

Introdução à Programação

AULA 6 – Estruturas de Repetição

Prof^a. Glaucia M. M. Campos

glaucia.melissa@gmail.com

Contextualização (1/3)...

Com o conhecimento adquirido até aqui, ainda não é possível fazer iterações!

Contextualização (2/3)...

Como escrever algoritmos
que façam iteração
de instruções?

Contextualização (3/3)...

Usar Estruturas
de Repetição!

Conceitos Fundamentais

- Existem situações que o processamento de um valor uma única vez não é suficiente para resolver o problema.
- Quando isto ocorre, deve-se utilizar uma **estrutura de repetição**.
- Estruturas de repetição permitem que um ou mais instruções sejam executadas um número definido de vezes, ou enquanto uma determinada condição não é alcançada.

Conceitos Fundamentais

- As estruturas de repetição também são conhecidas como **Laços** ou **Loops**
- As estruturas de repetição das linguagens de programação são de dois tipos
 - Condicional
 - Repetem até satisfazer a condição de repetição
 - São usadas quando não se sabe previamente quantas vezes
deve-se executar as instruções do bloco de repetição

Conceitos Fundamentais

- As estruturas de repetição também são conhecidas como **Laços** ou **Loops**
- As estruturas de repetição das linguagens de programação são de dois tipos
 - Contada
 - Repetem um número contado (pré-definido) de vezes
 - São usadas quando se sabe previamente quantas vezes
deve-se executar as instruções do bloco de repetição

Conceitos Fundamentais

- As estruturas de repetição condicional podem executar seus testes:
 - No Início
 - Enquanto/Faça
 - No Fim
 - Repita/Até
- As estruturas de repetição contada utilizam uma variável para controlar a quantidade de repetições
 - Para/Faça

Repetição Condicional

- Normalmente, a utilização de uma estrutura de repetição requer o uso de dois tipos de variáveis:
 - **Contadora:** é inicializada antes da estrutura de repetição e é incrementada no interior desta por um valor constante
 - **Acumuladora:** é inicializada antes da estrutura de repetição e é incrementada no interior desta por um valor variável

Repetição Condicional

- Exemplo:

Programa variavelContadora.c

```
...  
cont = 0  
<estrutura de repetição>  
    ...  
    cont = cont + 1  
    ...  
<fim estrutura de repetição>  
...
```

Programa variavelAcumuladora.c

```
...  
soma = 0  
<estrutura de repetição>  
    ...  
    soma = soma + X  
    ...  
<fim estrutura de repetição>  
...
```

Repetição Condicional

- Estrutura de Repetição **do/while**
- Sintaxe:

```
do{  
    <comando 1>;  
    <comando 2>;  
    ...  
}while(condicao for satisfeita);
```

Repetição Condicional

- Estrutura de Repetição **do/while**
- Semântica:
 - Efetue um teste lógico no fim do laço, garantindo que pelo menos uma vez as instruções deste são executadas. Ao contrário do enquanto, esta só repete o laço se o resultado do teste for F.

Repetição Condicional

- Estrutura de Repetição **do/while**
 - Não se sabe de antemão quantas vezes o bloco de repetição será executado. Todavia é garantido que ele será executado pelo menos uma vez
 - Testa a condição depois de entrar na estrutura de repetição
 - Repete a execução do bloco de instruções toda vez que a condição for F
 - A execução do bloco é finalizada quando a condição for V

Repetição Condicional

- Estrutura de Repetição **do/while**

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

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

```
int main(){  
    int num;  
    do{  
        printf("Digite numero: ");  
        scanf("%d", &num);  
    }while(num<=0);  
    system("PAUSE");  
    return 0;  
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>

int main(){
    int num, fat, i;
    do{
        printf("Digite numero: ");
        scanf("%d", &num);
    }while( (num<0) || (num>10) );

    fat=i=1;
    do{
        fat*=i;
        i++;
    }while(i<=num);
    printf("Fatorial de %d: %d", num, fat);
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>

int main() {
    int num_termos, a, b, c, i;
    do {
        printf("Digite numero: ");
        scanf("%d", &num_termos);
    } while (num_termos <= 0);
    a = b = 1;
    printf("%d, %d ", a, b);
    i = 3;
    do {
        c = a + b;
        printf(", %d", c);
        a = b;
        b = c;
        i++;
    } while (i <= num_termos);
    printf("\n");
    system("PAUSE");
    return 0;
}
```


Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>
#include <stdlib.h>
//imprimir caracteres ASCII
int main(){
    int i;
    i=0;
    do{
        printf("%c ", i);
        i++;
    }while(i<256);
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>
#include <stdlib.h>
//verificar se um número maior que zero é primo
int main(){
    int num, i, acm;
    char op;
    do{
        do{
            printf("Digite numero: ");
            scanf("%d", &num);
        }while((num<0));
        i=1;
        acm=0;
        do{
            if(num%i==0)
                acm+=1;
            i+=1;
        }while(i<=num);
        if(acm==2)
            printf("Numero primo!");
        else
            printf("Numero nao primo!");
        printf("\nDeseja verificar se outro numero e primo? (S/N) ");
        op=getche();
        op=toupper(op);
        printf("\n");
    }while(op=='S');
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

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

int main(){
    int i;
    float sal, novo_sal;
    i=1;
    do{
        printf("Digite salario do funcionario: ");
        scanf("%f", &sal);
        if(sal<1500)
            novo_sal=sal+0.35*sal;
        else if((sal>=1500)&&(sal<3000))
            novo_sal=sal+0.15*sal;
        else if((sal>=3000)&&(sal<=5000))
            novo_sal=sal+0.05*sal;
        else {
            novo_sal=sal;
            printf("Voce nao tera aumento!");
        }
        printf("Seu salario e: %f", novo_sal);
        printf("\n");
        i++;
    }while(i<=50);
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

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

int main(){
    int num, i;
    printf("Digite o numero para a tabuada: ");
    scanf("%d", &num);
    i=0;
    do{
        printf("%d * %d = %d", num, i, num*i);
        printf("\n");
        i++;
    }while(i<=10);
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros entre 100 e 200
int main(){
    int soma, i;
    i=100;
    soma=0;
    do{
        soma=soma+i;
        i++;
    }while(i<=200);
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **do/while**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros pares entre 100 e 200
int main(){
    int soma, i;
    i=100;
    soma=0;
    do{
        if(i%2==0)
            soma=soma+i;
        i++;
    }while(i<=200);
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**
- Sintaxe

```
while (condicao for satisfeita){  
    <comando 1>;  
    <comando 2>;  
    ...  
    <comando n>;  
}
```

Repetição Condicional

- Estrutura de Repetição **while**
- Semântica
 - Faz o teste no início do laço. Se o resultado for V as instruções do laço e/são executadas. Volta-se para o início do laço e testa-se novamente a sua condição. Isto é repetido enquanto a condição testada for V.

Repetição Condicional

- Estrutura de Repetição **while**
- Resumindo...
 - Não se sabe de antemão quantas vezes o bloco de repetição será executado. Isto é, ele pode ser executado várias vezes ou nenhuma vez
 - Testa a condição antes de entrar na estrutura de repetição
 - Repete a execução do bloco de instruções toda vez que condição for V
 - O bloco de instruções A execução do bloco é finalizada quando a condição for F

