Visualg

Em visualg, valem as regras:
. cada comando de visualg deve estar em uma linha
. não há separadores (;), nem blocos (como { e {) nem goto
. não há funções como sqr(), sqrt()
. todas as palavras reservadas não tem acentos ou cedilha
. não há separação entre maiúsculas e minúsculas nos comandos ou variáveis
. o limite de variáveis é de 500 (cada item de vetor conta como 1 variável)
. o ponto decimal é o ponto
. valores caractere estão entre aspas duplas
. O símbolo de atribuição é <-
. Existe recursividade

Algoritmo É a unidade básica de escrita de algoritmos.

Formato	Exemplo
algoritmo <nome> | var | <variáveis> --->local da definição de função | inicio | <comandos> fimalgoritmo	algoritmo TESTE | var | N : inteiro | inicio | leia (N) | escreva (N × 2) fimalgoritmo

Comentários são caracterizados por começar por \\.

Função A função é uma unidade não autônoma de programação. Ela sempre precisa receber seus dados (através de uma lista de parâmetros) e devolver um resultado a quem a chamou. Em Visualg são colocadas imediatamente antes da palavra `inicio` do bloco a que se referem.

Formato	Exemplo
funcao <nome>(<parametros>) : <tipo-res> | var | <variaveis locais> | inicio | <comandos> fimfuncao	função TESTE (N:inteiro) : inteiro var inicio | retorne N × 2 fimfunção

O <nome-de-função> obedece as mesmas regras de nomenclatura das variáveis. Por outro lado, a <seqüência-de-declarações-de-parâmetros> é uma seqüência de
[var] <seqüência-de-parâmetros>: <tipo-de-dado>
separadas por ponto e vírgula. A presença (opcional) da palavra-chave var indica passagem de parâmetros por referência; caso contrário, a passagem será por valor.

Nomes Ao construir algoritmos é necessário dar nomes a muitas coisas. A regra de construção de nomes é
. Uma única palavra
. Composta de letras e números
. Começando com uma letra
. Escrita em maiúsculo

Tipos Os tipos possíveis são 4: inteiro, real, lógico, caracter.

Tipo	Conteúdo	Exemplo
inteiro	qualquer número inteiro, variando entre $-\infty$ e $+\infty$	inteiro A A,B,C : inteiro
real	qualquer número inteiro ou não inteiro	A : real X,Y,Z : real
caracter	uma cadeia	A : caracter // o tamanho não é explícito
logico	somente pode conter os valores VERDADEIRO e FALSO	A : logico

Vetor Um vetor (ou matriz) é definido, escrevendo-se o nome, dois pontos (:) a palavra `vetor` e as especificações de repetição. Veja nos exemplos:

Formato	Exemplo
<nome> : vetor [inicio..fim] [inicio..fim] ... de <tipo>	AAA : vetor [1..10] de real BBB : vetor [0..4] [1..8] de inteiro

Comandos Os comandos em pseudo-código são: $<-$ (recebe), leia, escreva, se...fimse, enquanto...fimenquanto, para...fimpara, repita...até, retorne e abandone.

Leia Serve para introduzir um dado externo para dentro do algoritmo. As variáveis citadas no comando já terão que ter sido definidas e tipadas.

Formato	Exemplo
leia (<lista de variáveis>)	leia (A)

Escreva Serve para produzir um dado como resposta do algoritmo. As variáveis citadas no comando já terão que ter sido devidamente atribuídas.

Formato	Exemplo
escreva (<lista de variáveis>)	escreva (A, B, C)

Para variáveis numéricas, pode-se especificar var:tamanho:decimais.

$<-$ Este comando permite inicializar uma variável ou alterar-lhe o conteúdo.

Formato	Exemplo
<variável> $<-$ <variável> OU	A $<-$ B
<variável> $<-$ <constante> OU	B $<-$ "CURITIBA"
<variável> $<-$ <expressão> OU	C $<-$ A + 23

Expressão Numérica Qualquer combinação compatível de funções que acabem gerando um único resultado numérico.

