

Linguagem C - Controle de decisão

Prof^a. Rose Yuri Shimizu

Controle de fluxo : tomada de decisão

- Às vezes um conjunto de instruções precisam ser executadas e outras vezes esse conjunto não deve ser executado
- A tomada de decisão, depende de uma condição ou restrição para que a ação seja executada
- Para testar a condição usamos a ideia do “se *isso* então faça *aquilo*”

Controle de fluxo : IF

- SE a “condição” for verdadeira ($!=0$) o bloco de instruções será executado
- { } : chaves delimitam o bloco de instruções

```
1
2     if(condição 1)
3     {
4         //bloco de instruções
5     }
6
7     if(condição 2)
8     {
9         //bloco de instruções
10    }
11
```

Controle de fluxo : IF - exemplo

- Dividir dois números inteiros

```
1 #include <stdio.h>
2 int main() {
3     int num, den, div=0;
4     scanf("%d%d", &num, &den);
5
6     //se den for diferente de 0
7     if(den != 0)
8     {
9         //então faça a divisão e imprima o resultado
10        div = num/den;
11        printf("A divisão de %d e %d : %d\n", num, den, div);
12    }
13
14    //se den for igual a 0
15    if(den == 0)
16    {
17        //então imprima a mensagem de erro
18        printf("A divisão por zero não é permitida.\n");
19    }
20
21    return 0;
22 }
```

Controle de fluxo : IF, ELSE IF

```
1 if(condição 1)
2 {
3     //bloco 1
4 }
5 else if(condição 2)
6 {
7     //bloco 2
8 }
9 .
10 . //vários else if
11 .
12 else if(condição n)
13 {
14     //bloco n
15 }
```

SE a *condição 1* for verdadeira

- *bloco 1* de instruções é executado

SENÃO SE a *condição 2* for verdadeira

- *bloco 2* de instruções é executado

...

SENÃO SE a *condição n'* for verdadeira

- *bloco n* de instruções é executado

Controle de fluxo : IF, ELSE IF

```
1 if(condição 1)
2 {
3     //bloco 1
4 }
5 else if(condição 2) //condição 1 falsa
6 {
7     //bloco 2
8 }
9 .
10 . //vários else if
11 .
12 else if(condição n) //condição 1 até n-1 falsas
13 {
14     //bloco n
15 }
```

- “condição 2” será verificada se e somente “condição 1” for falsa
- “condição n” será verificada se e somente “condição 1” e “condição 2” forem falsas
- Caso nenhuma condição seja verdadeira, nenhum bloco será executado

Controle de fluxo : IF, ELSE IF - exemplo

- Dado um código imprimir o código e o nome do lanche correspondente

```
1 #include <stdio.h>
2 int main()
3 {
4     int cod;
5     scanf("%d", &cod);
6
7     if(cod == 1)
8     {
9         printf("%d Cachorro quente\n", cod);
10    }
11    else if(cod == 2)
12    {
13        printf("%d X-salada\n", cod);
14    }
15    else if(cod == 3)
16    {
17        printf("%d X-bacon\n", cod);
18    }
19    return 0;
20 }
```

Controle de fluxo : IF, ELSE

- SE a “condição” for verdadeira o “bloco 1” de instruções será executado
- SENÃO “bloco 2” de instruções será executado
- Ou seja, ALGUM bloco será executado: ou o bloco do `if` ou o bloco do `else`

```
1 if(condição)
2 {
3     //bloco 1
4 }
5 else
6 {
7     //bloco 2
8 }
```


Controle de fluxo : IF, ELSE - exemplo

- Dividir dois números inteiros

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, den, div=0;
5     scanf("%d%d", &num, &den);
6
7     if( den != 0 )
8     {
9         div = num/den;
10    }
11
12    if( den == 0 )
13    {
14        printf("A divisão por zero não é permitida.\n");
15    }
16
17    return 0;
18 }
```

Controle de fluxo : IF, ELSE - exemplo

- Dividir dois números inteiros

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, den, div=0;
5     scanf("%d%d", &num, &den);
6
7     if( den != 0 )
8     {
9         div = num/den;
10    }
11    else if( den == 0 )
12    {
13        printf("A divisão por zero não é permitida.\n");
14    }
15
16    return 0;
17 }
```

