

Algoritmos e Estruturas de Dados

Disciplina 301477

Programa de Pós-graduação em
Computação Aplicada

Prof. Alexandre Zaghetto

<http://alexandre.zaghetto.com>
zaghetto@unb.br



<http://www.nickgentry.com/>

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

O presente conjunto de *slides* não pode ser reutilizado ou republicado sem a permissão do instrutor.

Módulo 04

Algoritmos com Repetições

1. Repetições com Teste no Início

- Forma geral em C:

while (<condição>) {

<comandos>

}

1. Repetições com Teste no Início

Exemplo 1: Escreva um programa que solicita ao usuário N notas, calcula e imprime na tela do computador a média da turma. O programa deve continuar solicitando notas até que o valor -1 seja digitado pelo usuário.

1. Repetições com Teste no Início

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

int main()
{
    float numero, soma = 0, media = 0;
    int N = 0;

    printf("Escreva uma nota:");
    scanf("%f", &numero);

    while(numero != -1) {
        N++;
        soma = soma + numero;
        printf("Escreva uma outra nota:");
        scanf("%f", &numero);
    }
```

1. Repetições com Teste no Início

```
media = soma/N;  
printf("A media eh: %f \n", media);  
  
return 0;  
}
```

1. Repetições com Teste no Início

Exemplo 2: O número 3025 possui a seguinte característica :

$$30 + 25 = 55$$

$$55^2 = 3025$$

Escreva um programa que pesquise e imprima todos os números de quatro dígitos que apresentam tal característica.

1. Repetições com Teste no Início

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

int main()
{
    int N, N1, N2;
    float Soma2;

    N = 1000;
    while (N <= 9999) {

        N1 = N/100;
        N2 = N-(100*N1);

        Soma2 = pow(N1+N2, 2);
```



1. Repetições com Teste no Início

```
if (Soma2 == N) {  
    printf("N1: %d \n", N1);  
    printf("N2: %d \n", N2);  
    printf("Soma2: %.0f \n", Soma2);  
}  
  
    N++;  
}  
  
return 0;  
}
```

1. Repetições com Teste no Início

Exemplo 3: Achar o maior e o menor número de uma série de números positivos fornecidos pelo usuário via teclado. O programa deve solicitar valores até que o número -1 seja fornecido.

1. Repetições com Teste no Início

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

int main()
{
    int N, maior, menor;

    printf("Digite um numero:");
    scanf("%d", &N);
    maior = N;
    menor = N;
```



1. Repetições com Teste no Início

```
while(N != -1) {  
    if (N < menor) {  
        menor = N;  
    } else if (N > maior) {  
        maior = N;  
    }  
    printf("Digite outro numero:");  
    scanf("%d", &N);  
}  
printf("Maior: %d \n", maior);  
printf("Menor: %d \n", menor);  
  
return 0;  
}
```

2. Mais sobre Tipos de Variáveis e Códigos de Formatação

Tipo	bits	Precisão
int	32 bits	$-(2^{31})$ a $(2^{31} - 1)$ -2.147.483.648 a 2.147.483.647
char	8 bits	-128 a 127
float	32 bits	6 a 7 dígitos significativos
double	64 bits	15 a 16 dígitos significativos
unsigned int	32 bits	0 a $(2^{32} - 1)$ 0 a 4.294.967.295
unsigned char	8 bits	0 a 255
short int	16 bits	$-(2^{15})$ a $(2^{15} - 1)$ -32.768 a 32.767
unsigned short int	16 bits	0 a $(2^{16} - 1)$ 0 a 65.535

2. Mais sobre Tipos de Variáveis e Códigos de Formatação

Tipo	Códigos de formatação para o printf()
int	%d ou %i
char	%c
float	%f
double	%lf
unsigned int	%u
unsigned char	%c
short int	%hi
unsigned short int	%hu

2. Mais sobre Tipos de Variáveis e Códigos de Formatação

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

int main() {

    int          x1 = -2147483648;
    char         x2 = 'A';
    float        x3 = 3.141592;
    double       x4 = 3.141592653589793;
    unsigned int x5 = 4294967295;
    unsigned char x6 = 255;
    short int    x7 = 32767;
    unsigned short int x8 = 65535;

    printf("x1: %d \n", x1);    // int
    printf("x2: %c \n", x2);    // char
    printf("x3: %f \n", x3);    // float
    printf("x4: %.15lf \n", x4); // double
    printf("x5: %u \n", x5);    // unsigned int
    printf("x6: %u \n", x6);    // unsigned char
    printf("x7: %u \n", x7);    // short int
    printf("x8: %hu \n", x8);   // unsigned short int

    return 0;
}
```


3. Repetições com Teste no Final

- Forma geral em C:

do{

<comandos>

} while (<condição>;

3. Repetições com Teste no Final

Exemplo 4: Escreva um algoritmo que solicita ao usuário valores inteiros positivos N e conta a quantidade de número pares e a quantidade de números ímpares digitados. O usuário deve continuar fornecendo novos valores até que o -1 seja digitado. O algoritmo deve mostrar ao final quantos números pares e quantos números ímpares foram digitados.



