptado de Deise Saccol

Strings

- Não existe um tipo String em C.
- Strings em C são uma array do tipo char que termina com '\0'.
- Para literais String, o próprio compilador coloca '\0'.

```
#include <stdio.h>
#include <stdlib.h>
main(){
    char re[8] = "lagarto";
    printf ("%s", re);

}
```

Para ler uma String

- Comando gets

```
#include <stdio.h>
#include <stdlib.h>
main(){
    char re [80];
    printf ("Digite o seu nome: ");
    scanf("%s", &re);
    printf ("Oi %s\n", re);
}
```

Para ler uma String

- Comando gets

```
#include <stdio.h>
#include <stdlib.h>
main(){
    char re [80];
    printf ("Digite o seu nome: ");
    gets(re);
    printf ("Oi %s\n", re);
}
```

Para comparar duas strings

- strcmp (s1, s2); strcmp retorna 0 se as duas strings são iguais.
- Precisa do #include <string.h>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main( ){
    char re[80];
    printf ("Digite a senha: ");
    scanf ("%s", &re);
    if (strcmp(re,"laranja")==0) {
        printf ("Senha correta\n");
    }
    else {
        printf ("Senha invalida\n");
    }
}
```

Pode utilizar scanf no lugar do gets, só que o scanf não lê espaços em branco.
Se o usuário digitar:
>“Alexandre Costa e Silva”
o scanf vai pegar apenas “Alexandre”.
O gets pega tudo.

Para saber o tamanho de uma string

- `int size = strlen(str);`
 - Retorna um valor inteiro com o número de caracteres da String.
 - Precisa do `#include <string.h>`

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

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

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

```
main(){
```

```
    char re[80];
```

```
    printf ("Digite a palavra: ");
```

```
    scanf ("%s", &re);
```

```
    int size=strlen(re);
```

```
    printf ("Esta palavra tem %d caracteres.\n", size);
```

```
}
```

Para copiar o conteúdo de uma string para outra

- strcpy(para, de);
- Precisa do #include <string.h>

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main()
{
    char str[80];
    strcpy (str, "Alo");
    printf ("%s", str);

}
```

Exemplo do uso das funções

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char palavra[30], palavra2[30], teste[30];
    int tam, comp;
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    printf ("A palavra digitada eh %s\n", palavra2);
    strcpy (teste,"aula teste");
    printf ("A palavra copiada eh %s\n", teste);
    strcat (teste, " 1");
    printf ("A palavra concatenada eh %s\n", teste);
    tam=strlen(palavra);
    printf ("\nO tamanho da primeira string eh %d", tam);
    comp=strcmp (palavra, palavra2);
    if (comp==0)
        printf ("\nSao iguais: %d", comp);
        else
            printf ("\nSao diferentes: %d", comp);
    printf ("\nMaiusculo: %s", strupr(palavra));
    printf ("\nMinusculo: %s\n", strlwr(palavra));
}
```

Exemplo

1. Escreva um programa que leia duas strings e as coloque na tela. Imprima também a segunda letra de cada string.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char palavra[30];
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c", palavra[1]);
    printf ("\n Informe uma segunda string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\nA segunda letra da string digitada eh %c\n", palavra[1]);
}
```

Exercício

2. Escreva um programa que leia uma string, conte quantos caracteres desta string são iguais a 'a' e substitua os que forem iguais a 'a' por 'b'. O programa deve imprimir o número de caracteres modificados e a string modificada.

