SSC0101 - ICC1 – Teórica

Introdução à Ciência da Computação I

Comandos de Decisão – Parte I

Prof. Vanderlei Bonato: vbonato@icmc.usp.br
Prof. Claudio Fabiano Motta Toledo: claudio@icmc.usp.br

Sumário

- Estrutura condicional simples
- Estrutura condicional composta
- Operadores para expressões de teste
- Precedência de operadores

Estrutura condicional simples

ALGORITMO	LINGUAGEM C
SE (expressão de teste) ENTÃO instrução;	if (expressão de teste) instrução;

SE (expressão de teste)
  ENTÃO INÍCIO
  Instrução_1;
  Instrução_2;
  ...
  Instrução_n;
  FIM
	if (expressão de teste)
	{
	  Instrução_1;
	  Instrução_2;
	  ...
	  Instrução_n;
	}

Estrutura condicional simples

Exemplo:

ALGORITMO

```
ALGORITMO
DECLARE ch LITERAL
LEIA ch
SE (ch = 'p')
  ENTÃO ESCREVA “Tecla p”
FIM_ALGORITMO
```

LINGUAGEM C

```
int main()
{
    char ch;
    ch = getche();
    if (ch == 'p')
    {
        printf("\n Tecla p \n");
    }
    system("pause" );
}
```

Estrutura condicional composta

ALGORITMO

```
SE (expressão de teste)
  ENTÃO instrução_1;
  SENÃO instrução_2;
```

```
SE (expressão de teste)
  ENTÃO INÍCIO
          Instrução_1
          Instrução_2
          ...
          Instrução_n
          FIM
  SENÃO INÍCIO
          Instrução_1
          Instrução_2
          ...
          Instrução_n
          FIM
```

Estrutura condicional composta

Exemplo:

```
ALGORITMO
DECLARE ch LITERAL
LEIA ch
SE (ch = 'p')
  ENTÃO ESCREVA ”Tecla p foi pressionada”
  SENÃO ESCREVA “Voce não pressionou a tecla p”
FIM_ALGORITMO
```

Estrutura condicional composta

LINGUAGEM C

```
if (expressão de teste)
    instrução_1;
else
    instrução_2;
```

```
if (expressão de teste)
{
      instrução_1;
      Instrução_2;
      ...
      Instrução_n;
}
else
{
     instrução_1;
     instrução_2;
      ...
      Instrução_n;
}
```

Estrutura condicional composta

Exemplo:

```
int main()
{
     char ch;
     ch = getche();
     if (ch == 'p')
     {
          printf("\n  Voce pressionou a tecla p.\n");
     }
    else
    {
         printf("\n  Voce não pressionou a tecla p.\n");
    }
    system("pause" );
}
```

Comandos aninhados

ALGORITMO

```
SE (expressão de teste_1)
  ENTÃO SE (expressão de teste_2)
              ENTÃO instrução_1;
              SENÃO instrução_2;
  SENÃO instrução_3;
```

Exemplo I

```
ALGORITMO
DECLARE ch1, ch2 LITERAL
ESCREVA “Entre caractere 1”
LEIA ch1
ESCREVA “Entre caractere 2”
LEIA ch2
SE ch1 = 'p‘
  ENTÃO SE ch2 = ‘q‘
                ENTÃO ESCREVA “Você digitou p e q.”
                SENÃO  ESCREVA “Você digitou p e não q.”
  SENÃO ESCREVA “Você NÃO digitou p e q.”
FIM_ALGORITMO
```

Exemplo II

```
ALGORITMO
DECLARE ch1, ch2 LITERAL
ESCREVA “Entre caractere 1”
LEIA ch1
SE ch1 = 'p‘
  ENTÃO INÍCIO
              ESCREVA “Entre caractere 2:”
              LEIA ch2
             SE ch2 = 'q‘
                ENTÃO ESCREVA “Você digitou p e q.”
                SENÃO  ESCREVA “Você digitou p e não q.”
             FIM
  SENÃO ESCREVA “Você NÃO digitou p e q.”
FIM_ALGORITMO
```

Comandos aninhados

LINGUAGEM C

```
if (expressão de teste_1)
   if (expressão de teste_2)
         instrução_1;
   else
         instrução_2;
else
   instrução_3;
```

Exemplo de if/if-else aninhados em C

```
int main()
{
    char ch1, ch2;
    printf("\n  Entre caractere 1:");
    ch1 = getche();
    if (ch1 == 'p')
    {
       printf("\n  Entre caractere 2:");
       ch2 = getche();
       if (ch2 == 'q')
       {
          printf("\n  Você digitou p e q.\n");
       }
       else
       {
           printf("\n  Você digitou p e não q.\n");
       }
    }
    else
    {
       printf("\n  Voce NÃO digitou p e q.\n");
    }
    system("PAUSE");
}
```

Revisão:
operadores para expressão de teste

ALGORITMO

Relacionais

>	maior
>=	maior ou igual
<	menor
<=	menor ou igual
=	igualdade
≠	diferente

Lógicos

E

OU

NÃO (unário)

