unipampa

Universidade Federal do Pampa

Disciplina de Algoritmos e Programação

unipampa
Universidade Federal do Pampa

Último Conteúdo

- Estruturas de condição
  - Estrutura condicional simples
    - Utilização da estrutura de condição “if” com expressões lógicas simples
    - Utilização do comando “if” com expressões lógicas compostas (&& e ||)
  - Estrutura condicional composta
    - Utilização da cláusula “else” na estrutura “if”
    - Comandos “if” aninhados
  - Estrutura de seleção múltipla
    - Utilização da estrutura de condição “switch”

unipampa
Universidade Federal do Pampa

Aula de Hoje

- Teórica
  - Funções auxiliares para manipulação de String
  - Estruturas de Repetição
    - while
    - do ... while
    - for
  - Teste de Mesa

unipampa
Universidade Federal do Pampa

Funções de Manipulação de Strings

- As funções de manipulação de *strings* são funções pré-definidas presentes na biblioteca string.h

- Necessário incluir a biblioteca string.h no programa fonte

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

unipampa
Universidade Federal do Pampa

Funções de Manipulação de Strings

- gets (Leitura de Strings)
  - Função pré-definida na biblioteca string.h que permite a leitura de strings com espaços
  - Sintaxe:

gets(char *string);

  - Ex.: char string[20];

gets(string); // Teclado = aula de algoritmos

printf("%s\n",string); // Tela = aula de algoritmos

unipampa
Universidade Federal do Pampa

Funções de Manipulação de Strings

```
#include <stdio.h>

   int main()
   {
   char str[20];
   scanf("%[A-Z a-z]",str);
   printf("A string digitada foi: %s\n", str);
   return 0;
   }
```

/*ler caracteres de todo alfabeto em maiúsculo, o espaço em branco e minúsculo, respectivamente */

unipampa
Universidade Federal do Pampa

Funções de Manipulação de Strings

```
#include <stdio.h>

   int main(){
   ...
   scanf("%[^\n]", str);
   ...
   }
```

/* A instrução [^\n] diz ao comando scanf() para ler tudo até encontrar retorno de carro (ENTER, representado pelo símbolo '\n' */

unipampa
Universidade Federal do Pampa

Funções de Manipulação de Strings

```
#include <stdio.h>

   int main(){
   ...
   scanf("%[^\n]", str);
   ...
   }
```

/* A instrução [^\n] diz ao comando scanf() para ler tudo até encontrar retorno de carro (ENTER, representado pelo símbolo '\n' */

// fflush(stdin)

Funções de Manipulação de Strings

- puts (Escrita de Strings)
  - Função pré-definida na biblioteca string.h que permite a impressão de uma string na tela
  - Sintaxe:

```
puts(char *string);
```

  - Ex.: char string[20]="Caracteres";

```
puts(string); // Tela = caracteres
```

Funções de Manipulação de Strings

- strcpy (Cópia de Strings)
  - Função pré-definida na biblioteca string.h que permite copiar uma string para outra
  - Sintaxe:

```
char *strcpy(char *destino, const char *origem);
```

  - Ex.:

```
char destino[20];
strcpy(destino,"String origem");
printf("%s\n",destino); // Tela = String origem
```

Funções de Manipulação de Strings

- strcat (Concatenação de Strings)
  - Função pré-definida na biblioteca string.h que permite a concatenação de duas strings
  - Sintaxe:

```
char *strcat(char *destino, const char *origem);
```

  - Ex.:

```
char origem[10]="Fim",destino[20]="Inicio";
strcat(destino,origem);
printf("%s\n",destino); // Tela = InicioFim
```

Funções de Manipulação de Strings

- strcmp (Comparação de Strings)
  - Função pré-definida na biblioteca string.h que permite a comparação de duas strings
    - Retorna 0 se as duas strings forem iguais
    - Retorna -1 se a 1ª string for menor (alfabeticamente) que a 2ª
    - Retorna 1 se a 1ª string for maior (alfabeticamente) que a 2ª
  - Sintaxe:

```
int strcmp(char *string1, char *string2);
```

  - Ex.: char string1[10]="Carro",string2[10]="Moto";

```
printf("%d\n", strcmp(string1,string2)); // Tela = -1
```

Funções de Manipulação de Strings

- strlen (Comprimento de Strings)
  - Função pré-definida na biblioteca string.h que retorna o comprimento de uma string
  - Sintaxe:

```
int strlen(char *string);
```

  - Ex.:

```
char string[10]="Fim";
printf("%d\n",strlen(string)); // Tela = 3
```

unipampa
Universidade Federal do Pampa

Exercícios

- Faça um programa que leia um nome e um sobrenome e mostre na tela:
  - Nome completo, guardado em uma string diferente das lidas
  - Quantidade de caracteres do nome completo
    - RESTRIÇÃO: NÃO USE A BIBLIOTECA STDIO.H

unipampa
Universidade Federal do Pampa

Comando “while”

- Sintaxe:

```
while(condição)
    comando; // este será repetido enquanto a condição for V
```

– Em geral:

```
while(condição) {
    comando_1;
    comando_n;
}
```

Observação:
No “while”, para que o comando (ou bloco de comandos seja executado, pelo menos uma vez, é necessário que a condição seja verdadeira pelo menos uma vez

Comando “do ... while”

- Sintaxe:

```
do
    comando;
while(condição); // executa novamente o comando se a condição for V
```

  - Em geral:

```
do {
    comando_1;
    comando_n;
} while(condição);
```

Observação:
No “do ... while”, ao contrário do “while”, é garantido que, pelo menos uma vez, o comando (ou bloco de comandos) será executado.

unipampa
Universidade Federal do Pampa

Comando “for”

- Sintaxe:

```
for (comando_inicial; condição; comando_final)
    comando; // este será repetido enquanto a condição for V
```