Controle de fluxo : IF, ELSE - exemplo

- Dividir dois números inteiros

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, den, div=0;
5     scanf("%d%d", &num, &den);
6
7     if( den != 0 )
8     {
9         div = num/den;
10    }
11    else //if(den==0) desnecessário, pois (den==0) == !(den!=0)
12    {
13        printf("A divisão por zero não é permitida.\n");
14    }
15
16    return 0;
17 }
```

Controle de fluxo : IF, ELSE - exemplo

- Dividir dois números inteiros

```
1 #include <stdio.h>
2 int main()
3 {
4     int num, den, div=0;
5     scanf("%d%d", &num, &den);
6
7     if( den != 0 )
8     {
9         div = num/den;
10    }
11    else
12    {
13        printf("A divisão por zero não é permitida.\n");
14    }
15
16    return 0;
17 }
```

Controle de fluxo : IF, ELSE - exemplo

- Maior entre dois números

```
1 #include <stdio.h>
2 int main(){
3     float a, b;
4     scanf("%f %f", &a, &b);
5
6     if(a>=b)
7     {
8         printf("Largest number = %.2f\n", a);
9     }
10    else if(a<b)
11    {
12        printf("Largest number = %.2f\n", b);
13    }
14
15    return 0;
16 }
```

Controle de fluxo : IF, ELSE - exemplo

- Maior entre dois números

```
1 #include <stdio.h>
2 int main(){
3     float a, b;
4     scanf("%f %f", &a, &b);
5
6     if(a>=b)
7     {
8         printf("Largest number = %.2f\n", a);
9     }
10    else
11    {
12        printf("Largest number = %.2f\n", b);
13    }
14
15    return 0;
16 }
```

Controle de fluxo : IF, ELSE IF, ELSE

```
1 if(condição 1)
2 {
3     //bloco 1
4 }
5 else if(condição 2)
6 {
7     //bloco 2
8 }
9 .
10 . //vários else if
11 .
12 else if(condição n-1)
13 {
14     //bloco n-1
15 }
16 else
17 {
18     //bloco n
19 }
```

SE a *condição 1* for verdadeira

- o *bloco 1* de instruções é executado

SENÃO SE a *condição 2* for verdadeira

- o *bloco 2* de instruções é executado

...

SENÃO SE a *condição n-1* for verdadeira

- o *bloco n-1* de instruções é executado

SENÃO

- o "bloco n" de instruções é executado

- Caso nenhuma condição seja verdadeira, o último bloco é executado

Controle de fluxo : IF, ELSE IF, ELSE - exemplo

- Dado um código imprimir o código e o nome do lanche correspondente; se não for código de algum lanche, informe “Lanche desconhecido”

```
1 #include <stdio.h>
2 int main()
3 {
4     int cod;
5     scanf("%d", &cod);
6
7     if(cod == 1)
8     {
9         printf("%d Cachorro quente\n", cod);
10    }
11    else if(cod == 2)
12    {
13        printf("%d X-salada\n", cod);
14    }
15    else
16    {
17        printf("Lanche desconhecido\n");
18    }
19    return 0;
20 }
```


Controle de fluxo : IF, ELSE IF, ELSE - exemplo

- Maior entre 3 números

```
1 #include <stdio.h>
2 int main(){
3     float a, b, c;
4
5     scanf("%f %f %f", &a, &b, &c);
6
7     if(a>=b && a>=c)
8     {
9         printf("Maior = %.2f\n", a);
10    }
11    else if(b>=a && b>=c)
12    {
13        printf("Maior = %.2f\n", b);
14    }
15    else if(c>=a && c>=b)
16    {
17        printf("Maior = %.2f\n", c);
18    }
19    return 0;
20 }
```

Controle de fluxo : IF, ELSE IF, ELSE - exemplo

- Maior entre 3 números

```
1 #include <stdio.h>
2 int main(){
3     float a, b, c;
4
5     scanf("%f %f %f", &a, &b, &c);
6
7     if(a>=b && a>=c)
8     {
9         printf("Maior = %.2f\n", a);
10    }
11    else if(b>=a && b>=c)
12    {
13        printf("Maior = %.2f\n", b);
14    }
15    else //se nem a e nem b for o maior, só resta c
16    {
17        printf("Maior = %.2f\n", c);
18    }
19    return 0;
20 }
```

Controles de fluxo : resumo

- Tipos de decisão
 1. `if(){ }` : início
 2. `else if() { }` : 0 ou + alternativas condicionadas
 3. `else { }` : 0 ou 1 alternativa padrão (se tudo der errado)
- Condições:
 - Expressões aritméticas, relacionais e/ou lógicas
 - Executam o bloco de instruções se e somente se a condição for verdadeira (`!=0`)
- `{ }` : chaves delimitam o bloco de instruções

Controles de fluxo - resumo

```
1 if(condições 1) //condição inicial
2 {
3     //bloco de instruções
4 }
5 else if(condições 2) //alternativas condicionadas
6 {
7     //bloco de instruções
8 }
9 ...
10 else if(condições n) //alternativas condicionadas
11 {
12     //bloco de instruções
13 }
14 else //alternativa padrão
15 {
16     //instruções
17 }
```

Controle de fluxo : Blocos

Blocos do IF : { }

Quando usar as chaves?