Repetição Condicional

- Estrutura de Repetição **while**

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

int main() {
    int num;
    num=-1;
    while(num<=0) {
        printf("Digite numero: ");
        scanf("%d", &num);
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

```
#include <stdio.h>

int main() {
    int num, fat, i;
    num=-1;
    while((num<0) || (num>10)) {
        printf("Digite numero: ");
        scanf("%d", &num);
    }
    fat=i=1;
    while(i<=num) {
        fat*=i;
        i++;
    }
    printf("Fatorial de %d: %d", num, fat);
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

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

int main(){
    int a, b, c, cont, n_termos;
    n_termos=-1;
    while(n_termos<=0){
        printf("Digite numero de termos: ");
        scanf("%d", &n_termos);
    }
    a=b=1;
    if(n_termos==1)
        printf("%d",a);
    else if(n_termos==2)
        printf("%d, %d", a, b);
    else{
        printf("%d, %d", a, b);
        cont=3;
        while(cont<=n_termos){
            c=a+b;
            printf(", %d", c);
            a=b;
            b=c;
            cont++;
        }
    }
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

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

int main(){
    int i;
    i=0;
    while(i<=255){
        printf("Numero %d e caractere ASCII %c \n", i, i);
        getche();
        i++;
    }
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

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

int main(){
    int num, cont, acm;
    char op='S';
    while (op=='S'){
        printf("Digite numero: ");
        scanf("%d", &num);
        cont=1;
        acm=0;
        while(cont<=num){
            if(cont%2==0)
                acm=acm+1;
            cont++;
        }
        if(acm==2){
            printf("Numero primo!");
        }else{
            printf("Numero nao primo!");
        }
        printf("\n");
        printf("Deseja saber se outro numero e primo?(S/N)");
        op=getche();
        op=toupper(op);
        printf("\n");
    }
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

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

int main(){
    int i;
    float sal, novo_sal;
    i=1;
    while(i<=50){
        printf("Digite salario do funcionario: ");
        scanf("%f", &sal);
        if(sal<1500)
            novo_sal=sal+0.35*sal;
        else if((sal>=1500)&&(sal<3000))
            novo_sal=sal+0.15*sal;
        else if((sal>=3000)&&(sal<=5000))
            novo_sal=sal+0.05*sal;
        else {
            novo_sal=sal;
            printf("Voce nao tera aumento!");
        }
        printf("Seu salario e: %.2f", novo_sal);
        printf("\n");
        i++;
    }
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

- Estrutura de Repetição **while**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros entre 100 e 200
int main(){
    int soma, i;
    i=100;
    soma=0;
    while(i<=200){
        soma=soma+i;
        i++;
    }
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}
```


Repetição Condicional

- Estrutura de Repetição **while**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros pares entre 100 e 200
int main(){
    int soma, i;
    i=100;
    soma=0;
    while(i<=200){
        if(i%2==0)
            soma=soma+i;
        i++;
    }
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}
```

Repetição Condicional

Nota I

Toda repetição condicional pode ser representada por uma estrutura do tipo **while** ou **do/while**. Todavia, perceba que a primeira repete somente quando a condição é V e a segunda somente quando a condição é F. Por isso, a condição de uma é a negação da outra

Repetição Condicional

Nota II

Nos laços condicionais a variável que é testada, deve estar sempre associada a uma instrução que a atualize no interior do laço. Pois, se isto não ocorrer, o algoritmo ficará repetindo indefinidamente este laço, gerando uma situação conhecida como “laço/loop” infinito”

Repetição Condicionada

- Pode a execução de um programa com processamento iterativo não terminar? Observe:

```
s = 0;  
i = 0;  
while (i < 3)  
{  
    i--;  
    s = s + i;  
}
```

Este laço ou loop
nunca irá terminar!
(Erro de lógica)

```
s = 0;  
i = 0;  
while (i > -3)  
{  
    i--;  
    s = s + i;  
}
```

Maneira
correta

Repetição Condicionada

- Em alguns casos, o loop infinito pode ser desejável. Exemplo: um programa que monitora um reator nuclear deve estar sempre em execução.
- Neste caso, pode-se escrever:

```
while (1)
{
    ...
}
```

Repetição Condicionada

- Resumindo...
 - O exemplo abaixo mostra a diferença entre as estruturas de repetição **while** e **do/while**

```
s = 0;  
i = 0;  
do  
{  
    s = s + i;  
    i++;  
}  
while (i <= N)
```

Comando
do-while

```
s = 0;  
i = 1;  
while (i <= N)  
{  
    s = s + i;  
    i++;  
}
```

Comando
while

Repetição Contada

- Estrutura de Repetição **for**
- Sintaxe

