

INSTITUTO FEDERAL
BAIANO

Algoritmos e Introdução à Programação

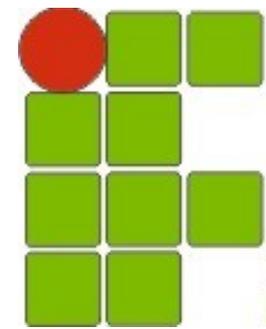
Lógica e Linguagem de Programação

Prof. José Honorato Ferreira Nunes

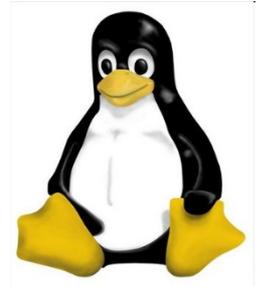
honoratonunes@softwarelivre.org

<http://softwarelivre.org/zenorato/honoratonunes>





INSTITUTO FEDERAL
BAIANO



Linguagem C – Seleção e Repetição

Prof. José Honorato Ferreira Nunes

honoratonunes@softwarelivre.org

<http://softwarelivre.org/zenorato/honoratonunes>

Resumo da aula

- Estruturas de Seleção
 - Simples
 - Composta
 - Aninhada
- Estruturas de Repetição
 - Pré-Teste (while)
 - Pós-Teste (do while)
 - Variável de Controle (for)

Estruturas de Controle: Seleção Simples

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

```
int main ()
```

```
{
```

```
    int idadeAluno;
```

```
    printf("Informe a idade do aluno: ");
```

```
    scanf("%d",&idadeAluno);
```

```
    if (idadeAluno < 18) {
```

```
        printf("O aluno é menor de idade.");
```

```
    }
```

```
    printf("A idade do aluno é: %d",idadeAluno);
```

```
    return 0;
```

```
}
```

Estruturas de Controle: Seleção Composta

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

```
int main ()
```

```
{
```

```
    int idadeAluno;
```

```
    printf("Informe a idade do aluno: ");
```

```
    scanf("%d",&idadeAluno);
```

```
    if (idadeAluno < 18) {
```

```
        printf("O aluno é menor de idade.");
```

```
    }
```

```
    else {
```

```
        printf("O aluno é maior de idade.");
```

```
    }
```

```
    printf("A idade do aluno é: %d",idadeAluno);
```

```
    return 0;
```

```
}
```

Atividade

Crie um algoritmo em C para calcular a média final do aluno. O sistema deve mostrar a matrícula do aluno e receber duas notas (trabalho e prova). Depois de calcular a média final, se o aluno tiver média ≥ 7 informar que ele foi aprovado, senão informar que ele foi reprovado.

Estruturas de Controle: Seleção Aninhada

```
#include <stdio.h>
int main ()
{
    float media;
    char nome[30];
    printf("Informe o nome do aluno: ");
    scanf("%s",&nome);
    printf("Informe a média do aluno: ");
    scanf("%f",&media);
    if (media >= 7) {
        printf("O aluno %s está %s",nome,"aprovado");
    }
    else if (media >= 3) {
        printf("O aluno %s está %s",nome,"na final");
    }
    else {
        printf("O aluno %s está %s",nome,"reprovado");
    }
    return 0;
}
```

Atividades

- Faça um algoritmo que receba um número e mostre uma mensagem caso este número seja maior que 10.
- Escrever um algoritmo que leia dois valores inteiro distintos e informe qual é o maior.
- Faça um algoritmo que receba um número e diga se este número está no intervalo entre 100 e 200.

Atividades

- Faça um algoritmo que receba três valores inteiros e organize esses valores em ordem decrescente.
- Fazer um algoritmo que solicite o nome, idade e sexo do aluno e caso ele seja homem e maior de idade solicitar ao o número da carteira de reservista, caso seja homem menor de idade emitir a mensagem: "aguardando idade para o exercício militar", caso seja mulher emitir a mensagem: "dispensada do exercício militar".

Atividades

- Escrever um algoritmo que leia o nome e as três notas obtidas por um aluno durante o semestre. Calcular a sua média (aritmética), informar o nome e sua menção aprovado (média ≥ 7), Reprovado (média ≤ 5) e Recuperação (média entre 5.1 a 6.9).

Atividades

- Crie um algoritmo, utilizando pseudocódigo, que leia dois números informados pelo usuário e, em seguida, exiba na tela uma mensagem informando se o maior deles é o primeiro, o segundo, ou se são iguais.
- Crie um algoritmo, utilizando pseudocódigo, que:
Leia três números informados pelo usuário;
Multiplique o menor valor lido pelo maior valor e some o resultado com o valor do meio;
Exiba na tela o resultado da soma.

Atividade

Crie um algoritmo em pseudocódigo para aplicar um percentual de desconto sobre o valor de uma compra informado pelo usuário. Os percentuais de desconto são:

15% para compras acima de R\$ 500,00;

10% para compras entre R\$ 200,00 e R\$ 499,99;

5% para compras abaixo de R\$ 200,00.

Mostre na tela uma mensagem informando: valor antes do desconto, valor do desconto e valor a ser pago.

Estruturas de Controle

Repetição

Pré-Teste (while)

Pós-Teste (do while)

Variável de controle (for)

Estruturas de Controle - Repetição

Nos exemplos e exercícios que vimos até agora, sempre foi possível resolver os problemas com uma sequência de instruções que eram executadas apenas uma vez. Existem três estruturas básicas para a construção de algoritmos, que são:

- algoritmos sequenciais,
- algoritmos com seleção;
- e algoritmos com repetição.