```
1 #include <stdio.h>
2 int main()
3 {
4     int a=40,b=30;
5     if(b>a)                //falso
6         printf("Alpha\n"); //Controlado pelo if
7     printf("Beta\n");     //Nao controlado pelo if
8
9     if(a==b)              //falso
10    {
11        printf("Charlie\n"); //Controlado pelo if
12        printf("Delta\n");  //Controlado pelo if
13    }
14    printf("Echo\n");      //Nao controlado pelo if
15    return 0;
16 }
17 /*
18 Saida:
19     Beta
20     Echo
21 */
```

Controle de fluxo : IF aninhados (if dentro de if)

Exemplo: dado uma idade, se for maior de idade ler da entrada (teclado) e imprimir "Possui carteira" se for 1 senão "Não possui carteira" se for zero.

```
1 int main()
2 {
3     int idade, dirige;
4     scanf("%d", &idade);
5     if(idade >= 18)
6     {
7         scanf("%d", &dirige);
8         if(dirige==1)
9         {
10            printf("Possui carteira\n");
11        }
12        else
13        {
14            printf("Não possui carteira\n");
15        }
16    }
17    return 0;
18 }
```

Controle de fluxo : múltiplos IF

- IFs independentes
- Todos serão verificados

```
1 int i;
2 scanf("%d", &i);
3
4 if(i>=0)
5 {
6     printf("%d eh positivo\n", i);
7 }
8
9 if(!i%2) //i%2==0 : se i%2 for zero, !zero = 1
10 {
11     printf("%d eh par\n", i);
12 }
13 else
14 {
15     printf("%d eh impar\n", i);
16 }
```

Controle de fluxo : IF, ELSE, ELSE IF

Quando utilizar cada um?

- Vários **if**:
 - Condições não relacionadas, que precisam ser testadas independentemente (testar todas as condições);
 - **Várias opções que podem ser satisfeitas**;
 - dados os lados, imprimir o(s) tipo(s) do triângulo
 - dado um número, imprimir se é par e se é positivo
- **if else if**:
 - Condições relacionadas, cujo bloco só será executado caso a condição anterior não for satisfeita;
 - **Várias opções mas somente uma pode ser satisfeita**;
 - dado o código, imprimir o lanche correspondente
 - dado um número, imprimir em qual intervalo se encontra
- **else**:
 - Quando há uma saída padrão, ou seja, quando **nenhuma condição for satisfeita** o bloco de instruções controladas pelo **else** será executado;
 - dados os lados, imprimir se forma triângulo ou não
 - dado um número, imprimir se é par ou ímpar

Controle de fluxo : IF + operadores lógicos - otimizar fluxo

- CENÁRIO:
 - Você está no ponto
 - Está vindo um ônibus
 - Você precisa decidir se vai pegar este ônibus
- DECISÕES
 - O que você analisa para decidir?
 - Para onde vai o ônibus.

Vai para o FGA? :)

Vai para o FGA? :)

SE *vai para o FGA*

– “vou pegar o ônibus”

Não vai para o FGA? :(

SE *vai para o FGA*

– “vou pegar o ônibus”

Não vai para o FGA? :(

SE *vai para o FGA*

– “vou pegar o ônibus”

SENÃO SE *não vai para o FGA*

– “vou esperar o próximo”

Mas não gosto de ônibus cheio. Só vou caso não esteja lotado :P

SE vai para o FGA

– “vou pegar o ônibus”

SENÃO SE não vai para o FGA

– “vou esperar o próximo”

Mas não gosto de ônibus cheio. Só vou caso não esteja lotado :P

SE *vai para o FGA*

SE *não está lotado*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

– “vou esperar o próximo”

SENÃO SE *não vai para o FGA*

– “vou esperar o próximo”

Vixi, tenho tempo para esperar?! :O

SE *vai para o FGA*

SE *não está lotado*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

– “vou esperar o próximo”

SENÃO SE *não vai para o FGA*

– “vou esperar o próximo”

Vixi, tenho tempo para esperar?! :O

SE *vai para o FGA*

SE *não está lotado*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

SE *não estou atrasado*

– “vou esperar o próximo”

SENÃO SE *estou atrasado*

– “vou pegar o ônibus”

SENÃO SE *não vai para o FGA*

– “vou esperar o próximo”

Putz...só consigo vou pegar o ônibus se tiver como pagar! :/

SE *vai para o FGA*

SE *não está lotado*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