```
for(valor inicial; condição; valor do incremento/decremento) {  
    <comando 1>  
    <comando 2>  
    ...  
}
```

Repetição Contada

- Estrutura de Repetição **for**
- Semântica:
 - Repete as instruções enquanto a variável contador não atingir o valor final. Ressalta-se que a variável contador é previamente inicializada e incrementada ou decrementada de uma constante a cada repetição

Repetição Contada

- Estrutura de Repetição **for**
- Resumindo...
 - Sabe-se de antemão quantas vezes o bloco de repetição será executado. Isto é, repete enquanto o valor final não atingir o valor final da variável de controle
 - Incrementa automaticamente a variável de controle cada vez que o bloco é executado (incremento 1 é o padrão)
 - A variável de controle deve ser um número inteiro.
 - A variável de controle não pode ser modificada dentro do bloco

Repetição Contada

- Estrutura de Repetição **for**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros pares entre 100 e 200
int main(){
    int soma, i;
    for(i=100,soma=0;i<=200;i++){
        if(i%2==0)
            soma=soma+i;
    }
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}

/**
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros pares entre 100 e 200
int main(){
    int soma, i;
    for(i=100,soma=0;i<=200;i=i+2){
        soma=soma+i;
    }
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}**/
```

Repetição Contada

- Estrutura de Repetição **for**

```
#include <stdio.h>
int main(){
    int fat, cont, num;
    printf("Digite numero: ");
    scanf("%d", &num);
    for(cont=1, fat=1; cont<=num; cont++){
        fat=fat*cont;
    }
    printf("O fatorial de %d e: %d", num, fat);
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Repetição Contada

- Estrutura de Repetição **for**

```
#include <stdio.h>
int main(){
    int a, b, c, cont, n_termos;
    printf("Digite numero de termos: ");
    scanf("%d", &n_termos);
    a=b=1;
    if(n_termos==1)
        printf("%d",a);
    else if(n_termos==2)
        printf("%d, %d", a, b);
    else{
        printf("%d, %d", a, b);
        for(cont=3; cont<=n_termos; cont++){
            c=a+b;
            printf(", %d", c);
            a=b;
            b=c;
        }
        printf("\n");
        system("PAUSE");
        return 0;
    }
}
```

Repetição Contada

- Estrutura de Repetição **for**

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

int main(){
    int i;
    for(i=0;i<=255;i++){
        printf("Numero %d e caractere ASCII %c \n", i, i);
        getche();
    }
    system("PAUSE");
    return 0;
}
```

Repetição Contada

- Estrutura de Repetição **for**

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

int main(){
    int i;
    float sal, novo_sal;
    for(i=1;i<=50;i++){
        printf("Digite salario do funcionario: ");
        scanf("%f", &sal);
        if(sal<1500)
            novo_sal=sal+0.35*sal;
        else if((sal>=1500)&&(sal<3000))
            novo_sal=sal+0.15*sal;
        else if((sal>=3000)&&(sal<=5000))
            novo_sal=sal+0.05*sal;
        else {
            novo_sal=sal;
            printf("Voce nao tera aumento!");
        }
        printf("Seu salario e: %f", novo_sal);
        printf("\n");
        i++;
    }
    system("PAUSE");
    return 0;
}
```

Repetição Contada

- Estrutura de Repetição **for**

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

int main() {
    int num, i;
    char op;
    op='S';
    while(op=='S') {
        printf("Digite o numero para a tabuada: ");
        scanf("%d", &num);
        for(i=0; i<=10; i++) {

            printf("%d * %d = %d", num, i, num*i);
            printf("\n");
            i++;

        }
        printf("Deseja outro número para a tabuada? (S/N)");
        op=getche();
        op=toupper(op);
        printf("\n");
    }
    system("PAUSE");
    return 0;
}
```

Repetição Contada

- Estrutura de Repetição **for**

```
#include <stdio.h>
#include <stdlib.h>
//somatorio dos numeros entre 100 e 200
int main(){
    int soma, i;
    for(i=100, soma=0; i<=200; i++){
        soma=soma+i;
    }
    printf("Somatorio dos numeros de 100 a 200 sao: %d \n", soma);
    system("PAUSE");
    return 0;
}
```


Comparando as Estruturas de

- As estruturas de Repetição são assim comparadas:

	Enquanto/Faça	Repita/Até	Para/Faça
Tipo da Estrutura	Condicional	Condicional	Contada
Ocorrência do Teste	Início	Fim	Início
Quantidade de Repetições	0 ou muitas	Mínimo 1	$((\text{fim} - \text{início}) \div \text{passo}) + 1$
Condição para Repetir	V	F	$\text{Início} \leq \text{Fim}$ ou $\text{Início} \geq \text{Fim}$

Estruturas de Repetição

- Semelhante as estruturas de decisão composta, as estruturas de repetição também podem ser **encadeadas/aninhadas**.
- Esta abordagem é usada quando há a necessidade de se usar laços dentro de laços.
 - Por exemplo: fazer um algoritmo para gerar toda a tabuada de multiplicação de 0 a 10.