3. Repetições com Teste no Final

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

int main( )
{
    int Npares = 0, Nimpares = 0, Numero;

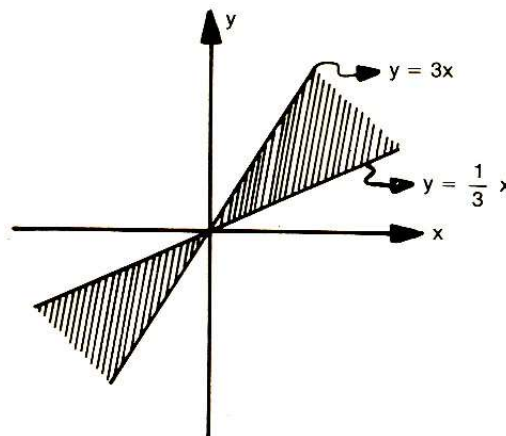
    do{
        printf("Digite um numero: ");
        scanf("%d", &Numero);
        if (Numero > 0)
        {
            if (Numero%2 == 0)
                Npares++;
            else
                Nimpares++;
        }
    } while(Numero != -1);
```

3. Repetições com Teste no Final

```
printf("Numero de pares: %d \n", Npares);  
printf("Numero de ímpares: %d \n", Nimpares);  
  
return 0;  
}
```

3. Repetições com Teste no Final

Exemplo 5 (pro lar): Escreva um algoritmo que repetidamente leia coordenadas (x,y) fornecidas pelo usuário e escreva na tela do computador "INTERIOR" ou "EXTERIOR", caso o ponto pertença ou não à região sombreada abaixo, respectivamente. A solicitação das coordenadas deve prosseguir até que o usuário solicite a interrupção. Utilize `faca...enquanto` (`do...while`) na implementação do laço.



4. Repetições com Variável de Controle

- Forma geral em C:

```
for(<inicialização>; <condição>; <incremento>) {
```

```
    <comandos>
```

```
}
```

4. Repetições com Variável de Controle

Exemplo 6: Escreva um algoritmo que solicita ao usuário um valor inteiro positivo N e imprime na tela do computador todos os número inteiros de 0 a N .



4. Repetições com Variável de Controle

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

int main()
{
    int N, i;

    printf("Digite o valor de N:");
    scanf("%d", &N);

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

    return 0;
}
```




4. Repetições com Variável de Controle

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

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

4. Repetições com Variável de Controle

Exemplo 7: Escreva um algoritmo que mostre a tabuada de 2 a 10 na tela do computador.

4. Repetições com Variável de Controle

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

int main()
{
    int i, j;

    for(i = 2; i<=10 ; i++) {
        printf("*****\n");
        printf("Tabuada de %d\n", i);
        printf("*****\n\n");

        for(j = 1; j<=10 ; j++) {
            printf("%d x %d = %d \n", i, j, i*j);
        }
        printf("\n");
    }
    return 0;
}
```

4. Repetições com Variável de Controle

Exemplo 8: Calcule o $\cos(x)$ por meio de 5 termos da série (Maclaurin expansion) abaixo e escreva a diferença entre o valor calculado por essa série e o valor dado pela função $\cos(x)$ de `<math.h>`.

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

4. Repetições com Variável de Controle

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

int main()
{
    float angulo, angrad, valcos, valserie, PI=3.14159265;
    float sinal, fatorial, termo;
    int i, j, Ntermos, Den;

    // Solicita o angulo em graus
    printf("Entre com o valor de um angulo:");
    scanf("%f", &angulo);

    // Converte para radianos
    angrad = angulo*PI/180;
```