SE *não estou atrasado*

– “vou esperar o próximo”

SENÃO SE *estou atrasado*

– “vou pegar o ônibus”

SENÃO SE *não vai para o FGA*

– “vou esperar o próximo”

Putz...só consigo vou pegar o ônibus se tiver como pagar! :/

SE *vai para o FGA*

SE *não está lotado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

SE *não estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

SENÃO SE *estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *não vai para o FGA*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

É...*não tenho como pagar :'*(...*então vou pra casa :D*

SE *vai para o FGA*

SE *não está lotado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *está lotado*

SE *não estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

SENÃO SE *estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *não vai para o FGA*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

É...não tenho como pagar :'(...então vou pra casa :D

SE vai para o FGA

SE não está lotado

SE tenho dinheiro ou tenho cartão

– “vou pegar o ônibus”

SENÃO SE não (tenho dinheiro ou tenho cartão)

– “vou para casa”

SENÃO SE está lotado

SE não estou atrasado

SE tenho dinheiro ou tenho cartão

– “vou esperar o próximo”

SENÃO SE não (tenho dinheiro ou tenho cartão)

– “vou para casa”

SENÃO SE estou atrasado

SE tenho dinheiro ou tenho cartão

– “vou pegar o ônibus”

SENÃO SE não (tenho dinheiro ou tenho cartão)

– “vou para casa”

SENÃO SE não vai para o FGA

SE tenho dinheiro ou tenho cartão

– “vou esperar o próximo”

SENÃO SE não (tenho dinheiro ou tenho cartão)

– “vou para casa”

Tem como simplificar?

SE *vai para o FGA*

SE *não está lotado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *não (tenho dinheiro ou tenho cartão)*

– “vou para casa”

SENÃO SE *está lotado*

SE *não estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

SENÃO SE *não (tenho dinheiro ou tenho cartão)*

– “vou para casa”

SENÃO SE *estou atrasado*

SE *tenho dinheiro ou tenho cartão*

– “vou pegar o ônibus”

SENÃO SE *não (tenho dinheiro ou tenho cartão)*

– “vou para casa”

SENÃO SE *não vai para o FGA*

SE *tenho dinheiro ou tenho cartão*

– “vou esperar o próximo”

SENÃO SE *não (tenho dinheiro ou tenho cartão)*

– “vou para casa”

Controle de fluxo : exemplo

- Simplificar/otimizar: juntar as condições que resultam nas mesmas decisões
- Proposições: condição Verdadeira ou Falsa
 - `int F; //1: vai para o FGA; 0: não vai para o FGA`
 - `int L; //1: está lotado`
 - `int A; //1: estou atrasado`
 - `int D; //1: tenho dinheiro`
 - `int C; //1: tenho cartão`
- Decisões:
 - “vou pegar o ônibus”
 - “vou esperar o próximo”
 - “vou para casa”

```
if(F) {  
    if(!L) {  
        if(D || C) {  
            printf("vou pegar o ônibus\n");  
        } else if(!(D||C)) {  
            printf("vou para casa\n");  
        }  
    }  
}
```



```
if(F) {  
    if(!L) {  
        if(D || C) {  
            printf("vou pegar o ônibus\n");  
        } else if(!(D||C)) {  
            printf("vou para casa\n");  
        }  
    } else if(L) {  
        if(!A) {  
            if(D || C) {  
                printf("vou esperar o próximo\n");  
            } else if(!(D||C)) {  
                printf("vou para casa\n");  
            }  
        }  
    }  
}
```

```
if(F) {
    if(!L) {
        if(D || C) {
            printf("vou pegar o ônibus\n");
        } else if(!(D||C)) {
            printf("vou para casa\n");
        }
    } else if(L) {
        if(!A) {
            if(D || C) {
                printf("vou esperar o próximo\n");
            } else if(!(D||C)) {
                printf("vou para casa\n");
            }
        } else if(A) {
            if(D || C) {
                printf("vou pegar o ônibus\n");
            } else if(!(D||C)) {
                printf("vou para casa\n");
            }
        }
    }
}
```

