Lendo uma string do teclado

- Usando a função scanf()
  - Existem várias maneiras de se fazer a leitura de uma sequência de caracteres do teclado. Uma delas é utilizando a já conhecida função scanf() com o formato de dados “%s”:
  - Ex:

```
char str[20];
scanf(“%s”,str);
```

Quando usamos a função **scanf()** para ler uma string, o símbolo de & antes do nome da variável não é utilizado. Os colchetes também não são utilizados, pois queremos ler a string toda e não apenas uma letra.

A função scanf() padrão (com uso da string de formatação "%s") lê apenas strings digitadas sem espaços em branco, ou seja, somente uma palavra.

Lendo uma string do teclado

- Usando a função scanf()
  - Para se fazer a leitura de uma sequência de caracteres (string) do teclado, incluindo os espaços em branco, com a função scanf(), pode-se utilizar a seguinte string de formatação de dados: “%[^\n]s”, que fará a leitura até que uma quebra de linha seja encontrada.
  - Ex:

```
char *a;
scanf("%[^\n]s",a);
```

Limpando o buffer do teclado

- Usando a função setbuf()
  - Às vezes, podem ocorrer erros durante a leitura de caracteres ou strings do teclado. Para resolver esses pequenos erros, podemos limpar o buffer do teclado (entrada-padrão) usando a função setbuf(stdin, NULL) antes de realizar a leitura de caracteres ou strings:
  - Ex:

```
char ch;
setbuf(stdin, NULL);
scanf("%c", &ch);
```

```
char str[10];
setbuf(stdin, NULL);
gets(str);
```

Lendo uma string do teclado

- Usando a função gets()
  - Uma alternativa mais eficiente para a leitura de uma string é a função gets(), a qual faz a leitura do teclado considerando todos os caracteres digitados (incluindo os espaços em branco) até encontrar uma tecla <Enter>:
  - Ex:

```
char str[20];
gets(str);
```

Funções de manipulação de caracteres

```
char a;
```

- `isalpha(a)`: testa se o caractere é uma letra.
- `isdigit(a)`: testa se o caractere é um algarismo.
- `isespace(a)`: testa se é o caractere espaço `' '`.
- `islower(a)`: testa se é uma letra minúscula.
- `isupper(a)`: testa se é uma letra maiúscula.
- `tolower(a)`: converte o caractere para minúscula.
- `toupper(a)`: converte o caractere para maiúscula.

Estas funções estão definidas na biblioteca `ctype.h`.

Funções para manipulação de strings

Nome	Função
strcpy(s1, s2)	Copia s2 em s1.
strcat(s1, s2)	Concatena s2 ao final de s1.
strlen(s1)	Retorna o tamanho de s1.
strcmp(s1, s2)	Retorna 0 se s1 e s2 são iguais; menor que 0 se s1<s2; maior que 0 se s1>s2.
strchr(s1, ch)	Retorna um ponteiro para a primeira ocorrência de ch em s1.
strstr(s1, s2)	Retorna um ponteiro para a primeira ocorrência de s2 em s1.

Essas funções usam o cabeçalho padrão **STRING.H**.

Operações em string: biblioteca string.h

- Tamanho:
  - size_t strlen(const char * str);
    - Retorna o número de caracteres da string passada.
  - Ex:

```
char s1[80], s2[80];

gets(s1);

gets(s2);

printf("comprimentos: %d %d\n", strlen(s1), strlen(s2));
```

- Cópia:
  - char * strcpy(char *, const char *);
    - Copia todo o conteúdo da segunda string para a primeira string.
  - char * strncpy(char *, const char *, size_t);
    - Copia os “n”primeiros caracteres(indicados por size_t) da segunda string para a primeira string.
  - int sprintf(char *, const char *, …);
    - Grava dados formatados em uma string.