Encadeando as Estruturas

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

int main(){
    int num, i, j;
    for(i=0;i<=10;i++){
        for(j=0;j<=10;j++){

            printf("%d * %d = %d", i, j, i*j);
            printf("\n");
        }
        printf("\n\n");
        getche();

    }
    printf("\n");

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

Encadeando as Estruturas

**Também pode-se encadear
estruturas de repetição
com estruturas de decisão**

Encadeando as Estruturas

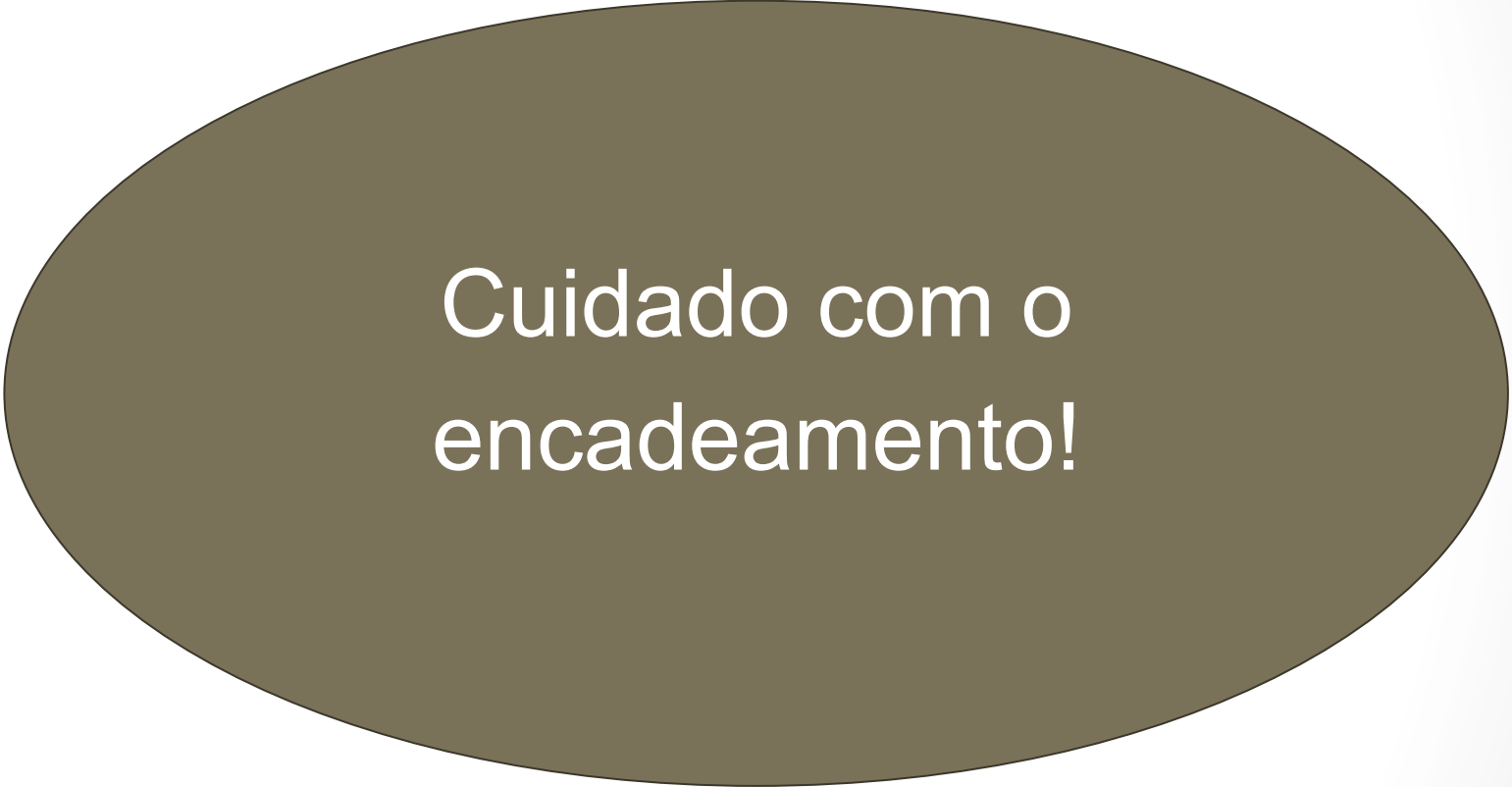
```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int r, i, j;
    for(i=0;i<=10;i++){
        for(j=0;j<=10;j++){
            r=i+j;
            if(r%2==0)
                printf("%d + %d = %d - Par", i, j, i+j);
            else
                printf("%d + %d = %d - Impar", i, j, i+j);
            printf("\n");
        }
        printf("\n\n");
        getche();

    }
    printf("\n");

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

Encadeando as Estruturas



Cuidado com o
encadeamento!

Encadeando as Estruturas

Correto



Errado



Encadeando as Estruturas

Você sabe o que é
IDENTAÇÃO?

Encadeando as Estruturas

Identação

=

Organização hierárquica das estruturas e suas instruções.



Facilita visualizar o que está contido em que.
Auxilia no entendimento do código
e na busca de erros

Lista de Exercícios

1. Fazer um algoritmo estruturado que calcule e escreva a soma dos 30 primeiros termos da série:

- $\underline{X} + \underline{4*X} + \underline{9*X} + \underline{16*X} + \underline{25*X} + \dots$
- 2 4 6 8 10

Obs.: O valor de X é fornecido como dado de entrada.

2. Anacleto tem 1,50m e cresce 2 centímetros por ano, enquanto Felisberto tem 1,10 e cresce 3 centímetros por ano. Construa um programa que calcule e apresente quantos anos serão necessários para que Felisberto seja maior que Anacleto.

3. Construa um programa que apresente a tabuada dos números de 0 a 10, apresentando a resposta no seguinte formato:

0 * 1 = 0	1 * 1 = 1
0 * 2 = 0	1 * 2 = 2
...	...
0 * 10 = 0	1 * 10 = 10

Lista de Exercícios

4. Um marciano chegou a uma floresta e se escondeu atrás de uma das 100 árvores quando viu um caçador. O caçador só tinha 5 balas em sua espingarda. Cada vez que ele atirava, e não acertava, é claro, o marciano dizia: estou mais à direita ou mais à esquerda. Se o caçador não conseguir acertar o marciano, ele será levado para Marte. Implementar este jogo para dois jogadores, onde um escolhe a árvore em que o marciano irá se esconder, e o outro tenta acertar.
5. Uma das maneiras de se conseguir a raiz quadrada de um número é subtrair do número os ímpares consecutivos a partir de 1, até que o resultado da subtração seja menor ou igual a zero. O número de vezes que se conseguir fazer a subtração é a raiz quadrada exata (resultado 0) ou aproximado do número (resultado negativo).