```

if(F) {
    if(!L) {
        if(D || C) {
            printf("vou pegar o ônibus\n");
        } else if(!(D||C)) {
            printf("vou para casa\n");
        }
    } else if(L) {
        if(!A) {
            if(D || C) {
                printf("vou esperar o próximo\n");
            } else if(!(D||C)) {
                printf("vou para casa\n");
            }
        } else if(A) {
            if(D || C) {
                printf("vou pegar o ônibus\n");
            } else if(!(D||C)) {
                printf("vou para casa\n");
            }
        }
    }
} else if(!F) {
    if(D || C) {
        printf("vou esperar o próximo\n");
    } else if(!(D||C)) {
        printf("vou para casa\n");
    }
}
}

```

Decisão: "vou pegar o ônibus"

```
if(F) {
    if(!L) {
        if(D || C)
            printf("vou pegar o ônibus\n");
        else if(!(D||C))
            printf("vou para casa\n");
    }
    else if(L) {
        if(!A) {
            if(D || C)
                printf("vou esperar o próximo\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
        else if(A) {
            if(D || C)
                printf("vou pegar o ônibus\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
    }
}
else if(!F) {
    if(D || C)
        printf("vou esperar o próximo\n");
    else if(!(D||C))
        printf("vou para casa\n");
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L) {  
        if(D || C)  
            printf("vou pegar o ônibus\n");  
  
    }  
    else if(L) {  
  
        else if(A) {  
            if(D || C)  
                printf("vou pegar o ônibus\n");  
  
        }  
    }  
  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L ? (D || C)) {  
        printf("vou pegar o ônibus\n");  
    }  
    else if(L) {  
  
        if(A ? (D || C)) {  
            printf("vou pegar o ônibus\n");  
        }  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L && (D || C)) {  
        printf("vou pegar o ônibus\n");  
    }  
    else if(L) {  
  
        if(A && (D || C)) {  
            printf("vou pegar o ônibus\n");  
        }  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L && (D || C)) {  
  
        printf("vou pegar o ônibus\n");  
  
    }  
    else if(L ? A && (D || C)) {  
  
  
  
        printf("vou pegar o ônibus\n");  
  
    }  
  
}
```


Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L && (D || C)) {  
  
        printf("vou pegar o ônibus\n");  
  
    }  
    else if(L && A && (D || C)) {  
  
  
  
        printf("vou pegar o ônibus\n");  
  
  
  
    }  
  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if((!L && (D || C)) ? (L && A && (D || C))) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if((!L && (D || C)) || (L && A && (D || C))) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if((!L && (D || C)) || (L && A && (D || C))) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if((!L || (L && A)) && (D || C)) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if((!L || (L && A)) && (D || C)) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “vou pegar o ônibus”

```
if(F) {  
    if(!L || A) && (D || C) {  
        printf("vou pegar o ônibus\n");  
    }  
}
```

Decisão: “*vou pegar o ônibus*”

```
if(F ? (!L || A) && (D || C)) {  
  
    printf("vou pegar o ônibus\n");  
  
}
```


Decisão: “*vou pegar o ônibus*”

```
if(F && (!L || A) && (D || C)) {  
  
    printf("vou pegar o ônibus\n");  
  
}
```

Decisão: "vou esperar o próximo"

```
if(F) {
    if(!L) {
        if(D || C)
            printf("vou pegar o ônibus\n");
        else if(!(D||C))
            printf("vou para casa\n");
    }
    else if(L) {
        if(!A) {
            if(D || C)
                printf("vou esperar o próximo\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
        else if(A) {
            if(D || C)
                printf("vou pegar o ônibus\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
    }
}
else if(!F) {
    if(D || C)
        printf("vou esperar o próximo\n");
    else if(!(D||C))
        printf("vou para casa\n");
}
```


Decisão: “vou esperar o próximo”

```
if(F) {  
  
    if(L ? !A && (D || C)) {  
  
        printf("vou esperar o próximo\n");  
  
    }  
}  
else if(!F && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if(F) {  
  
    if(L && !A && (D || C)) {  
  
        printf("vou esperar o próximo\n");  
  
    }  
}  
else if(!F && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if(F ? L && !A && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}  
else if(!F && (D || C)) {  
    printf("vou esperar o próximo\n");  
}
```


Decisão: “vou esperar o próximo”

