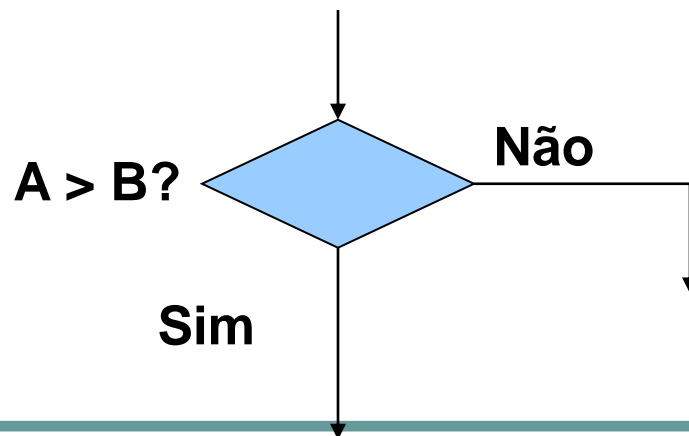


Comandos de Controle Condicional

Fluxogramas

- **Condição ou Decisão**
 - Representado por losangos
 - Normalmente contém uma pergunta do tipo Sim/Não ou um teste de Verdadeiro/Falso.
 - Mudança no fluxo



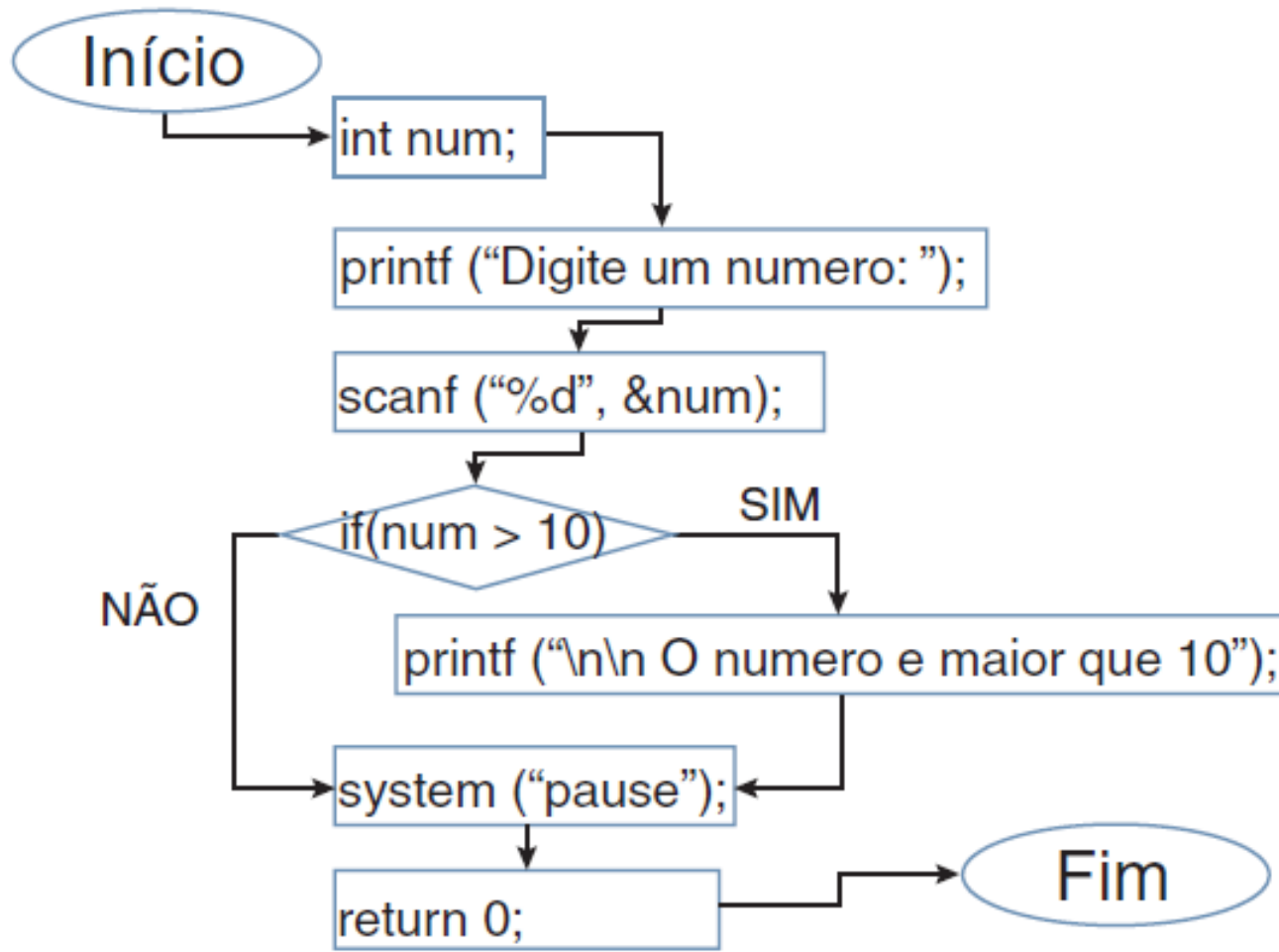
Comando if

- Em C, o comando if é utilizado quando for necessário escolher entre dois fluxos de execução, ou quando se deseja executar um comando sujeito ao resultado de um teste.
- A forma geral de um comando **if** é:
if (condição) {
 sequência de comandos;
}

Exemplo if

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      int num;
05      printf ("Digite um numero: ");
06      scanf("%d",&num);
07      if(num > 10)
08          printf ("O numero e maior do que 10\n");
09
10      system("pause");
