



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Repetição

Enquanto-Repita

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período

Introdução à Programação

Roteiro

- **Estruturas de Repetição**
 - **Variável Contadora**
 - **Variável Acumuladora**
 - **Laços Contados**
 - Para .. Faça
 - **Exercícios de Fixação**
 - **Laços Condicionais**
 - Enquanto .. Faça
 - Repita .. Até que
 - **Exercícios de Fixação**

Introdução à Programação

Laços Condicionais

- **Laços condicionais são aqueles cujo conjunto de comandos em seu interior é executado até que uma determinada condição seja satisfeita.**
- **Ao contrário do que acontece nos laços contados, nos laços condicionais não se sabe de antemão quantas vezes o corpo do laço será executado.**

Introdução à Programação

Laços Condicionais

- As construções que implementam laços condicionais mais comuns nas linguagens de programação modernas são:
 - Enquanto - **laço condicional com teste no início.**
 - Repita - **laço condicional com teste no final.**

Introdução à Programação

Laços Condicionais

- Nos laços condicionais a variável que é testada, tanto no início quanto no final do laço, deve sempre estar associada a um comando que a atualize no interior do laço.
- Caso isso não ocorra, o programa ficará repetindo indefinidamente este laço, gerando uma situação conhecida como “laço infinito”.

Laços Condicionais com Teste no Início

Introdução à Programação

Laços Condicionais com Teste no Início

- Caracteriza-se por uma estrutura que **efetua um teste lógico no início de um laço**, verificando se é permitido ou não executar o conjunto de comandos no interior do laço.
- Os comandos no interior do laço, **devem ser repetidos, enquanto a condição for satisfeita**.

Introdução à Programação

Laços Condicionais com Teste no Início

A sintaxe é mostrada a seguir:

```
ENQUANTO <condição> FAÇA  
    <comando_composto>  
FIMENQUANTO
```

Se seu resultado for falso, então o comando composto no seu interior não é executado e a execução prossegue normalmente pela instrução seguinte à palavra-reservada *fimenquanto* que identifica o final da construção.

Introdução à Programação

Laços Condicionais com Teste no Início

Se a **condição for verdadeira** o comando no interior do laço é executado e ao seu término retorna-se ao teste da condição. Assim, o processo acima será repetido **enquanto a condição testada for verdadeira**.

```
ENQUANTO <condição> FAÇA  
    <comando_composto>  
FIMENQUANTO
```

Neste tipo de laço condicional a variável a ser testada deve possuir um **valor associado antes da construção do laço**.

Laços Condicionais com Teste no Final

Introdução à Programação

Laços Condicionais com Teste no Final

- Caracteriza-se por uma estrutura que **efetua um teste lógico no final de um laço**, verificando se é permitido ou não executar novamente o conjunto de comandos no interior do mesmo.
- Essa estrutura permite que na primeira vez os comandos sejam executados e no final da estrutura é realizado o teste lógico.

Introdução à Programação

Laços Condicionais com Teste no Final

A sintaxe é mostrada a seguir:

REPITA

<comando_composto>

ATÉ QUE *<condição>* ;

A seguir, a condição é testada: se ela for falsa, o comando composto é executado novamente e este processo é repetido até que a condição seja verdadeira, quando então a execução prossegue pelo comando imediatamente seguinte ao final da construção.

Introdução à Programação

Laços Condicionais com Teste no Final

Esta construção difere da construção Enquanto pelo fato do comando composto ser executado uma ou mais vezes (pelo menos uma vez), ao passo que na construção Enquanto o comando composto é executado zero ou mais vezes (possivelmente nenhuma).

Isto acontece porque na construção Repita o teste é feito no final da construção, ao contrário do que acontece na construção Enquanto, onde o teste da condição é efetuado no início da mesma.

Exemplos

Laços Condicionais com

Teste no Inicio

Introdução à Programação

Exemplo 1 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

Algoritmo exemplo_enquanto

Declare soma, num : inteiro;

num \leftarrow 1; soma \leftarrow num;

enquanto (soma < 10) faça

 escreva (soma);

 num \leftarrow num + 1;

 soma \leftarrow soma + num;

fimenquanto

Fim.

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 1 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

```
int main() {  
    int soma, num;  
    num=1; soma = num;  
    while (soma < 10) {  
        printf("%d", soma);  
        num = num + 1;  
        soma = soma + num;  
    }  
    return 0; }
```

Linguagem C

Introdução à Programação

Exemplo 1.1 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

Algoritmo exemplo_enquanto

Declare soma, num : inteiro;

num \leftarrow 1; soma \leftarrow num;

enquanto (soma < 10) faça

 num \leftarrow num + 1;

 soma \leftarrow soma + num;

fimenquanto

escreva (soma);

Fim.