```
if (F && L && !A && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}  
else if (!F && (D || C)) {  
    printf("vou esperar o próximo\n");  
}
```

Decisão: “vou esperar o próximo”

```
if (F && L && !A && (D || C) ? !F && (D || C)) {
```

```
    printf("vou esperar o próximo\n");
```

```
}
```

Decisão: “vou esperar o próximo”

```
if (F && L && !A && (D || C) || !F && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if (F && L && !A && (D || C) || !F && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if((F && L && !A || !F) && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if((F && L && !A || !F) && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: “vou esperar o próximo”

```
if((L && !A || !F) && (D || C)) {  
  
    printf("vou esperar o próximo\n");  
  
}
```

Decisão: "vou para casa"

```
if(F) {
    if(!L) {
        if(D || C)
            printf("vou pegar o ônibus\n");
        else if(!(D||C))
            printf("vou para casa\n");
    }
    else if(L) {
        if(!A) {
            if(D || C)
                printf("vou esperar o próximo\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
        else if(A) {
            if(D || C)
                printf("vou pegar o ônibus\n");
            else if(!(D||C))
                printf("vou para casa\n");
        }
    }
}
else if(!F) {
    if(D || C)
        printf("vou esperar o próximo\n");
    else if(!(D||C))
        printf("vou para casa\n");
}
```


Decisão: "vou para casa"

```
if(F) {
    if(!L) {

        else if(!(D||C))
            printf("vou para casa\n");
    }
    else if(L) {
        if(!A) {

            else if(!(D||C))
                printf("vou para casa\n");
        }
        else if(A) {

            else if(!(D||C))
                printf("vou para casa\n");
        }
    }
}
else if(!F) {

    else if(!(D||C))
        printf("vou para casa\n");
}
```

Decisão: "vou para casa"

```
if(F) {
    if(!L ? !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!A ? !(D||C)) {

            printf("vou para casa\n");
        }
        else if(A ? !(D||C)) {

            printf("vou para casa\n");
        }
    }
}
else if(!F ? !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: "vou para casa"

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!A && !(D||C)) {

            printf("vou para casa\n");
        }
        else if(A && !(D||C)) {

            printf("vou para casa\n");
        }
    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: “*vou para casa*”

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!A && !(D||C) ? A && !(D||C)) {

            printf("vou para casa\n");

        }
    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: “*vou para casa*”

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!A && !(D||C) || A && !(D||C)) {

            printf("vou para casa\n");

        }
    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: "vou para casa"

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!A && !(D||C) || A && !(D||C)) {

            printf("vou para casa\n");

        }
    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: “vou para casa”

```
if(F) {  
    if(!L && !(D||C)) {  
  
        printf("vou para casa\n");  
    }  
    else if(L) {  
        if((!A || A) && !(D||C)) {  
  
            printf("vou para casa\n");  
  
        }  
    }  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “vou para casa”

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L) {
        if(!(D||C)) {

            printf("vou para casa\n");

        }
    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```


Decisão: “*vou para casa*”

```
if(F) {  
    if(!L && !(D||C)) {  
  
        printf("vou para casa\n");  
    }  
    else if(L ? !(D||C)) {  
  
        printf("vou para casa\n");  
  
    }  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “vou para casa”

```
if(F) {
    if(!L && !(D||C)) {

        printf("vou para casa\n");
    }
    else if(L && !(D||C)) {

        printf("vou para casa\n");

    }
}
else if(!F && !(D||C)) {

    printf("vou para casa\n");
}
```

Decisão: “vou para casa”

```
if(F) {  
    if(!L && !(D||C) ? L && !(D||C)) {  
  
        printf("vou para casa\n");  
    }  
  
}  
  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “*vou para casa*”

```
if(F) {  
    if(!L && !(D||C) || L && !(D||C)) {  
  
        printf("vou para casa\n");  
    }  
  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “*vou para casa*”

```
if(F) {  
    if(!(D||C)) {  
  
        printf("vou para casa\n");  
    }  
  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “*vou para casa*”

```
if(F ? !(D||C)) {  
  
    printf("vou para casa\n");  
  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “*vou para casa*”

```
if(F && !(D||C)) {  
  
    printf("vou para casa\n");  
  
}  
else if(!F && !(D||C)) {  
  
    printf("vou para casa\n");  
}
```

Decisão: “*vou para casa*”

```
if (F && !(D||C) ? !F && !(D||C)) {  
  
    printf("vou para casa\n");  
  
}
```


Decisão: “*vou para casa*”