11      return 0;
12  }
```

Exemplo if



Comando if

- Usa-se chaves { } para delimitar o bloco de instruções que pertence ao if

```
if(num > 10){  
    printf ("\n\n O numero e maior que 10");  
}
```
- As chaves podem ser ignoradas se a instrução for única.

```
if(num > 10)  
    printf ("\n\n O numero e maior que 10");
```

Condição do if

- A condição pode ser uma expressão usando operadores matemáticos, lógicos e relacionais
 - +, -, *, /, %
 - &&, ||
 - >, <, >=, <=, ==, !=
- Ex:
 - $(x > 10 \ \&\& \ y \leq x-1)$

Exercício

- Dada o valor da nota de um aluno, monte a expressão if que verifica se ele precisará fazer o exame especial. O aluno deverá fazer exame especial se sua nota for maior ou igual a 40 e menor do que 60.

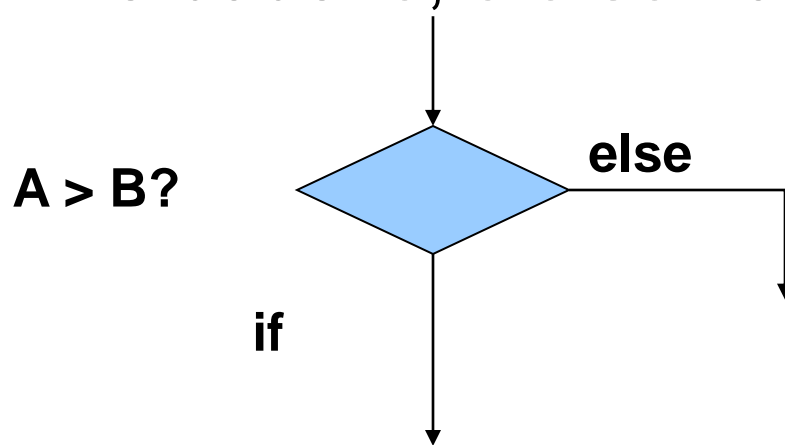
Exercício