Exemplo com operadores lógicos

```
ALGORITMO
DECLARE ch1, ch2;
ESCREVA “Entre caractere 1”
LEIA ch1
ESCREVA “Entre caractere 2:”
LEIA ch2
SE ch1 = 'p' E ch2 = 'q’
  ENTÃO  ESCREVA “Você digitou p e q”
  SENÃO SE ch1 = 'p' OU ch2 = 'q'
                ENTÃO ESCREVA “Você digitou p ou q”
SE(NÃO(ch1 = 'p') E NÃO(ch2 = 'q'))
   ENTÃO ESCREVA “Você NÃO digitou p e nem q”
FIM_ALGORITMO
```

Revisão:
operadores para expressão de teste

LINGUAGEM C

Relacionais

>	maior
>=	maior ou igual
<	menor
<=	menor ou igual
==	igualdade
!=	diferente

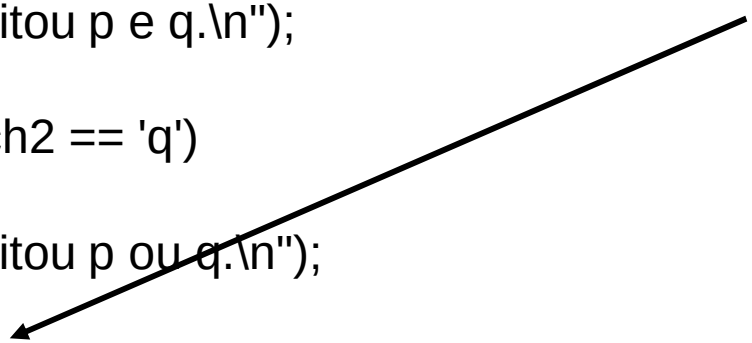
Lógicos

&&	E
||	OU
!	Negação (unário)

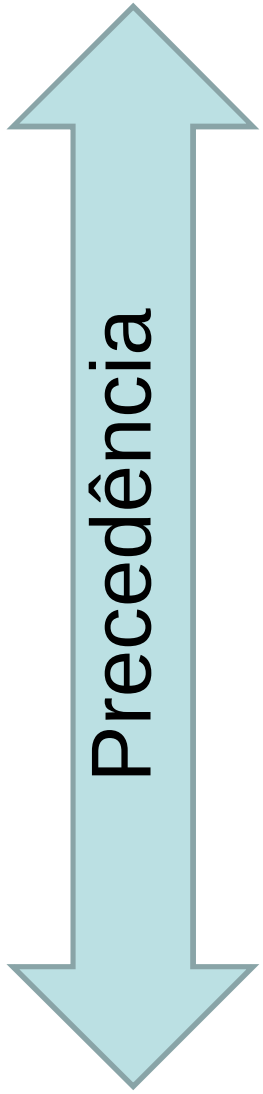
Exemplo com operadores lógicos

```
int main()
{
    char ch1, ch2;
    printf("\n  Entre caractere 1:");
    ch1 = getche();
    printf("\n  Entre caractere 2:");
    ch2 = getche();
    if (ch1 == 'p' && ch2 == 'q')
    {
       printf("\n  Você digitou p e q.\n");
    }
    else if (ch1 == 'p' || ch2 == 'q')
    {
       printf("\n  Você digitou p ou q.\n");
    }
    if (!(ch1 == 'p') && !(ch2 == 'q'))
    {
       printf("\n  Você NÃO digitou p e nem q.\n");
    }
    system("PAUSE");
}
```

O que acontece se retirarmos os parênteses do operador unário <!> ?



Precedência e associatividade de operadores



Operadores	Associatividade
() [] ->	Esquerda para direita
! ~ ++ -- + - * & (tipo) sizeof	Direita para esquerda
* / %	Esquerda para direita
+ -	Esquerda para direita
<< >>	Esquerda para direita
< <= > >=	Esquerda para direita
== !=	Esquerda para direita
&	Esquerda para direita
^	Esquerda para direita
|	Esquerda para direita
&&	Esquerda para direita
||	Esquerda para direita
?:	Direita para esquerda
= += -= *= /=	Direita para esquerda

Exercícios

Apresente os programas dos exercícios abaixo em pseudocódigo e linguagem C.

1. Faça um programa que receba três números e mostre-os em ordem decrescente.
2. Faça um programa que receba três notas de um aluno, calcule e mostre a média e a mensagem constante nos itens a seguir. Aos alunos que ficaram para exame, calcule e mostre a nota que deverão tirar para serem aprovados, considerando que a média exigida é 6,0.

   [0,0; 3,0[: Reprovado

   [3,0; 7,0[: Exame

   [7,0; 10,0] : Aprovado

Tabela ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Referências

Ascencio AFG, Campos EAV. Fundamentos de programação de computadores. São Paulo : Pearson Prentice Hall, 2006. 385 p.

VICTORINE VIVIANE MIZRAHI, Treinamento em Linguagem C – Módulo 1 e Módulo 2, Makron Books, 1990.

FIM Aula 5