Estruturas de Controle - Repetição

A combinação dessas três estruturas permite-nos a construção de algoritmos para a resolução de problemas extremamente complexos. Nesta aula veremos as estruturas de repetição possíveis em algoritmos e existentes na maioria das Linguagens de Programação.

Estruturas de Controle - Repetição

Uma estrutura de repetição permite que uma sequência de instruções (comandos) seja executada várias vezes, até que uma condição (teste) seja satisfeita, ou seja, repete-se um conjunto de instruções sem que seja necessário escrevê-las várias vezes. As estruturas de repetição também são chamadas de Laços ou Loops.

Estruturas de Controle - Repetição

Para sabermos quando utilizar uma estrutura de repetição, basta analisarmos se uma instrução ou uma sequência de instruções precisa ser executada várias vezes, se isto se confirmar, então deve-se utilizar uma estrutura de repetição. As estruturas de repetição, assim como a de decisão (seleção), envolvem a avaliação de uma condição (teste).

Repetição com pré-teste: while

Sintaxe:

```
while(condição) comando;
```

Uma maneira possível de executar um laço é utilizando o comando while. Ele permite que o código fique sendo executado numa mesma parte do programa de acordo com uma determinada condição.

- o comando pode ser vazio, simples ou bloco
- ele é executado desde que a condição seja verdadeira
- testa a condição antes de executar o laço

Repetição com Pré-Teste : while

A repetição com Pré-Teste é uma estrutura de looping que repete um bloco de comandos enquanto a expressão avaliada for verdadeira.

No C, esse tipo de repetição é representado pelo comando **while**, que utiliza a seguinte estrutura:

```
while (condição) {comandos;}
```

Repetição com Pré-Teste : while

Ex:

```
#include <stdio.h>
main()
{
    char ch;
    while(ch!='a') {
        ch=getchar();
    }
}
```

Atividades: utilizar estrutura de repetição com Pré-Teste (while)

- Crie um algoritmo que leia um valor inteiro para X e escreva na tela X^3 . O algoritmo deve continuar pedindo o valor de X até que o usuário informe 0 (zero), então o programa encerra.
- Desenvolva um algoritmo capaz de apresentar na tela o fatorial de um número inteiro informado pelo usuário.

Repetição com Pós-Teste : do while

A repetição com Pós-Teste é similar a repetição com Pré-Teste, porém com duas diferenças cruciais:

- Na repetição com Pós-Teste, o teste que define a execução ou não do bloco de comandos é realizada no final do bloco.
- Ao contrario do Pré-Teste, no Pós-Teste o bloco de comando é executado enquanto o teste for falso.

```
do{  
    Comandos;  
}  
While (condição);
```

Repetição com variável de controle: for

Sintaxe:

```
for(inicialização;condição;incremento) comando;
```

O comando for é de alguma maneira encontrado em todas linguagens procedurais de programação.

Em sua forma mais simples, a inicialização é um comando de atribuição que o compilador usa para estabelecer a variável de controle do loop. A condição é uma expressão de relação que testa a variável de controle do loop contra algum valor para determinar quando o loop terminará. O incremento define a maneira como a variável de controle do loop será alterada cada vez que o computador repetir o loop.

Repetição com variável de controle: for

Ex 1:

```
main()
```

```
{
```

```
    int x;
```

```
    for(x=1;x<100;x++)printf("%d\n",x);
```

```
}
```

Ex2:

```
main()
```

```
{
```

```
    int x,y;
```

```
    for (x=0,y=0;x+y<100;++x,++y)
```

```
        printf("%d ",x+y);
```

```
}
```

Repetição com variável de controle: for

Ex 3:

main()

```
{
    int linha,coluna;
    for(linha=1;linha<=24;linha++)
    {
        for(coluna=1;coluna<40;coluna++)    printf("-");
        putchar('\n');
    }
}
```

Atividades: utilizar estrutura repetição variável de controle (for)

- Crie um algoritmo que realize as seguintes atividades:
 - a) Pergunte a quantidade de alunos da turma.
 - b) Solicite ao usuário o nome de cada um dos X alunos.
 - c) Envie cada nome lido para impressora.

Bibliografia

- ❑ MANZANO, Wilson Y. Yamaturni-São Paulo-SP. **Lógica estruturada para programação de computadores**, Ed. Érica 1997 e 2001.
- ❑ MORAES, Celso Roberto. **Estruturas de Dados e Algoritmos**. Ed. Érica, São Paulo
- ❑ LOPES, Anita. **Introdução à programação**. Rio de Janeiro: Campus, 2002.
- ❑ BENEDUZZI, Humberto M. e METZ, João A. **Lógica e Linguagem de Programação – Introdução ao Desenvolvimento de Software (1ª edição)**. Editora do Livro Técnico, 2010.
- ❑ SEBESTA, Robert W. **Conceitos de linguagens de programação**. 9. ed. Porto Alegre: Bookman, 2003.
- ❑ CORMEN, Thomas H. **Algoritmos: teoria e prática**. Rio de Janeiro: Campus, 2002.
- ❑ ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal e C/C++ e Java**. 2. ed. São Paulo: Pearson Prentice Hall, 2008.
- ❑ ZIVIANI, Nivio. **Projeto de algoritmos com implementações em Pascal e C**. São Paulo: Pioneira Thomson Learning.