```
#include <stdio.h>
#include <stdlib.h>
int main ( ){
    int num;

    printf ("Digite a nota: ");
    scanf ("%d",&num);
    if ((num >= 40) && (num < 60))
        printf ("O aluno deve fazer a prova sub \n");
    system("pause");
    return 0;
}
```

Comando else

- O comando else pode ser entendido como sendo um complemento do comando if.
 - Se o **if** diz o que fazer quando a condição é verdadeiro, o **else** tratá da condição falsa.



Comando else

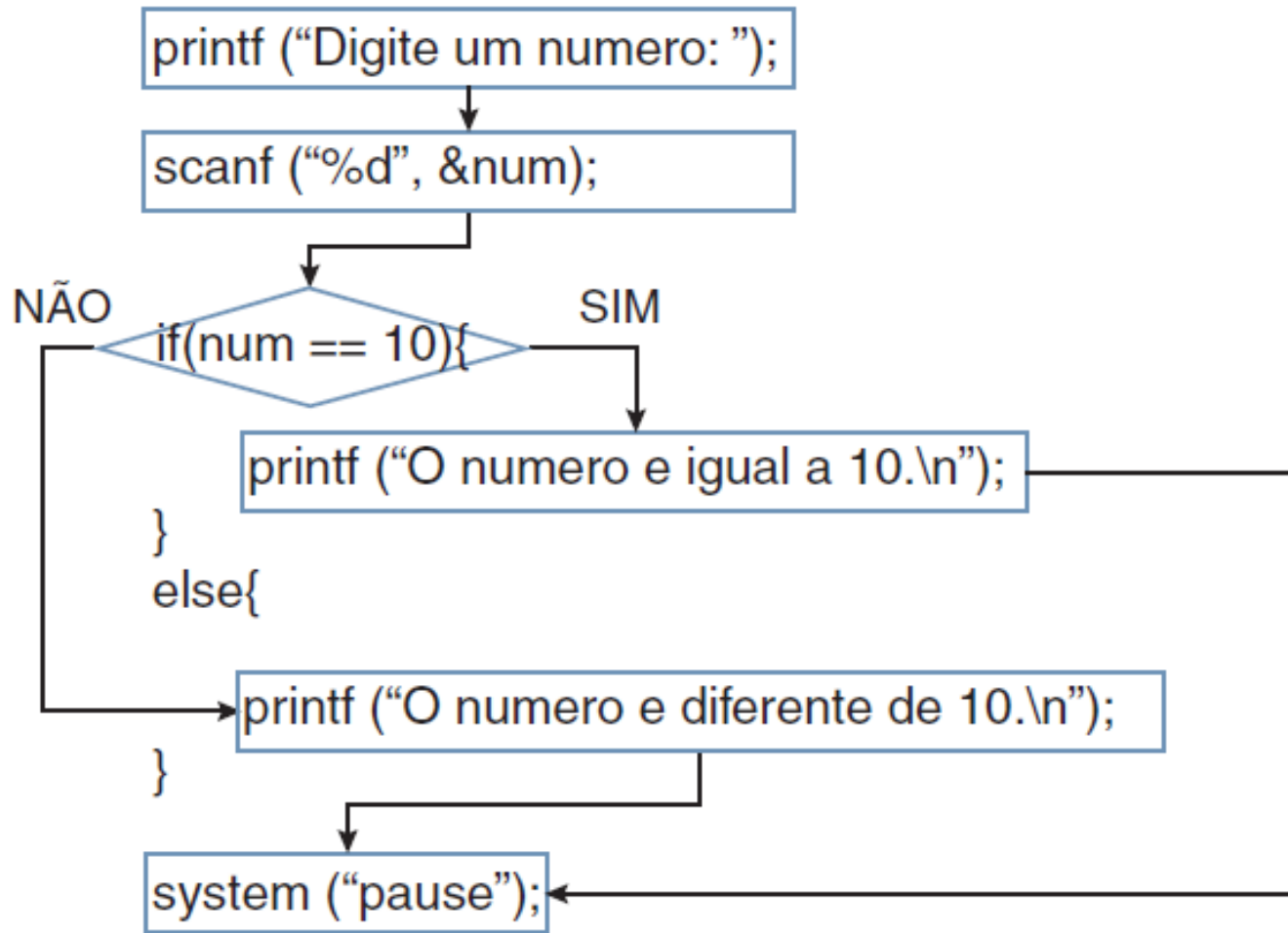
- O comando if-else tem a seguinte forma geral:

```
if(condição) {  
    sequência de comandos 1;  
} else{  
    sequência de comandos 2;  
}
```

Exemplo if-else

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      int num;
05      printf ("Digite um numero: ");
06      scanf("%d", &num);
07      if(num == 10){
08          printf ("O numero e igual a 10.\n");
09      } else{
10          printf ("O numero e diferente de 10.\n");
11      }
12      system("pause");
13      return 0;
14  }
```

Exemplo if-else



Comando if-else

- Como no caso do comando if, as chaves podem ser ignoradas se a instrução contida no **else** for única.

else

```
printf ("O numero e diferente de 10.\n");
```

- ou

```
else{
```

```
printf ("O numero e diferente de 10.\n");
```

```
}
```

Comando if-else



A sequência de comandos de **if** é independente da sequência de comandos de **else**. Cada comando tem o seu próprio conjunto de chaves ({}).

Uso das chaves no comando if-else

	Certo	Errado
01	if (condicao){	if (condicao){
02	sequencia de comandos;	sequencia de comandos;
03	}	else
04	else {	sequencia de comandos;
05	sequencia de comandos;	}
06	}	

Aninhamento de if

- O **if** aninhado é simplesmente um **if** dentro da declaração de um outro **if** externo.
 - A estrutura if-else-if é apenas uma extensão da estrutura if-else.
- O único cuidado que devemos ter é o de saber exatamente a qual **if** um determinado **else** está ligado.

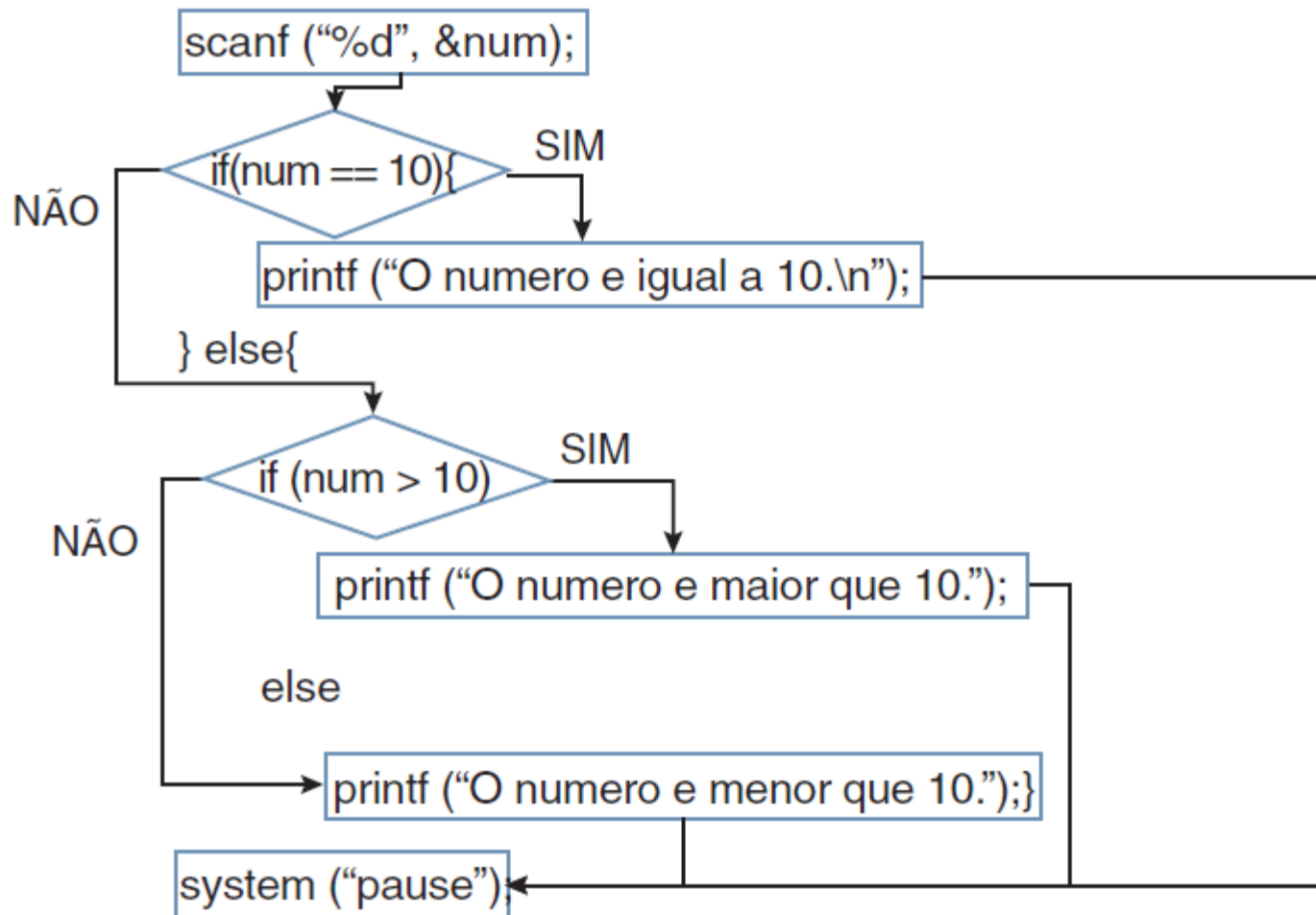
Aninhamento de if

```
if(condição 1) {  
    seqüência de comandos 1;  
} else if(condição 2) {  
    seqüência de comandos 2;  
}  
  
...  
else if(condição_n) {  
    seqüência de comandos n;  
} else {  
    seqüência de comandos default;  
}
```

