

Algoritmo

Linguagem C

A idéia desse material tem como objetivo uma tradução de um algoritmo para a Linguagem C ANSI

Nessa fase iremos apenas traduzir sem utilizar compilador pois nessa fase iremos apenas montar a base de um programa que será inserido em um ambiente com um compilador para que possa posteriormente ser executado.

Como ainda iremos conhecer detalhes da estrutura e definições de comando poderemos adiantar e em muito o aprendizado de linguagem de programação.

Poderemos ter outros “DE-PARA” para outras linguagens que por ventura se queira utilizar esse método se mostra muito eficaz no processo e tempo para uso de linguagem.

Durante as próximas aulas iremos abordar todos os itens básicos dessa linguagem

Universidade Federal Tecnológica do Paraná - Departamento de Informática

Prof Etéocles da Silva Cavalcanti

eteocles@dainf.ct.utfpr.edu.br

Nome de variáveis, funções e programas em C

Identificadores para nome de variáveis

- 1-Sempre começa com uma letra a..z ou A..Z**
- 2-Não pode conter símbolo especial exceto _**
- 3- Não mais de 32 caracteres**
- 4-Não pode ser uma palavra reservada da linguagem**

Obs : Após o primeiro caracter é possível usar caracteres numéricos.

A partir da tradução para linguagem deverá ser considerado que uma variável deverá ter sua característica definida. Ou será definida como uma variável inteira (int) ou não inteira (float).

Ler x **scanf("%?", &x);**
 %d inteiro
 %f float

Ler x **printf("Entre com X ->");scanf("%f", &x);**

Ler x **printf("Entre com X ->");**
 scanf("%f", &x);

Ler a, b **printf("Entre com a ->");scanf("%f", &a);**
 printf("Entre com b ->");scanf("%f", &b);

Mostre x

```
printf("\n x= %?",x);
```

%d inteiro
%f float

Mostre a,b

```
printf("\n a= %f    b=%f ",a,b);
```

Mostre x,f(x)

```
printf("\n x=%f    f(x)=%f",x,fx);
```

Mostre R,Vol

```
printf("\n Raio=%f  Volume=%f",x,fx);
```

Algoritmo

Linguagem C

| | | |
|--------------------------------------|--------------------------------------|-------|
| $x \leftarrow 0$ | <code>x=0;</code> | int |
| $x \leftarrow 0$ | <code>x=0.;</code> | float |
| $x \leftarrow x + 1$ | <code>x=x+1;</code> | int |
| $x \leftarrow x + 1$ | <code>x=x+1.;</code> | float |
| $f[x] \leftarrow x + 3$ | <code>fx=x+3.;</code> | float |
| $z \leftarrow \frac{\sqrt{x}}{3}$ | <code>z=sqrt(x)/3.;</code> | float |
| $V \leftarrow 4/3 * \pi * r^3$ | <code>Vol=4./3.*3.1416*r*r*r;</code> | float |
| $\alpha \leftarrow -\beta + 3\gamma$ | <code>alfa=beta+3.*gamma;</code> | float |
| $R \leftarrow n - (n/2) * 2$ int | <code>R=n-(n/2)*2;</code> | |

Atenção

Não misturar tipos diferentes nas expressões e atenção com a atribuição. A variável que recebe o valor de uma expressão deve ser do mesmo tipo da expressão.

Observe a ordem da hierarquia de cálculos e evite uso demasiado de parênteses.

Algoritmo

Linguagem C

| | |
|---------------|---|
| Adição | + |
| Subtração | - |
| Divisão | / |
| Multiplicação | * |

| | |
|----------------|----|
| Menor que | < |
| Maior que | > |
| Menor ou igual | <= |
| Maior ou igual | >= |
| Igualdade | == |
| Desigualdade | != |

| Tipo | Operador |
|----------------------|-----------------|
| Expressão | () [] |
| Unário | - ~ ! * & ++ -- |
| Multiplicativo | * / % |
| Aditivo | + - |
| Deslocamento | << >> |
| Relacional | < <= > >= |
| Relacional | == != |
| And bit | & |
| Xor bit | ^ |
| Or bit | |
| And lógico | && |
| Or lógico | |
| Condicional | ? : |
| Atribuição | = *= /= ... |
| Avaliação Sequencial | , |

| | |
|----------------|----|
| e (conjunção) | && |
| ou (disjunção) | |
| não (negação) | ! |

if (condição)
Seqüência de Comandos 1

Se $x < b$ então

X<- 0

Fim-se

if (condição)
Seqüência de Comandos 1
else
Seqüência de Comandos 2

if(x<b)

x=0;

Se $x < b$ ou $c=0$ então

X<- 0

Y<-9

Fim-se

if(x<b | c==0)

{x=0;

y=9;

}

* {bloco} necessário mais de um comando

Algoritmo

Linguagem C

Se $x < b$ e $c=0$ então

X← 0

Y←-9

Senao

X←-0

y← - 89

fim-se

if($x < b$ && $c==0$)

{x=0;

y=9;

}

else

{x=0;

y=89;

}

Se $x < b$ e $c=0$ então

X← 0

Senao

X←-0

y← - 89

fim-se

if($x < b$ && $c==0$)

x=0;

else

{x=0;

y=89;

}

Especiais**Se mês = 1 entao****Se dia = 31 entao****dia<-dia+1****Fim-se****Fim-se****if(mes==1)****if(dia==31)****dia=dia+1;****Se 1<=x<=10 entao****Dia<-dia+1****Fim-se****if(x>=1 && x<=10)****dia=dia+1;**

```
do {  
    Seqüência de Comandos  
    } while (condição);  
  
while (condição)  
    {  
        Seqüência de Comandos  
    }  
  
for (inicialização ; condição ; incremento)  
    {  
        Sequência de comandos  
    }
```

(*) Como SEMPRE iremos escrever um algoritmo para depois realizar sua conversão para C , a tradução será feita de forma natural e usando as definições das construções dos algoritmo.

Algoritmo

Linguagem C

Repita

```
x<-x+1
mostre x
ler a
at  x<=20
```

do{

```
x=x+1;
printf("\n x= %d",x);
scanf("%d",&a);
}while( x <=20 );
```

enquanto x<=20 fa a

```
x<-x+1
mostre x
ler a
fim-enq
```

while(x <= 20.)

```
{x=x+1.;
printf("\n x= %f",x);
scanf("%d",&a);
}
```

para x de 1 ate 20 fa a

```
mostre x
a<- a + x
fim-para
```

for(x=1;x<=20;x++)

```
{printf("\n x= %d",x);
a=a+x;
}
```

Algoritmo

Linguagem C

enquanto x<=20 faça

se x = 0

x<-x+1

fim-se

mostre x

ler a

fim-enq

while(x <= 20)

{if(x==0)

x=x+1;

printf("\n x= %d",x);

scanf("%d",&a);

}

para x de 1 ate 20 faça

mostre x

fim-para

for(x=1;x<=n;x++)

{printf("\n x= %d",x);

}

(*) Inicialmente iremos utilizar BLOCO em todos os comandos de repetição mesmo para um único comando não será considerado como erro.