

Tabela de conversão de Pascal para C

ESTRUTURA BÁSICA DE UM PROGRAMA

Pascal	C	C (outra opção)
<pre> program uses ... begin ... seu código ... end.</pre>	<pre> /* comentario */ #include ... int main() { ...seu código... return (0); }</pre>	<pre> /* comentario */ #include ... void main() { ...seu código... }</pre>

USO DE BIBLIOTECAS

Pascal	Chamada	C
<code>uses CRT;</code>	biblioteca padrão	<code>#include < stdio.h ></code>
<code>interface;</code>	biblioteca do usuário no subdiretório de trabalho	<code>#include "agua.i"</code>

FUNÇÕES DAS BIBLIOTECAS MAIS USADAS

Nome	Uso geral	Funções mais usadas
<code>stdio.h</code>	standard input output entrada e saída padrão	<code>printf(), scanf(), ...</code> <code>gets(), puts(), ...</code> <code>atoi(), atof(), atol(), ...</code> comandos para uso de arquivos.
<code>conio.h</code>	console input output uso de teclado e monitor	<code>gotoxy(x, y), clrscr(), cbreak(), getch(), ...</code>
<code>math.h</code>	mathematics operações matemáticas	<code>pow(x , y), exp(), sqrt(),</code> <code>log(), log10(),</code> <code>sin() , asin(), sinh(), ...</code> <code>ceil(), floor(),</code> <code>fabs(), abs(), labs(),</code> <code>modf(), fmod(),</code> <code>poly(x, grau, coeficientes), hypot(x,y), ...</code>
<code>ctype.h</code>	character type manuseio de caracteres	<code>toupper(), tolower(), ...</code>
<code>string.h</code>	strings manuseio de strings	<code>strcmp(palavra1, palavra2),</code> <code>strncmp(palavra1, palavra2), ...</code>
<code>dos.h</code>	disk operating system recursos do sistema operacional	<code>sound(); delay(); nosound(); ...</code>
<code>stdlib.h</code>	standard library diversas utilidades	<code>strtod(), strtol(),</code> <code>abort(), execl(), exit(), system(),</code> <code>max(x,y), min(x,y), qsort(),</code>

		<code>ran(), random(), srand(), ...</code>
<code>graphics.h</code>	graphics uso de tela grafica (ambiente DOS)	<code>initgraph(), closegraph(), detectgraph(), setcolor(), setbkground(), moveto(), lineto(), outtext(), ...</code>

DELIMITADORES

Pascal	Função	C
<code>(* *)</code> ou <code>{ }</code>	delimitadores de comentário	<code>/* */</code>
<code>;</code>	separar comandos e terminar declarações	<code>;</code>
<code>,</code>	separar identificadores e parâmetros	<code>,</code>
<code>=</code>	termo à esquerda é igual ao termo à direita	<code>==</code>
<code>< ></code>	termo à esquerda é diferente do termo à direita	<code>!=</code>
<code><</code> <code>></code> <code><=</code> <code>>=</code>	comparação de valores	<code><</code> <code>></code> <code><=</code> <code>>=</code>
<code>:=</code>	atribuição de valor	<code>=</code>
<code>begin ... end</code>	delimita blocos de instruções	<code>{ ... }</code>
<code>(...)</code>	delimita teste de condição	<code>(...)</code>

DECLARAÇÕES

Pascal	Função	C
<code>var</code>	declaração de variáveis	não há necessidade
<code>const</code>	declaração de constantes	<code>const</code>
<code>type</code>	declaração de estruturas ou tipos	<code>struct</code>
<code>typedef</code>	definição de estruturas ou tipos	<code>enum</code>

TIPOS DE DADOS NUMÉRICOS

Pascal	Inteiros	C
<code>shortint</code>	-128 a +127	<code>char<</code>
<code>integer</code>	-32.768 a +32.767	<code>int</code>
<code>longint</code>	-2.147.483.648 a +2.147.483.647	<code>long</code>

Pascal	Números de Ponto Flutuante	C
<code>real</code>	2,9*10 ⁽⁻³⁹⁾ a 1,7*10 ⁽³⁸⁾ (6 dígitos significativos) (pascal) 3,4*10 ⁽⁻³⁸⁾ a 3,4*10 ⁽³⁸⁾ (6 dígitos significativos) (c)	<code>float</code>
<code>double</code>	5*10 ⁽⁻³²⁴⁾ a 1,7*10 ⁽³²⁴⁾ (15-16 dígitos significativos) (pascal) 1,7*10 ⁽⁻³⁰⁸⁾ a 1,7*10 ⁽³⁰⁸⁾ (10-15 dígitos significativos) (c)	<code>double</code>
<code>extended</code>	3,4*10 ⁽⁻⁴⁹³²⁾ a 1,1*10 ⁽⁴⁹³²⁾ (19-20 dígitos significativos) (pascal) 1,7*10 ⁽⁻³⁰⁸⁾ a 1,7*10 ⁽³⁰⁸⁾ (18 dígitos significativos) (c)	<code>long double</code>

TIPOS DE DADOS ALFANUMÉRICOS

Pascal	Função	C
char	um caracter	char
string	corrente de caracteres	char

FORMATO PARA AS VARIÁVEIS

Pascal	Função	C
não há	declaração de formato de variáveis	<pre> "%c" character "%i" ou "%d" inteiro "%f" float "%lf" double </pre>