Exemplo aninhamento

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      int num;
05      printf("Digite um numero: ");
06      scanf("%d", &num);
07      if(num == 10){
08          printf("O numero e igual a 10.\n");
09      } else{
10          if(num > 10)
11              printf("O numero e maior que 10.\n");
12          else
13              printf("O numero e menor que 10.\n");
14      }
15      system("pause");
16      return 0;
17  }
```

Exemplo aninhamento



Aninhamento de if

- Não existe aninhamento de else's
 - Para cada else deve existir um if anterior, mas nem todo if precisa ter um else.

```
if (cond1)
```

```
    comando if1;
```

```
    else
```

```
        comando else1;
```

```
    else
```

```
        comando else2;
```

Errado

Exercício

- Dada o valor da nota de um aluno, monte o conjunto de if's e else's que verifica se ele foi aprovado, reprovado ou precisará fazer o exame especial.

Exercício

```
printf ("Digite a nota: ");
scanf ("%d",&d);
if (d >= 60)
    printf ("Aluno aprovado \n");
else if (d < 40)
    printf ("Aluno reprovado \n");
else
    printf ("O aluno deve fazer a prova sub \n");
```

Expressão Condicional

- Quando o compilador avalia uma condição, ele quer um valor de retorno para poder tomar a decisão.
 - 0 – falso
 - 1 – verdadeiro (ou outro valor qualquer)
- Uma variável sozinha pode ser uma "expressão" e esta retornar o seu próprio valor.

Expressão Condicional

- Isto quer dizer que teremos as seguintes expressões:
 - *int num;*
 - *if (num!=0)*
 - *if (num==0)*
- equivalem a
 - *int num;*
 - *if (num)*
 - *if (!num)*

O Operador ?

- Também conhecido como operador ternário
- A expressão condicional “?:” é uma simplificação do if-else utilizada tipicamente para atribuições condicionais

O Operador ?

- Uma expressão como
if (a > 0)
 b = -150;
else
 b = 150;
- pode ser simplificada usando-se o operador ? da seguinte maneira:
b = a > 0 ? -150 : 150;

Exercício

- Dado dois números x e y , retorne o maior na variável z :
 - Usando if-else
 - Usando o operador ternário

Exercício

	Usando if-else	Usando operador ternário
01	#include <stdio.h>	#include <stdio.h>
02	#include <stdlib.h>	#include <stdlib.h>
03	int main(){	int main(){
04	int x,y,z;	int x,y,z;
05	printf("Digite x:");	printf("Digite x:");
06	scanf("%d",&x);	scanf("%d",&x);
07	printf("Digite y:");	printf("Digite y:");
08	scanf("%d",&y);	scanf("%d",&y);
09	if (x > y)	z = x > y ? x : y;
10	z = x;	printf("Maior = %d\n",z);
11	else	system("pause");
12	z = y;	return 0;
13	printf("Maior = %d\n",z);	}
14	system("pause");	
15	return 0;	
16	}	

○ Operador ?

- O operador ? é limitado
 - não atende a uma gama muito grande de casos.
- mas pode ser usado para simplificar expressões complicadas. Uma aplicação interessante é a do contador circular.
 - `index = (index== 3) ? index= 0: ++index;`

O comando switch

- O comando switch é próprio para se testar uma variável em relação a diversos valores pré-estabelecidos.
 - Parecido com if-else-if, porém não aceita expressões, apenas constantes.
 - O switch testa a variável e executa a declaração cujo “case” corresponda ao valor atual da variável.

O comando switch

```
switch (var) {  
  case valor 1:  
    comandos 1;  
    break;  
  case valor k:  
    comandos k;  
    break;  
  default:  
    comandos default;  
    break;  
}
```

O comando switch

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      char ch;
05      printf("Digite um simbolo de pontuacao: ");
06      ch = getchar();
07      switch( ch ) {
08          case \': printf("Ponto.\n" ); break;
09          case \',: printf("Virgula.\n" ); break;
10          case \': printf("Dois pontos.\n" ); break;
11          case \';: printf("Ponto e virgula.\n"); break;
12          default : printf("Nao eh pontuacao.\n" );
13      }
14      system("pause");
15      return 0;
16  }
```


O comando switch

Início

```
char ch;
```

```
ch = getchar();
```

```
switch( ch) {
```

Igual?

```
case '.': printf( "Ponto.\n" ); break;
```

Igual?

```
case ',': printf( "Virgula.\n" ); break;
```

Igual?

```
case ' ': printf( "Dois pontos.\n" ); break;
```

Igual?

```
case ';': printf( "Ponto e virgula.\n" ); break;
```

```
default : printf( "Nao eh pontuacao.\n" );
```