Exemplo: Raiz de 16 ($16-1=15-3=12-5=7-7=0$) – os números negativos representam a quantidade, que ao final será a raiz. Fazer um algoritmo que identifique a raiz de um número usando esse mecanismo.

Lista de Exercícios

6. Repare a seguinte característica do número: $3025:30 + 25 = 55$ e $55^2 = 3025$. Criar um algoritmo que possa ler vários números inteiros de 4 algarismos, um de cada vez, e diga se o número apresenta a mesma característica (repare que $3025/100=30$ com resto 25). O algoritmo termina quando for lido um valor menor que 1000 ou maior que 9999.
7. Criar um algoritmo em que o usuário digite a opção de calcular o MDC ou o MMC de dois números inteiros lidos, mostrando o resultado ao final. A interação com o usuário finaliza quando o mesmo digita 'N'. Crie um menu para apresentar as opções ao usuário.
8. Criar um algoritmo que entre com um número e verifique se ele é um número triangular. Um número é triangular quando é resultado do produto de três números consecutivos. Exemplo: $24=2*3*4$.

Lista de Exercícios

9. Criar um algoritmo que leia vários números inteiros e apresente o valor da função

$$f(x) = x!/0^2! - (x-1)!/2^2! + (x-2)!/4^2! - (x-3)!/6^2!$$

O algoritmo se encerra quando se digita um número menor do que 1. Não esqueça que o fatorial de (0) zero é (1) um.

10. Em uma eleição presidencial, existem quatro candidatos. Os votos são informados através de código. Os códigos utilizados são:

1,2,3,4 votos para os respectivos candidatos;

5 voto nulo; 6 voto em branco.

Escreva um algoritmo que calcule e imprima:

- total de votos para cada candidato;
- total de votos nulos e em branco;
- porcentagem de votos nulos sobre o total de votos;
- porcentagem de votos em branco sobre o total de votos.

Para finalizar o conjunto de votos, tem-se o valor zero.

Lista de Exercícios

- I 1. Faça um Programa para desenhar na tela a letra inicial do seu nome, desenhando com "*".
- I 2. Construa um programa que leia vários números inteiros e mostre qual foi o menor valor fornecido. Para cada valor digitado, deve ser solicitado ao usuário que ele digite se deseja continuar entrando com valores. **OBS:** Não se esqueça de validar a resposta do usuário, pois ele só pode responder “S” ou “N”.
- I 3. Desenvolver um algoritmo que efetue a soma de todos os números ímpares que são múltiplos de três e que se encontram no conjunto dos números de 1 até 500.
- I 4. Escrever um algoritmo estruturado que leia uma quantidade desconhecida de números e conte quantos deles estão nos seguintes intervalos: [0,25], [26,50], [51,75] e [76,100]. A entrada de dados deve terminar quando for lido um número negativo.

Lista de Exercícios