Operações em string: biblioteca string.h

- Concatenação:
  - char * strcat(char *, const char *);
    - Concatena o conteúdo da segunda string no final da primeira string.
  - char * strncat(char *, const char *, size_t);
    - Adiciona (concatena) os “n” primeiros caracteres da segunda string no final da primeira string.
- Inversão:
  - char * strrev(char *);
    - Inverte a string passada sobre ela mesma.

Operações em string: biblioteca string.h

- Busca:
  - char *strchr(const char *str, int ch);
    - Retorna um ponteiro para a localização em que o caractere 'ch' aparece pela primeira vez na string apontada por *str, ou NULL se não encontrar.
  - char *strrchr(const char *str, int ch);
    - Faz a mesma coisa da função anterior, mas ao invés de localizar a primeira ocorrência de 'ch', localiza e retorna a última ocorrência.
  - char * strstr(const char * strOrigem, char * strChave);
    - Retorna um ponteiro para a primeira ocorrência da string apontada por strChave na string apontada por strOrigem.

Operações em string: biblioteca string.h

- Comparação:
  - strcmp(s1,s2);
    - Retorna 0 se s1 e s2 são iguais; menor que 0 se s1<s2; maior que 0 se s1>s2 (comparação alfabética).
  - stricmp(s1,s2);
    - Retorna 0 se s1 e s2 são iguais; menor que 0 se s1<s2; maior que 0 se s1>s2 (comparação alfabética). Essa função considera letras maiúsculas ou minúsculas como símbolos iguais.

Exercício 3 – Ling. C (Strings)

1) Faça um programa que leia uma string e a imprima na tela.

2) Faça um programa que leia uma string e imprima as quatro primeiras letras dela.

3) Sem usar a função strlen(), faça um programa que leia uma string e imprima quantos caracteres ela possui.

4) Faça um programa que leia uma string e a imprima de trás para a frente.

5) Faça um programa que leia uma string e a inverta. A string invertida deve ser armazenada na mesma variável. Em seguida, imprima a string invertida.

6) Leia uma string do teclado e conte quantas vogais (a, e, i, o, u) ela possui. Entre com um caractere (vogal ou consoante) e substitua todas as vogais da palavra dada por esse caractere. Ao final, imprima a nova string e o número de vogais que ela possui.

7) Faça um programa que leia uma string e imprima uma mensagem dizendo se ela é um palíndromo ou não. Um palíndromo é uma palavra que tem a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita.

   Exemplos: ovo, arara, rever, asa, osso, ana etc.

8) Leia dois nomes (strings de caracteres), compare se os mesmos são iguais e guarde o resultado em uma variável lógica “saoIguais”. Por fim, teste se “saoIguais” é VERDADEIRO: Se sim, imprima: “Nomes iguais”. Caso contrário, imprima “Nomes diferentes”.

Exercício 3 – Ling. C (Strings)

9) Construa um programa que leia duas strings do teclado. Imprima uma mensagem informando quantas vezes a segunda string lida está contida dentro da primeira.

10) Escreva um programa que leia uma string do teclado e converta todos os seus caracteres em maiúscula. Dica: subtraia 32 dos caracteres cujo código ASCII está entre 97 e 122.

11) Escreva um programa que leia uma string do teclado e converta todos os seus caracteres em minúscula. Dica: some 32 dos caracteres cujo código ASCII está entre 65 e 90.

12) Construa um programa que leia duas strings do teclado. Imprima uma mensagem informando se a segunda string lida está contida dentro da primeira.

13) Escreva um programa que leia o nome e o valor de determinada mercadoria de uma loja. Sabendo que o desconto para pagamento à vista é de 10% sobre o valor total, calcule o valor a ser pago à vista. Escreva o nome da mercadoria, o valor total, o valor do desconto e o valor a ser pago à vista.

14) Escreva um programa que recebe uma string S e dois valores inteiros não negativos i e j. Em seguida, imprima os caracteres contidos no segmento que vai de i a j da string S.

FIM