Exemplo

2. Escreva um programa que leia uma string, conte quantos caracteres desta string são iguais a 'a' e substitua os que forem iguais a 'a' por 'b'. O programa deve imprimir o número de caracteres modificados e a string modificada.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
main(){
    char string[30];
    int tamanho, i, contaA=0;
    printf ("\nInforme uma string: ");
    gets (string);
    tamanho=strlen(string);
    for (i=0;i<tamanho;i++)    {
        if (string[i]=='a')    {
            contaA++;
            string[i]='b';
        }
    }
    printf ("\nO numero de caracteres modificados eh %d", contaA);
    printf ("\nA string modificada eh %s\n", string);
}
```

Exercícios

4. Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabela abaixo. Faça um programa que leia o salário e o cargo de um funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá então receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

•Cargo	Percentual
•Gerente	10%
•Engenheiro	20%
•Técnico	30%

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
main(){
    char cargo[20];
    float salario, novoSalario;
    printf ("\nDigite o cargo: ");
    scanf ("%s", cargo);
    printf ("\nDigite o salario: ");
    scanf ("%f", &salario);
    if (strcmp(cargo, "gerente")==0)
        novoSalario=salario*1.1;
    else if (strcmp(cargo, "engenheiro")==0)
        novoSalario=salario*1.2;
    else if (strcmp(cargo, "tecnico")==0)
        novoSalario=salario*1.3;
    else
        novoSalario=salario*1.4;
    printf ("\nO salario antigo eh %.2f, o salario novo eh %.2f e a diferenca eh
de R$ %.2f\n", salario, novoSalario, novoSalario-salario);
}
```

Matriz

Matriz

A sintaxe em C para a criação de um vetor multi dimensional ou matriz é:

tipo identificador[n. de variáveis 1][n. de variáveis 2];

Onde

- *tipo* é o tipo das variáveis que devem ser criados. Exemplo: *int, char, float*, entre outros;
- *identificador* é o nome que será utilizado para referenciar o conjunto de variáveis;
- *n. de variáveis 2* é o número de variáveis de cada vetor que será criado.
- *n. de variáveis 1* é o número de vetores que será criado.

Matriz

Um exemplo de declaração de matriz em C:

```
int notas[3][8];
```

	0	1	2	3	4	5	6	7
0								
1								
2								

Matriz

Após declarar uma matriz precisamos de um modo de referenciar seus elementos individualmente. Isto é feito através do número entre colchetes para cada dimensão seguindo o nome do vetor.

Note a diferença nas linhas que se segue.

```
int notas[50][4];
notas[5][3] = 85;
```

Vale lembrar, assim como em vetores, os índices de uma matriz variam entre 0 e $n - 1$. Além disso, assim como em vetores, podemos referenciar posições da matriz utilizando variáveis inteiras. O exemplo que segue tem o mesmo resultado que o exemplo anterior.

```
int notas[50][4];
int i = 5;
int j = 3;
notas[i][j] = 85;
```

Matriz - Armazenando

Em seguida temos um exemplo de como fazer a leitura das notas de cada prova para cada aluno.

```
for(i=0; i<50; i++)
  for(j=0; i<4; j++){
    printf("Digite a nota da prova %d do aluno %d: ",j,i);
    scanf("%d",&notas[i][j]);
  }
```

Matriz - Lendo

Em seguida temos um exemplo de como é feita a leitura dos dados já armazenados em uma matriz.

```
soma_turma = 0;
for(i=0; i<50; i++){
    soma_aluno = 0;
    for(j=0; j<4; j++)
        soma_aluno = soma_aluno + notas[i][j];

    soma_turma = soma_turma + soma_aluno/4;
    printf("Media do aluno %d é %f",i,soma_aluno/5);
}
printf("Media da turma %f.",soma_turma/50);
```

Matriz - Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#define TAM 4

int main(){
int mat[TAM][TAM];
int i=0,j=0;
for (i=0; i<TAM; i++) {
for (j=0; j<TAM; j++) {
printf("Digite o valor da posicao (%d, %d): ",i,j);
scanf("%d",&mat[i][j]);
}
}
for (i=0; i<TAM; i++){
for (j=0; j<TAM; j++)
printf("%d ",mat[i][j]);
printf("\n");
}
}
```

Exercício

- Faça um programa que leia uma *string* digitada pelo usuário (máximo 20 caracteres) e imprima na tela o número de vogais existente nesta *string*.
- Faça um programa que dado 2 palavras, determine:
 - Se as palavras são iguais;
 - Caso as palavras sejam diferentes, qual delas tem maior comprimento;