4. Repetições com Variável de Controle

```
// Calcula o cosseno utilizando a função cos de math.h  
valcos = cos(angrad);
```

```
/* Calcula o valor máximo do denominador,  
dada a quantidade de termos a serem adicionados */  
Ntermos = 5;  
Den = (Ntermos*2)-2;
```

```
// Primeiro termo da serie  
valserie = 1;
```

```
/* Dado que a cada interação o sinal  
do termo muda faz-se necessário declarar  
uma variável para controlar isso */  
sinal = 1;
```



4. Repetições com Variável de Controle

```
// Inicializa o valor do fatorial com 1
fatorial = 1;

// Calcula um novo termo a cada valor de denominador
for(i = 2; i<=Den ; i+=2) {

    /* O calculo do novo termo depende de um fatorial.
    O fatorial do novo termo é calculado a partir do
    valor do fatorial anteriormente calculado.
    Isso aumenta a eficiente do programa
    */

    for (j = i-1; j<= i; j++){
        fatorial = fatorial*j;
    }
```



4. Repetições com Variável de Controle

```
// Inverte o sinal
sinal = -sinal;

// Calcula o novo termo
termo = sinal*pow(angrad,i)/fatorial;

// Acrescenta o novo termo ao somatório
valserie = valserie + termo;
}

printf("\nFuncao cos(x): %f \n", valcos);
printf("Calculado      : %f \n\n", valserie);

return 0;
}
```


5. Repetições com Variável de Controle e Teste no Início

Exemplo 9: Calcule o valor aproximado de $\cos(x)$ por meio da série abaixo e com uma precisão menor que 0.001. A precisão é aqui definida como o valor absoluto da diferença entre duas aproximações consecutivas de $\cos(x)$. Considere também que o cálculo do fatorial em uma iteração deve ser realizado a partir do fatorial calculado na iteração anterior. O valor do ângulo x deve ser fornecido em graus, mas o valor de $\cos(x)$ deve ser realizado para x em radianos. Inclua instruções para realizar a conversão.

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

5. Repetições com Variável de Controle e Teste no Início

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

int main()
{
    float angulo, angrad, valcos, valserie;
    float PI=3.14159265;
    float sinal, fatorial, termo, erro, precisao = 0.0001;
    int i, j;

    // Solicita o angulo em graus
    printf("Entre com o valor de um angulo:");
    scanf("%f", &angulo);
```

5. Repetições com Variável de Controle e Teste no Início

```
// Converte para radianos
angrad = angulo*PI/180;
printf("Angulo em rad: %f \n", angrad);

// Calcula o cosseno utilizando a função cos de math.h
valcos = cos(angrad);

// Primeiro termo da serie
valserie = 1;

/* Dado que a cada interação o sinal
do termo muda faz-se necessário declarar
uma variável para controlar isso */
sinal = 1;
```

5. Repetições com Variável de Controle e Teste no Início

```
// Inicializa o valor do fatorial com 1
fatorial = 1;

// Calcula um novo termo a cada valor de denominador
i = 2;

// Inicializa termo com o valor para entrar no laço
termo = 1;
while( fabs(termo) >= precisao) {

    /* O cálculo do novo termo depende de um fatorial.
    O fatorial do novo termo é calculado a partir do
    valor do fatorial anteriormente calculado.
    Isso aumenta a eficiência do programa. */

    for (j = i-1; j<= i; j++) fatorial = fatorial*j;
```



5. Repetições com Variável de Controle e Teste no Início

```
// Inverte o sinal
sinal = -sinal;

// Calcula o novo termo
termo = sinal*pow(angrad,i)/fatorial;
printf("termo: %f \n", termo);

// Acrescenta o novo termo ao somatório
valserie = valserie + termo;
printf("serie: %f \n", valserie);

/* Incrementa i para calculo do denominador
   e potência de x */
i +=2;

}
```

5. Repetições com Variável de Controle e Teste no Início

```
// Mostra os resultados na tela do computador
printf("\nFuncao cos(x)      : %f \n", valcos);
printf("Calculado pela serie : %f \n\n", valserie);

return 0;

}
```

“No momento, a Física está mais uma vez em terrível confusão. De qualquer modo, para mim é muito difícil. Gostaria de ter-me tornado um comediante de cinema ou algo do gênero e nunca ter ouvido falar de Física.”

Wolfgang Pauli, nos meses que precederam o artigo de Heisenberg que indicaria o caminho para uma nova teoria dos quanta.

“O tipo de mecânica proposta por Heisenberg devolveu-me a esperança e a alegria de viver. Sem dúvida alguma, ela não proporciona a solução para a charada, mas acredito que agora é possível avançar novamente.”

Wolfgang Pauli, cinco meses depois.