Formato	O que faz	Exemplo
adição (+), subtração (−), multiplicação (*), divisão real (/)	o usual da aritmética	A+3 A/2.5
divisão inteira (barra ao contrário)	não usa números decimais	
resto (percentagem)	o resto da divisão inteira	
abs(a:real):real	o valor absoluto de a	
arcxxxx(a:real):real	arco xxx	xxx pode ser cos, sen e tan
asc(s:caracter):inteiro	converte o caracter s em inteiro	
carac(c:inteiro):caracter	converte o byte c em caracter	
caracpnum(c:caracter):inteiro ou real		
compr(c:caracter):inteiro	deve ser o comprimento do string	
copia(c:caracter; p,n:inteiro):caracter	copia um substring	
xxx(a:real):real	funcao trigonometrica	xxx pode ser sen,cos,tan,cotan
a div b	divisao inteira de a po b	
exp(b,e)	eleva a base ao expoente	
grauprad(a:real):real	converte graus em radianos	
int(a:real):inteiro	converte real em inteiro	
log(a:real):real	logaritmo base 10	
logn(a:real):real	logaritmo base e	
maiusc(c:caracter):caracter	converte para maiuscula	
minusc(c:caracter):caracter	converte para minuscula	
a mod b	resto da divisao inteira de a por b	
numpcarac(n:inteiro ou real):caracter	converte numerica a caracter	
pi:real	devolve o valor de pi	
pos(subc,c:caracter):inteiro	devolve a posicao de subc em c	
quad(a:real):real	devolve o quadrado de a	
radpgrau(a:real):real	converte radiano para grau	
raizq(a:real):real	devolve a raiz quadrada	

Expressão Relacional Qualquer combinação compatível de relações que acabem gerando um único resultado lógico.

Formato	O que faz	Exemplo
Igual (=)	devolve VERDADEIRO se seus operandos são iguais e FALSO senão	3=3 é VERDADEIRO
Diferente (<>)	devolve VERDADEIRO se seus operandos são diferentes e FALSO senão	3<>3 é FALSO
Maior (>)	Em $A > B$ devolve VERDADEIRO se A é maior do que B	3>4 é FALSO
Menor (>)	Em $A < B$ devolve VERDADEIRO se A é menor do que B	3<4 é VERDADEIRO
Maior ou igual (>=)	Em $A >= B$ devolve VERDADEIRO se A é maior ou igual do que B	3>=3 é VERDADEIRO
Menor ou igual (<=)	Em $A <= B$ devolve VERDADEIRO se A é menor ou igual do que B	3<=3 é VERDADEIRO

Expressão Lógica Qualquer combinação compatível de expressões relacionais e/ou lógicas que acabem gerando um único resultado lógico.

Formato	O que faz	Exemplo
E	Em A E B, devolve VERDADEIRO se A e B são verdadeiros e devolve FALSO senão	VERDADEIRO E VERDADEIRO é VERDADEIRO Todas as outras combinações dão FALSO
Ou	Em A OU B, devolve VERDADEIRO se A ou B ou ambos são VERDADEIRO e devolve FALSO senão	FALSO OU FALSO é FALSO Todas as outras combinações dão VERDADEIRO
Não (NAO)	Inverte o valor lógico do operando	NAO VERDADEIRO é FALSO e NAO FALSO é VERDADEIRO

Se Este comando é denominado alternativo, pois permite escolher caminhos da programação dependendo de uma condição (expressão lógica). Note que o trecho entre `senão` e o comando imediatamente anterior a `fimse` são opcionais, razão pela qual no formato eles aparecem entre colchetes. Os comandos entre `então` e `senão` ou `fimse` (se não houver senão) serão executados apenas se a condição do comando for verdadeira. Se houver comandos entre `senão` e `fimse` os mesmos serão executados apenas se a condição for falsa.

Formato	Exemplo
se (<condição>) então | <comando-1> | <comando-2> | <...> [senao | <comando-1> | <comando-2> | <...>] fimse	se (A $\geq$ 4) então | B $\leftarrow$ B + 1 | C $\leftarrow$ C + D + A senao | escreva (N × 2) fimse

Enquanto Este comando permite a realização de laços (loops) dentro de programas. Começando o comando, a condição é avaliada. Se ela for falsa, há um desvio para o comando seguinte ao `fimenquanto`. Se a condição for verdadeira os comando internos ao `enquanto` são executados. Ao se encontrar o `fimenquanto` há um desvio incondicional ao início do `enquanto` e a condição inicial é reavaliada.

Formato	Exemplo
enquanto (<condição>) faça | <comando-1> | <comando-2> | <...> fimenquanto	A $\leftarrow$ 5 enquanto (A $\leq$ 9) então | escreva (A) | A $\leftarrow$ A + 3 fimenquanto Serão impressos os valores 5 e 8.