– Em geral:

```
for (i = valor_inicial ; condição em relação a i; inc/dec i) {
    comando_1;
    comando_n;
}
```

unipampa
Universidade Federal do Pampa

Diferenças e Semelhanças

```
?
    {
        comando_1;
        comando_n;
    }
?
```

unipampa
Universidade Federal do Pampa

Exemplo de Equivalência

```
i=0; //inicialização
while (i<10) {
    comando_1;
    comando_n;
    i++; // incremento
}
```

```
for(i=0; i<10; i++) {
    comando_1;
    comando_n;
}
```

unipampa
Universidade Federal do Pampa

Exercícios

A) Faça um programa em Linguagem C que leia um número inteiro positivo (n) e que apresente na tela os números inteiros existentes no intervalo fechado entre 0 e n, um por linha, em ordem crescente. Se for lido um número que não seja inteiro positivo, o programa deve informar que o número lido não é inteiro positivo e terminar.

1) Usando o comando “while”
2) Usando o comando “do ... while“
3) Usando o comando “for”

unipampa
Universidade Federal do Pampa

1) Usando “while”

```
#include <stdio.h>
int main(void) {
    int n,i;
    printf(“Digite um número inteiro positivo: “);
    scanf(“%d”,&n);
    if (n<1) {
        printf(“Número não é inteiro positivo\n”);
    } else {
        i=0;
        while (i<=n) {
            printf(“%d\n”,i);
            i++;
        }
    }
    return 0;
}
```

unipampa
Universidade Federal do Pampa

2) Usando “do ... while”

```
#include <stdio.h>
int main(void) {
    int n,i;
    printf(“Digite um número inteiro positivo: “);
    scanf(“%d”,&n);
    if (n<1) {
        printf(“Número não é inteiro positivo\n”);
    } else {
        i=0;
        do {
            printf(“%d\n”,i);
            i++;
        } while (i<=n);
    }
    return 0;
}
```

unipampa
Universidade Federal do Pampa

3) Usando “for”

```
#include <stdio.h>
int main(void) {
    int n,i;
    printf(“Digite um número inteiro positivo: “);
    scanf(“%d”,&n);
    if (n<1) {
        printf(“Número não é inteiro positivo\n”);
    } else {
        for (i=0; i<=n; i++)
            printf(“%d\n”,i);
    }
    return 0;
}
```

unipampa
Universidade Federal do Pampa

Comparação

Com "while"

```
i=0;
while (i<=n) {
    printf("%d\n",&n);
    i++;
}
```

Com "do ... while"

```
i=0;
do {
    printf("%d\n",i);
    i++;
} while (i<=n);
```

Com "for"

```
for (i=0; i<=n; i++)
    printf("%d\n",i);
```

unipampa
Universidade Federal do Pampa

Exercícios

B) Faça um programa em Linguagem C que leia um número inteiro positivo (n) e que apresente na tela os números inteiros existentes no intervalo fechado entre 0 e n, um por linha, em ordem decrescente. Se for lido um número que não seja inteiro positivo, o programa deve informar que o número lido não é inteiro positivo e terminar.

1) Usando o comando “while”
2) Usando o comando “do ... while“
3) Usando o comando “for”

unipampa
Universidade Federal do Pampa

Exercícios

C) Faça um programa em Linguagem C que leia um número inteiro positivo (n) e que apresente na tela os números inteiros existentes no intervalo aberto entre 0 e n, um por linha, em ordem crescente. Se for lido um número que não seja inteiro positivo, o programa deve informar que o número lido não é inteiro positivo e terminar.

1) Usando o comando “while”
2) Usando o comando “do ... while“
3) Usando o comando “for”

unipampa
Universidade Federal do Pampa

Exercícios

D) Faça um programa em Linguagem C que leia números reais e que, conforme vão sendo lidos os números, apresente a soma dos valores já lidos. Quando o usuário digitar o valor 0, o programa deve terminar.

1) Usando o comando “while”
2) Usando o comando “do ... while“
3) Usando o comando “for”

unipampa
Universidade Federal do Pampa

Exercícios

E) Faça um programa em Linguagem C que leia números reais e que, conforme vão sendo lidos os números, apresente a média dos valores já lidos. Quando o usuário digitar o valor 0, o programa deve terminar.

1) Usando o comando “while”
2) Usando o comando “do ... while“
3) Usando o comando “for”

unipampa
Universidade Federal do Pampa

3) Usando “for”

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
      float media=0,valor,i;
      printf("Digite um número real: ");
      scanf("%f",&valor);
      for (i=1;valor!=0;i++) {
       media=(valor+(media*(i-1)))/i;
       printf("Média atual = %f\n",media);
       printf("Digite um número real: ");
         scanf("%f",&valor);
    }
      system("PAUSE");
      return 0;
}
```

unipampa
Universidade Federal do Pampa

Teste de Mesa

- Conforme os programas ganham complexidade, mais complicada fica a “depuração” – processo de busca e correção de erros de programação
- “Teste de Mesa”
  - Método adotado para auxiliar na depuração de um programa e/ou do algoritmo
  - Simulação de todos os passos: entradas, comandos e instruções do algoritmo; a fim de saber se ele chega ao resultado a que se propõe e se a lógica está correta

unipampa
Universidade Federal do Pampa

Teste de Mesa

- Simulação
  - Preencher uma tabela com valores para as variáveis e seguir o fluxo de execução do programa, simulando a execução de cada instrução
  - Refazendo o que o computador faria ao executar cada instrução
- A cada comando simulado (executado), os valores das variáveis na tabela devem ser atualizados
- Se, para uma instrução executada, uma ou mais variáveis não ficarem com os valores esperados, há um erro na lógica do algoritmo