```
if (F && !(D||C) || !F && !(D||C)) {  
  
    printf("vou para casa\n");  
  
}
```

Decisão: “*vou para casa*”

```
if (!(D||C)) {  
  
    printf("vou para casa\n");  
  
}
```

Controle de fluxo : IF + operadores lógicos - otimizar fluxo

- IFs independentes
- Todos seriam verificados

```
1 if(!(D||C))
2 {
3     printf("vou para casa\n");
4 }
```

```
1 if((L && !A || !F) && (D || C))
2 {
3     printf("vou esperar o próximo\n");
4 }
```

```
1 if(F && (!L || A) && (D || C))
2 {
3     printf("vou pegar o ônibus\n");
4 }
```

Controle de fluxo : IF + operadores lógicos - otimizar fluxo

- Como são situações dependentes, podemos juntar

```
1 if(!(D||C))
2 {
3     printf("vou para casa\n");
4 }
5 else if((L && !A || !F) && (D || C))
6 {
7     printf("vou esperar o próximo\n");
8 }
9 else if(F && (!L || A) && (D || C))
10 {
11     printf("vou pegar o ônibus\n");
12 }
```

Controle de fluxo : IF + operadores lógicos - otimizar fluxo

- Como são situações dependentes, podemos juntar

```
1 if(!(D||C))
2 {
3     printf("vou para casa\n");
4 }
5 else if(L && !A || !F)
6 {
7     printf("vou esperar o próximo\n");
8 }
9 else if(F && (!L || A))
10 {
11     printf("vou pegar o ônibus\n");
12 }
```

Controle de fluxo : IF + operadores lógicos - otimizar fluxo

- Como são situações dependentes, podemos juntar

```
1 if(!(D||C)) //!D && !C
2 {
3     printf("vou para casa\n");
4 }
5 else if(L && !A || !F)
6 {
7     printf("vou esperar o próximo\n");
8 }
9 else //if(F && (!L || A))
10 {
11     printf("vou pegar o ônibus\n");
12 }
```

- Negação da conjunção : $!(A \&\& B) \equiv (!A \vee !B)$
- Negação da disjunção : $!(A \vee B) \equiv (!A \&\& !B)$
- A verificação `if(F && (!L || A))` pode ser omitida pois é equivalente a negação da condição anterior
- $!(!F \vee L \&\& !A) \equiv !!F \&\& !(L \&\& !A)$
 $\equiv F \&\& !L \vee !!A$
 $\equiv F \&\& !L \vee A$

Vamos praticar

Elabore um algoritmo que lê um número que representa uma senha, verifica se a senha está correta, comparando-a com 34567, e informa “Acesso autorizado” caso esteja correta.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5
```

Vamos praticar

Elabore um algoritmo que lê um número que representa uma senha, verifica se a senha está correta, comparando-a com 1234, e informa “Acesso autorizado” caso esteja correta.

```
1 #include <stdio.h>
2
3 int main(){
4     int senha;
5     scanf("%d", &senha);
6
7     if(senha == 1234)
8     {
9         printf("Acesso autorizado\n");
10    }
11
12    return 0;
13 }
14
```


Vamos praticar

Faça um programa que leia a nota e a frequência e imprima "aprovado" caso a nota seja maior ou igual a 6 e a frequência seja maior ou igual a 75.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if (condicoes)
2  {
3      ...
4  }
5
```

Vamos praticar

Faça um programa que leia a nota e a frequência e imprima “aprovado” caso a nota seja maior ou igual a 6 e a frequência seja maior ou igual a 75.

```
1 #include <stdio.h>
2
3 int main(){
4     int nota, freq;
5     scanf("%d%d", &nota, &freq);
6
7     if(nota >= 6 && freq >= 75)
8     {
9         printf("aprovado\n");
10    }
11
12    return 0;
13 }
14
```

Vamos praticar

Escreva um algoritmo que leia um número e verifique se ele se encontra no intervalo entre 5 e 20. Em caso afirmativo, imprima na tela a mensagem “N esta no intervalo entre 5 e 20”, sendo N o número lido.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5
```

Vamos praticar

Escreva um algoritmo que leia um número e verifique se ele se encontra no intervalo entre 5 e 20. Em caso afirmativo, imprima na tela a mensagem “N esta no intervalo entre 5 e 20”, sendo N o número lido.

```
1 #include <stdio.h>
2
3 int main(){
4     int num;
5     scanf("%d", &num);
6
7     //Errado: 5 <= num <= 20
8     //Seja num = 21: (5 <= 21) <= 20: 1 <= 20: 1 (errado! 21 <=
9     //20)
10    if(num >= 5 && num <= 20)
11    {
12        printf("%d esta no intervalo entre 5 e 20\n", num);
13    }
14
15    return 0;
16 }
```