COMANDOS MAIS FREQUENTES

Pascal	Função	C
read;	lê um ou mais valores de um arquivo para uma ou mais variáveis	scanf();
write;	retorna caracteres gerados	printf();
gotoxy(x,y);	o cursor é movido para (x,y) sendo que: x varia entre 1 e 79 (colunas) y varia entre 1 e 24 (linhas)	gotoxy(x,y);
clrscr;	limpa a tela corrente; o cursor é movido para (1,1) (canto superior esquerdo da tela.	clrscr ();
readkey;	lê um caractere do teclado (qualquer) sem apresentá-lo na tela	getch ();
begin instruções; end;	(bloco) bloco de instruções {bloco}	{ instruções; }

COMANDOS DE LAÇOS DE DECISÃO

Pascal	for	C
for (k:=1 to k<=n) do (bloco);	executa um comando repetidamente, indicando um aumento na variável de controle	for (k=0 ;k<=n-1 ; k=k+1) {bloco}

Pascal	if	C
if (condição) then (bloco 1) else (bloco 2);	testa uma condição, se verdadeira executa o comando subordinado then, caso contrário, executa o comando else	if (condição) {bloco1} else {bloco2}

Pascal	while	C
--------	-------	---

while (condição) do (bloco);	repete uma série de comandos até a condição tornar-se falsa	while (condição) {bloco}
-------------------------------------	---	---------------------------------

Pascal	do	C
repeat (bloco) until ();	repete uma série de comandos até a condição tornar-se falsa, (é diferente do comando anterior pois executa pelo menos uma vez)	do {bloco} while ();

Pascal	goto	C
goto	transfere o fluxo do programa para o primeiro comando seguinte ao rótulo	goto

Pascal	switch	C
case (escolha) of opção: (bloco1); opção: (bloco2); opção: (bloco3); else opção_inválida; end;	selecionar uma de várias ações, dependendo do resultado de um teste	switch (escolha) { case opção: {bloco1}; break; case opção: {bloco2}; break; case opção: {bloco3}; break; default: msg_erro(); }

PRECEDÊNCIA DE OPERADORES

Pascal	Precedências	C
()	1	()
negação	2	negação
multiplicação divisão módulo	3	multiplicação divisão módulo
adição subtração	4	adição subtração
igual diferente maior (igual) menor (igual)	5	igual diferente maior (igual) menor (igual)

OPERADORES ARITMÉTICOS

Pascal	Função	C
+	adição	+
-	subtração	-
*	multiplicação	*
div	divisão inteira	/

/	divisão real	/
mod	resto da divisão expresso como inteiro (módulo)	%

OPERADORES LÓGICOS

Considerações sobre Verdadeiro ou Falso

Pascal	C
TRUE = verdadeiro FALSE = falso	1 = verdadeiro 0 = falso

Pascal	() E ()	C
() and ()	((1) e (1)) da (1) ((1) e (0)) da (0) ((0) e (1)) da (0) ((0) e (0)) da (0) caso as expressões sejam verdadeiras retorna 1 caso a expressão à esquerda seja falsa (0) não avalia a segunda parte e retorna 0.	() && ()

Pascal	() OU ()	C
() or ()	((1) ou (1)) da (1) ((1) ou (0)) da (1) ((0) ou (1)) da (1) ((0) ou (0)) da (0) caso uma das expressões seja verdadeira retorna 1 se a expressão à esquerda for verdadeira retorna 1 ele não avalia a segunda parte, caso contrário retorna 0.	() ()

APONTADORES

Pascal	Função	C
^nome	dado	*nome
^nome	referência ao dado na memória	&nome

EXEMPLO SIMPLES

Pascal	C
<pre> program exemplo; uses crt; var a,b,c,d: real; resp,s,n: char; begin clrscr; resp:='S'; while Resp:='S' do begin write('Somatória de 3 números'); write('Digite o primeiro número'); readln(a); write('Digite o segundo número'); readln(b); write('Digite o terceiro número'); readln(c); d:=a+b+c; write('A soma dos três números é: ',d) ; write('Continua (S/N) ?'); readln(resp); end; </pre>	<pre> /* programa exemplo */ #include < stdio.h > #include < conio.h > float a,b,c,d; char resp; int main() { clrscr (); do { printf("Somatória de 3 números"); printf("Digite o primeiro número "); scanf("%f",&a); printf("Digite o segundo número"); scanf("%f",&b); printf("Digite o terceiro número"); scanf("%f",&c); d=a+b+c; printf("A soma dos três números é: %f ",d) ; printf("Continua (S/N) ?"); scanf("%c",resp); } while (resp='s'); return (0); </pre>

```
end.
```

```
}
```

Observações importantes para converter programas **C for DOS** em programas **C for Windows**.

- Talvez seja necessário incluir novas bibliotecas:
conio.h se usar clrscr();
ctype.h se usar toupper();
dos.h se usar sound(), delay() e nosound(); (*não é garantido que funcione!*).
 - O comando **getch()** não é necessário no final do programa, pode-se excluí-lo.
Em vez disso, informe o usuário para fechar a janela após rodar o programa (clcando no X).
 - Inclua o comando **return(0)** ao final do programa, antes da última chave!
Ou use a opção **void main ()**
-

Preparada em 15/03/97. Última atualização em 10/04/97. LEIA/DEA/FEA/Unicamp

```

\!!!!/
( õ õ )
-----oOOO--( )-----
| Arquivo baixado da GEEK BRASIL |
| O seu portal de informática e internet |
| http://www.geekbrasil.com.br |
| Dúvidas ou Sugestões? |
| webmaster@geekbrasil.com.br |
-----oOOO-----
|_|_|
|| ||
ooO Ooo
```