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 1.1 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

```
int main() {  
    int soma, num;  
    num=1; soma = num;  
    while (soma < 10) {  
        num= num + 1;  
        soma= soma + num;  
    }  
    printf ("%d", soma);  
    return 0; }
```

Linguagem C

Introdução à Programação

Exemplo 1.2 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

Algoritmo exemplo_enquanto

Declare soma, num : inteiro;

num \leftarrow 1; soma \leftarrow num;

enquanto (soma < 10) faça

 num \leftarrow num + 1;

 soma \leftarrow soma + num;

 escreva (soma);

fimenquanto

Fim.

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 1.2 - Enquanto

Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva a soma:

```
int main() {  
    int soma, num;  
    num=1; soma = num;  
    while (soma < 10) {  
        num= num + 1;  
        soma= soma + num;  
        printf ("%d", soma);  
    }  
    return 0; }
```

Linguagem C

Introdução à Programação

Exemplo 2 - Enquanto

Escreva um algoritmo que leia os números menores que 10 e informe quantos números existem até que seja maior ou igual ao número 2, escreva a quantidade:

Algoritmo exemplo2_enquanto

Declare quant, num : inteiro;

num \leftarrow 9; quant \leftarrow 0;

enquanto (num < 10) e (num \geq 2) faça

quant \leftarrow quant + 1;

num \leftarrow num - 1;

fimenquanto

escreva(“A quantidade de números é”, quant);

Fim.

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 2 - Enquanto

Escreva um algoritmo que leia os números menores que 10 e informe quantos números existem até que seja maior ou igual ao número 2, escreva a quantidade:

```
int main() {  
    int quant, num;  
    num= 9; quant = 0;  
    while ((num < 10) && (num >= 2)) {  
        quant= quant + 1;  
        num= num - 1;  
    }  
    printf("A quantidade de números é %d", quant);  
    return 0; }  
Linguagem C
```

Introdução à Programação

Exemplo 3 - Enquanto

Escreva um algoritmo que leia os números informados e imprimir o triplo de cada número. O algoritmo acaba quando entrar com o número 999:

Algoritmo exemplo3_enquanto

Declare quant, num : inteiro;

num \leftarrow 9;

escreva("Informe um número ou 999 para terminar");

leia(num);

enquanto (num \neq 999) faça

escreva("O triplo é: ", num * 3);

escreva("Informe um número ou 999 para terminar");

leia(num);

fimenquanto

Fim.

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 3 - Enquanto

Escreva um algoritmo que leia os números informados e imprimir o triplo de cada número. O algoritmo acaba quando entrar com o número 999:

Linguagem C

```
int main() {  
    int quant, num;  
    num= 9;  
    printf("Informe um número ou 999 para terminar");  
    scanf("%d", &num);  
    while (num != 999) {  
        printf("O triplo é: %d", num * 3 );  
        printf("Informe um número ou 999 para terminar");  
        scanf("%d", &num);  
    }  
    return 0;  
}
```

Introdução à Programação

Exemplo 4 - Enquanto

Algoritmo exemplo4_enquanto;

Declare resposta : caracter;

Declare descricao : string;

Declare preco, nv_preco : real;

resposta \leftarrow 'S';

enquanto (resposta = 'S') ou (resposta = 's') faça

 escreva("Informe o produto");

 leia(descricao);

 escreva("Informe o preço");

 leia(preco);

 nv_preco \leftarrow preco * 1.30;

 escreva ("O novo preço de ",descricao," é = ", nv_preco);

 escreva("Deseja continuar S – sim ou N - não");

 leia(resposta);

fimenquanto

Fim.

Dada a descrição de um produto e o preço desenvolver um algoritmo que calcule e mostre o novo preço do produto com um aumento de 30%. Repetir o processo enquanto o usuário desejar:

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 4 - Enquanto

```
int main() {  
    char resposta, descricao[10];  
    float preco, nv_preco;  
    resposta = 'S';  
    while ((resposta == 'S') || (resposta == 's')) {  
        printf("Informe o produto");  
        gets(descricao);  
        printf("Informe o preço");  
        scanf("%f", &preco);  
        nv_preco = preco * 1.30;  
        printf ("O novo preço de %s é = %.2f", descricao, nv_preco);  
        printf("Deseja continuar S – sim ou N - não");  
        scanf("%c", &resposta);  
    }  
    return 0;  
}
```

Dada a descrição de um produto e o preço desenvolver um algoritmo que calcule e mostre o novo preço do produto com um aumento de 30%. Repetir o processo enquanto o usuário desejar:

Linguagem C

Introdução à Programação

Exemplo 5 - Enquanto

```
Algoritmo exemplo5_enquanto;  
  Declare N, CON, ACM : inteiro;  
  Declare MNP : real;  
  CON ← 0; ACM ← 0; N ← 1; MNP ← 0;  
  enquanto (N <> 0) faça  
    escreva("Informe o número");  
    leia(N);  
    se((N mod 2) = 0) então  
      ACM ← ACM + N;  
      CON ← CON + 1;  
    fimse  
  fimenquanto  
  se(CON > 0) então  
    MNP ← ACM / CON;  
    escreva("A média =", MNP );  
  senão  
    escreva("Nenhum número par foi fornecido");  
  fimse  
Fim.
```

Construa um algoritmo que calcule a média aritmética de um conjunto de números pares que forem fornecidos pelo usuário. O valor de finalização será o valor 0. Nada impede o usuário de informar quantos números ímpares quiser com a ressalva de que não poderão ser acumulados:

Vamos
Rastrear !!!

Introdução à Programação

Exemplo 5 - Enquanto

```
int main() {  
    int N, CON, ACM;  
    float MNP;  
    CON=0; ACM=0; N=1; MNP=0;  
    while (N != 0) {  
        printf("Informe o número");  
        scanf("%d", &N);  
        if((N % 2) == 0) {  
            ACM=ACM + N;  
            CON=CON + 1;  
        }  
    }  
    if(CON > 0) {  
        MNP = (float) ACM / CON;  
        printf("A média = %.2f" , MNP );  
    }else{  
        printf("Nenhum número par foi fornecido");  
    }  
    return 0; }
```

Construa um algoritmo que calcule a média aritmética de um conjunto de números pares que forem fornecidos pelo usuário. O valor de finalização será o valor 0. Nada impede o usuário de informar quantos números ímpares quiser com a ressalva de que não poderão ser acumulados:

Linguagem C

Exemplos

Laços Condicionais com

Teste no Final

Introdução à Programação

Exemplo 1 - Repita

Escreva um algoritmo que leia os números maiores que 3 enquanto a sua soma não ultrapasse 20, escreva a soma:

Algoritmo exemplo_repita

Declare soma, num : inteiro;

num \leftarrow 4; soma \leftarrow num;

Repita

 escreva (soma);

 num \leftarrow num + 1;

 soma \leftarrow soma + num;

até que (soma \geq 20);

Fim.

Algoritmo exemplo_repita

Declare soma, num : inteiro;

num \leftarrow 4; soma \leftarrow num;

Faça

 escreva (soma);

 num \leftarrow num + 1;

 soma \leftarrow soma + num;

enquanto (soma $<$ 20);

Fim.

Introdução à Programação

Exemplo 1 - Repita (Linguagem C)

Escreva um algoritmo que leia os números maiores que 3 enquanto a sua soma não ultrapasse 20, escreva a soma:

```
int main() {  
    int soma, num ;  
    num=4; soma = num;  
    do{  
        printf ("%d", soma);  
        num= num + 1;  
        soma= soma + num;  
    }while (soma < 20);  
    return 0;  
}
```

Introdução à Programação

Exemplo 2 - Repita

Escreva um algoritmo que leia números informados pelo usuário e imprimir o quadrado do número até que informe um número par:

Algoritmo exemplo_repita

Declare num : inteiro;

repita

escreva("Informe um número");

leia(num);

escreva("O número é", num, "e o quadrado", num*num);

até que(num % 2 = 0);

Fim.

Introdução à Programação

Exemplo 2 - Repita (Linguagem C)

Escreva um algoritmo que leia números informados pelo usuário e imprimir o quadrado do número até que informe um número par:

```
int main() {  
    int num;  
    do{  
        printf("Informe um número");  
        scanf("%d", &num);  
        printf("O número é %d e o quadrado é %.f", num, pow(num,2));  
    }while(num % 2 != 0);  
    return 0;  
}
```

Introdução à Programação

Exemplo 3 - Repita

Faça um algoritmo que peça a leitura de um valor para a variável num, multiplicar este valor por 3, colocando o resultado em uma variável R e apresentar o valor. Tudo isso deverá ser repetido por 5 vezes:

Algoritmo exemplo_repita

Declare num, r, cont : inteiro;

cont \leftarrow 1;

repita

 escreva("Informe um número");

 leia(num);

 r \leftarrow num * 3;

 escreva("O número é", num, " – o seu triplo é", r);

 cont \leftarrow cont + 1;

até que(cont > 5);

Fim.