15. Dados os valores de L (**int**) e A (**float**), a partir de um quadrado de lado L , uma seqüência infinita de quadrados pode ser construída onde a medida do lado de cada novo quadrado é sempre a metade da medida do lado do quadrado anterior. Calcule o número de quadrados necessários na seqüência até que se obtenha um quadrado com área menor do que A .
16. Dado um número n inteiro e positivo, dizemos que n é perfeito se n for igual à soma de seus divisores positivos diferentes de n . Construa um programa em C que verifica se um dado número é perfeito. Ex: 6 é perfeito, pois $1+2+3 = 6$.
17. Imagine uma brincadeira entre dois colegas, na qual um pensa um número e o outro deve fazer chutes até acertar o número imaginado. Como dica, a cada tentativa é dito se o chute foi alto ou foi baixo. Elabore um algoritmo dentro deste contexto que leia o número imaginado e os chutes, ao final mostre quantas tentativas foram necessárias para descobrir o número.

Lista de Exercícios

18. Faça um programa em C para ler um número real e exibir uma tabela em que o número apareça multiplicado até 200, sendo 10 em cada linha. Esta tabela é útil para deixar afixada em lojas de Xerox, por exemplo.

Exemplo: Valor da Xerox: R\$ 0,06

1 = 0,06 2= 0,12 3= 0,18 10= 0,60

11 = 0,66

...

191=11,46 ... 200=12,00

19. Fazer um algoritmo que calcule e escreva a soma dos 50 primeiros termos da seguinte série:

$$\frac{1000}{1} + \frac{997}{2} + \frac{994}{3} + \frac{991}{4} + \dots$$

Lista de Exercícios

20. Criar um algoritmo que leia um conjunto de valores inteiros e positivos e cujo último valor é -1 . Dentre os valores lidos, o algoritmo deve imprimir:

- O menor valor dentre os maiores que 100 e menores que 1000;
- A média desses valores dentre os maiores que 100 e menores que 1000;
- A soma desses valores dentre os maiores que 100 e menores que 1000;
- A soma de todos os valores lidos.

O valor -1 não deve ser considerado. Se nenhum valor estiver dentro do intervalo, o algoritmo deve imprimir uma mensagem para o usuário explicando o ocorrido.

21. Faça um algoritmo para calcular a área de N Círculos. Fórmula: $\text{Área} = \pi \text{ raio}^2$, onde $\pi = 3,141592$. Defina o valor de π como sendo uma constante.

Lista de Exercícios

22. Uma pesquisa de opinião realizada no Rio Grande do Norte teve as seguintes perguntas:

- | | | |
|--------------------------------|--------------------|------------------------|
| a) Qual o seu time do coração? | b) Onde você mora? | c) Qual o seu salário? |
| 1. ABC | 1. Natal | |
| 2. América | 2. Mossoró | |
| 3. Alecrim | 3. Outros | |
| 4. Potiguar | | |
| 5. Outros | | |

Crie um algoritmo que imprima(escreva):

- O número de torcedores por clube;
- A média salarial dos torcedores do América;
- O número de pessoas de Mossoró torcedoras do Alecrim.

Obs.: O algoritmo acaba quando se digita 0 para o time.

Lista de Exercícios