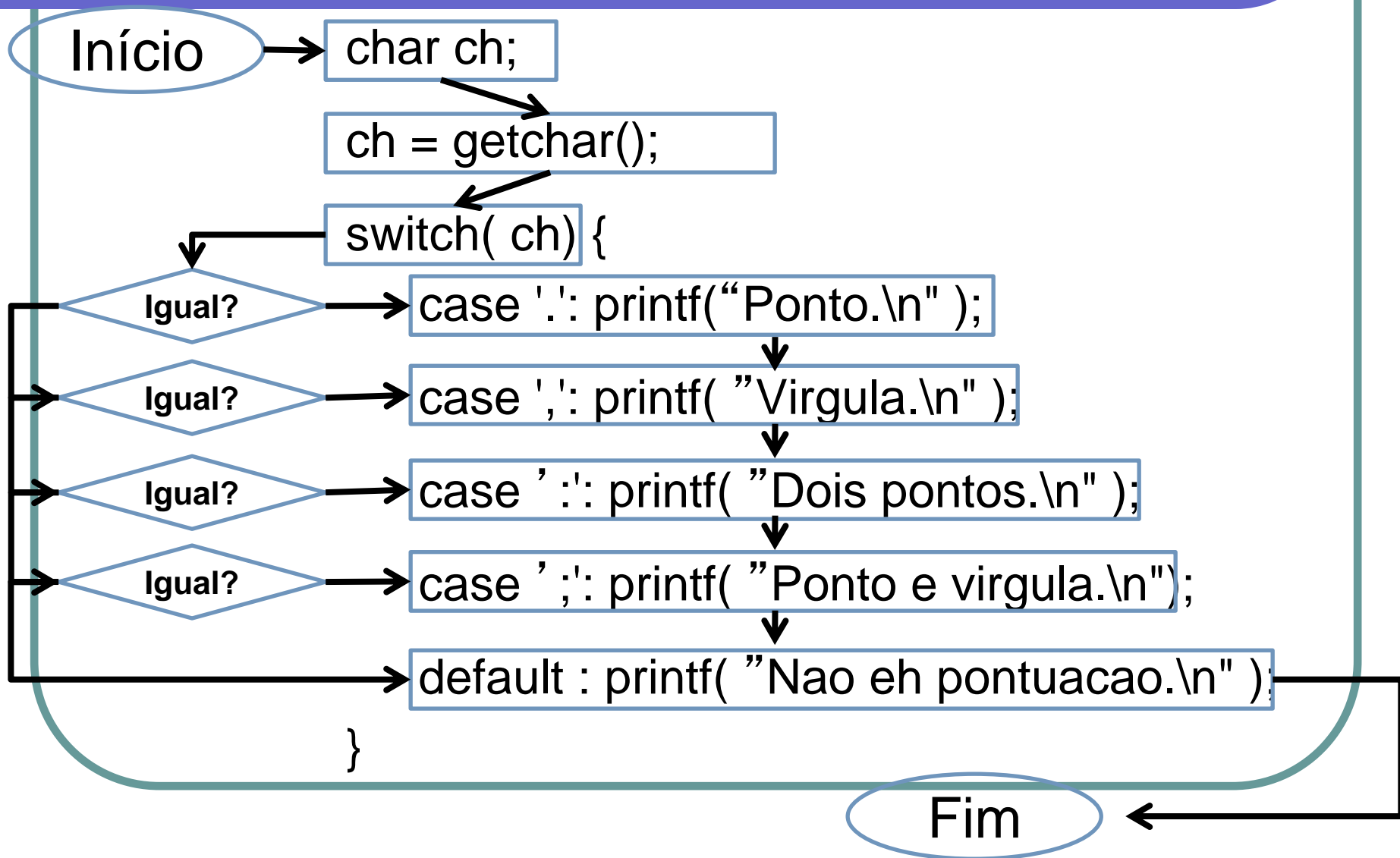
```
}
```

Fim

O comando switch sem break

```
01  #include <stdio.h>
02  #include <stdlib.h>
03  int main(){
04      char ch;
05      printf("Digite um simbolo de pontuacao: ");
06      ch = getchar();
07      switch( ch ) {
08          case '.': printf("Ponto.\n" );
09          case ',': printf("Virgula.\n" );
10          case ':': printf("Dois pontos.\n");
11          case ';': printf("Ponto e virgula.\n");
12          default : printf("Nao eh pontuacao.\n" );
13      }
14      system("pause");
15      return 0;
16  }
```

O comando switch sem break



O comando switch

```
int num;  
scanf("%d",&num);  
switch( num ) {  
case 0: printf("0"); /* 0123456789 */  
case 1: printf("1"); /* 123456789 */  
case 2: printf("2"); /* 23456789 */  
case 3: printf("3"); /* 3456789 */  
case 4: printf("4"); /* 456789 */  
case 5: printf("5"); /* 56789 */  
case 6: printf("6"); /* 6789 */  
case 7: printf("7"); /* 789 */  
case 8: printf("8"); /* 89 */  
case 9: printf("9"); /* 9 */  
}
```