Introdução à Programação

Exemplo 3 - Repita (Linguagem C)

Faça um algoritmo que peça a leitura de um valor para a variável num, multiplicar este valor por 3, colocando o resultado em uma variável R e apresentar o valor. Tudo isso deverá ser repetido por 5 vezes:

```
int main() {  
    int num, r, cont;  
    cont = 1;  
    do{  
        printf("Informe um número");  
        scanf("%d", &num);  
        r = num * 3;  
        printf("O número é %d – o seu triplo é %d", num, r);  
        cont ++;  
    }while(cont <= 5);  
    return 0; }
```


Introdução à Programação

Exemplo 4 - Repita

Faça um algoritmo que leia os números inteiros não determinados e escreva seu cubo até que o usuário deseje sair:

Algoritmo exemplo_repita

Declare num, cubo : inteiro;

Declare resp : caracter;

repita

escreva("Informe um número");

leia(num);

cubo \leftarrow num ** 3;

escreva("O número é", num, " – o seu cubo é", cubo);

escreva("Deseja continuar S – sim e N - não");

leia(resp);

até que(resp \neq 'S') e (resp \neq 's') ;

Fim.

Introdução à Programação

Exemplo 4 - Repita (Linguagem C)

Faça um algoritmo que leia os números inteiros não determinados e escreva seu cubo até que o usuário deseje sair:

```
int main() {  
    int num ;  
    float cubo;  
    char resp;  
    do{  
        printf("Informe um número");  
        scanf("%d", &num);  
        cubo = pow(num , 3);  
        printf("O número é %d – o seu cubo é %.f", num, cubo);  
        printf("\nDeseja continuar S – sim e N - não");  
        scanf("%c", &resp);  
    }while( resp == 'S' || resp == 's' ) ;  
    return 0; }
```

Introdução à Programação

Exemplo 5 - Repita

Algoritmo exemplo_repita;
Declare resposta : character;
Declare descricao : string;
Declare preco, novo_preco : real;
Repita

 escreva("Informe o produto");

 leia(descricao);

 escreva("Informe o preço");

 leia(preco);

 novo_preco \leftarrow preco * 1.30;

 escreva ("O novo preço de ",descricao, "é =", novo_preco);

 escreva("Deseja continuar S – sim ou N - não");

 leia(resposta);

até que(resposta = 'N') ou (resposta = 'n');

Fim.

Dada a descrição de um produto e o preço desenvolver um algoritmo que calcule e mostre o novo preço do produto com um aumento de 30%. Repetir o processo enquanto o usuário desejar:

Introdução à Programação

Exemplo 5 - Repita (Linguagem C)

```
int main() {  
    char resposta, descricao[10];  
    float preco, nv_preco;  
    do{  
        printf("Informe o produto");  
        gets(descricao);  
        printf("Informe o preço");  
        scanf("%f", &preco);  
        nv_preco = preco * 1.30;  
        printf ("O novo preço de %s é = %.2f", descricao, nv_preco);  
        printf("Deseja continuar S – sim ou N - não");  
        scanf("%c", &resposta);  
    } while ((resposta == 'S') || (resposta == 's'));  
    return 0;  
}
```

Dada a descrição de um produto e o preço desenvolver um algoritmo que calcule e mostre o novo preço do produto com um aumento de 30%. Repetir o processo enquanto o usuário desejar:

Introdução à Programação

Exercício de Fixação (Enquanto e Repita)

- P1 – Escreva um algoritmo que leia os números maiores que 0 enquanto a sua soma não ultrapasse 10, escreva o sucessor do número:**
- P2 – Escreva um algoritmo que leia os números informados e imprimir o dobro de cada número. O algoritmo acaba quando entrar com o número -99:**
- P3 – Faça um algoritmo para entrar com números enquanto forem positivos e imprimir quantos números foram informados:**
- P4 – Faça um algoritmo para entrar com números positivos e imprimir a média dos números informados:**

Introdução à Programação

Exercício de Fixação (Enquanto e Repita)

- P5 – Escreva um algoritmo que leia os números menores que 20 e imprimir a soma somente dos números pares:**
- P6 – Escreva um algoritmo que leia números inteiros e imprimir seu dobro enquanto o número for diferente de 0:**
- P7 – Escreva um algoritmo que leia os números menores que 30, e imprimir a soma dos números divisíveis por 3, enquanto o número for diferente de 0:**
- P8 – Escreva um algoritmo que leia os números menores que 50 e imprimir o quadrado somente dos números ímpares:**

Introdução à Programação

Exercício de Fixação (Enquanto e Repita)

P9 – Escreva um algoritmo que leia dez números positivos e escreva o menor dos números:

P10 – Escreva um algoritmo que leia dez números positivos e escreva o maior dos números::

P11 – Faça um algoritmo para entrar com três notas de 10 alunos, calcule a média de cada aluno e imprimir quantos alunos foram aprovados (media > 7.0):

P12 – Faça um algoritmo para imprimir a sequência da série de Fibonacci até o seu décimo quinto termo:

Série = 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 ...



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Repetição

Enquanto-Repita

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período