23. Tem-se um conjunto de dados contendo a altura e o sexo (M ou F) de 15 pessoas. Faça um algoritmo que calcule e escreva:

- a maior e a menor altura do grupo;
- a média de altura das mulheres;
- número de homens.

24. A conversão de graus Fahrenheit para Centígrados é obtida por $c = 5/9 * (f - 32)$. Faça um algoritmo que calcule e escreva uma tabela de graus Centígrados e graus Fahrenheit, que variam de 50 a 65 de 1 em 1.

25. Faça um algoritmo para calcular a área de um triângulo. Este programa não pode permitir a entrada de dados inválidos, por exemplo, medidas menores ou iguais a zero.

Lista de Exercícios

26. Construa um programa que calcule e mostre a média de 10 alunos (três notas, média aritmética)
27. Construa um programa que calcule a media de 10 alunos e mostre a maior media (três notas, media aritmética)
28. Construa um programa que calcule a media de n alunos e mostre quantos foram reprovados, quantos passaram por media e quantos ficaram em recuperação (Reprovado – media < 3, Recuperação – $3 \leq \text{media} < 7$, Aprovado – media ≥ 7)
29. Entrar com quatro números e imprimir o cubo e a raiz cúbica de cada número.
30. Ler o número de termos da série (N) e imprimir o valor de S, sendo:
$$S = (1/4) + (2/8) + (3/12) + (4/16) \dots$$
31. Entrar com números enquanto forem ímpares e imprimir quantos números ao todo e quantos números ímpares foram digitados.

Lista de Exercícios

32. Ler vários números até entrar o número -999. Para cada número, imprimir seus divisores.
33. Criar um algoritmo que receba a idade, a altura e o peso de várias pessoas. Calcule e imprima:
- Quantidade de pessoas com idade superior a 50 anos
 - A média das alturas das pessoas com idade entre 20 e 50 anos
 - A porcentagem de pessoas com peso inferior a 40 quilos entre todas as pessoas analisadas.

Lista de Exercícios

34. Uma pousada estipulou o preço para sua diária em R\$ 30,00 e mais uma taxa de serviços diários de:

R\$ 15,00, se o número de dias for menor do que 10

R\$ 8,00 se o número de dias for maior ou igual a 10

Criar um algoritmo que imprima o código do cliente, conta e o número da conta de cada cliente e ao final o total ganho pela pousada.

35. Entrar com números e imprimir o quadrado de cada número até entrar um número múltiplo de 6 que deverá ter seu quadrado também impresso.

Referências

- Medina, Marco; Fertig, Cristina. Algoritmos e Programação: teoria e prática. São Paulo: Novatec Editora, 2006.
- Lopes, Anita; Garcia, Guto. Introdução à Programação: 500 algoritmos resolvidos. Rio de Janeiro: Editora Campus, 2002.
- Mizrani, Victorine Viviane. Treinamento em Linguagem C, Módulo I. Editora Makron Books.
- Transparências modificadas do professor Dr. Flavio Luiz Cardeal Pádua, do Centro Federal de Educação Tecnológica de Minas Gerais
- Transparências modificadas do professor Robson Fidalgo, da UFRPE.