Repita Este comando também permite a realização de laços (loops) dentro de programas.

Formato	Exemplo
repita | <comando-1> | <comando-2> | <...> ate (<condição>)	A $\leftarrow$ 5 repita | escreva (A) | A $\leftarrow$ A + 3 ate (A > 9) Serão impressos os valores 5 e 8.

Para Este comando também permite a realização de laços (loops) dentro de programas. No início a variável citada no comando é inicializada com <constante1>. Depois é feita a condição. Se o passo está

ausente ou é positivo, a variavel é testada para $\leq$ <constante2>. (Se o passo é negativo, o teste é com $\geq$). Se o resultado é VERDADEIRO os comandos internos são executados. Ao final deles, a variável é incrementada (ou decrementada se o passo é negativo) e depois há um retorno ao teste inicial. Quando o passo não é explicito, ele vale 1.

Formato	Exemplo
para <var> de <constante1> ate <constante2> [passo <constante3>] faça | <comando-1> | <comando-2> | <...> fimpara	para K de 3 ate 8 passo 2 faça | escreva (A) fimpara Serão impressos os valores 3, 5 e 7.

Retorne Usado exclusivamente dentro de funções, tem a finalidade de devolver um resultado a quem chamou esta função. Ao contrário do "C", não necessariamente encerra a execução da função.

Formato	Exemplo
retorne <expressão compatível>	se A > 5 então | retorne A fim{se}

Outros Comandos aleatorio

arquivo <nome-de-arquivo>
algoritmo "lendo do arquivo"arquivo "teste.txt"
timer on / timer off
pausa
debug
eco
cronômetro

Exemplo 1

```
algoritmo "primos"
var
J,K,R: inteiro

funcao QP(N: inteiro): inteiro
var
A,B:inteiro
inicio
A <- 2
B <- 1
enquanto B <= N faca
   se EPRIMO(A) entao
      B <- B + 1
   fimse
   A <- A + 1
fimenquanto
retorne A - 1
fimfuncao

funcao EPRIMO(M: inteiro): logico
var
QT,DI:inteiro
inicio
QT <- 0
DI <- 2
enquanto DI < M faca
   se M % DI = 0 entao
      QT<- QT + 1
   fimse
   DI <- DI + 1
fimenquanto
retorne QT = 0
fimfuncao

inicio
leia (J,K)
R <- QP(K)-QP(J)
escreva (R)
escreva (QP(K))
escreva (QP(J))
fimalgoritmo
```

Exemplo 2

```
algoritmo "palito"
var
N,SEQ,J,TV:inteiro
inicio
N<-0
enquanto ((N<20) ou (N>30)) faca
   escreval ("com quanto comecamos ?")
   leia (N)
fimenquanto
enquanto (N>0) faca
   SEQ <- 1
   enquanto ((N-SEQ)>=0) faca
      SEQ <- SEQ + 4
 //     escreval ("depuracao... N=",N,"  SEQ=",SEQ)
   fimenquanto
   J<-4+(N-SEQ)
   se (J=0) entao
      J<-1
   fimse
   escreval ("eu joguei ",J)
   N<-N-J
   escreval ("Existem ",N," palitos")
   se (N<=0) entao
      escreva ("por incrivel que pareca,... perdi,... babaca")
      interrompa
   fimse
   TV<-0
   enquanto ((TV<1) ou (TV>3)) faca
      escreval ("jogue")
      leia (TV)
   fimenquanto
   N <- N - TV
   escreval("existem ",N," palitos")
   se (N<=0) entao
      escreval("burrao, eu ganhei...")
      interrompa
   fimse
fimenquanto
fimalgoritmo
```

Exemplo 3

```
algoritmo "raiz quadrada"
var
a:real
 funcao sqrt(N:real):real
   retorne teste(N,1)
fimfuncao
funcao teste(X,G:real):real
inicio
   se perto((X/G),G) entao
      retorne G
   senao
      retorne teste(X, melhor(X,G))
   fimse
fimfuncao
funcao melhor(X,G:real):real
inicio
   retorne (G+(X/G))/2  //a media entre G e X/G
fimfuncao

funcao perto(A,B:real):logico
inicio
   retorne (B*0.001) > abs(A-B)
fimfuncao
inicio
leia(a)
escreval(a, sqrt(a))
fimalgoritmo
```