Vamos praticar

Escreva um programa que dado uma hora, devolva a próxima hora. Considere a hora de 1 - 24 horas.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else
6  {
7      ...
8  }
9
```

Vamos praticar

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int hora;
6
7     scanf("%d", &hora);
8     if ( hora < 24 )
9     {
10        printf("A proxima hora eh %d\n", hora+1);
11    }
12    else
13    {
14        printf("A proxima hora eh %d\n", 1);
15    }
16
17    return 0;
18 }
19
20
```

Vamos praticar

Escreva um programa que leia um inteiro, imprima **i: par**, caso o número lido seja par, ou imprima **i: impar** caso seja o número lido seja ímpar. Substitua **i** pelo inteiro lido.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else
6  {
7      ...
8  }
9
```

Vamos praticar

Vogal ou consoante

```
1 #include <stdio.h>
2 int main(){
3     int i;
4     scanf("%d",&i);
5
6     if(i%2==0)
7     {
8         printf("%d:par\n",i);
9     }
10    else
11    {
12        printf("%d:impar\n",i);
13    }
14    return 0;
15 }
16
```


Vamos praticar

Escreva um programa que leia um caractere, imprima **c: vogal**, sendo **c** o caractere lido e caso seja uma vogal, ou imprima **c: consoante**, sendo **c** o caractere lido e caso seja uma consoante.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else
6  {
7      ...
8  }
9
```

Vamos praticar

Vogal ou consoante

```
1 #include <stdio.h>
2 int main(){
3     char c;
4     scanf("%c",&c);
5
6     if(c=='a' || c=='A' || c=='e' || c=='E' ||
7         c=='i' || c=='I' || c=='o' || c=='O' ||
8         c=='u' || c=='U')
9     {
10        printf("%c:vogal\n",c);
11    }
12    else
13    {
14        printf("%c:consoante\n",c);
15    }
16    return 0;
17 }
18
```

Vamos praticar

Leia 2 valores inteiros (A e B). Após, o programa deve mostrar uma mensagem **Sao Multiplos** ou **Nao sao Multiplos**, indicando se os valores lidos são múltiplos entre si.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else
6  {
7      ...
8  }
9
```

Vamos praticar

Múltiplos

```
1 #include <stdio.h>
2 int main(){
3     int a, b;
4     scanf("%d %d",&a, &b);
5
6     if((a > b && a%b==0) || (a < b && b%a==0))
7     {
8         printf("Sao Multiplos\n");
9     }
10    else
11    {
12        printf("Nao sao Multiplos\n");
13    }
14    return 0;
15 }
16
```

Vamos praticar

Leia 4 valores inteiros A, B, C e D. A seguir, se B for maior do que C e se D for maior do que A, e a soma de C com D for maior que a soma de A e B e se C e D, ambos, forem positivos e se a variável A for par escrever a mensagem “Valores aceitos”, senão escrever “Valores nao aceitos”.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else
6  {
7      ...
8  }
9
```

Vamos praticar

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int A, B, C, D;
6     scanf("%d%d%d%d", &A, &B, &C, &D);
7
8     if( (B > C) && (D > A) &&
9         ((C + D) > (A + B)) &&
10        (C >= 0) &&
11        (D >= 0) &&
12        (A%2 == 0) )
13     {
14         printf("Valores aceitos\n");
15     }
16     else
17     {
18         printf("Valores nao aceitos\n");
19     }
20
21     return 0;
22 }
23
```

Vamos praticar

Escreva um algoritmo que receba um inteiro correspondente a um DDD e imprima a cidade correspondente. Sabendo que: 61 Brasilia, 71 Salvador, 11 Sao Paulo.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else if(condicoes)
6  {
7      ...
8  }
9
```

Vamos praticar

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int ddd;
6     scanf("%d", &ddd);
7
8     if( ddd == 61 )
9     {
10        printf("Brasilia\n");
11    }
12    else if( ddd == 71 )
13    {
14        printf("Salvador\n");
15    }
16    else if( ddd == 11 )
17    {
18        printf("Sao Paulo\n");
19    }
20    return 0;
21 }
22
```


Vamos praticar

Faça um algoritmo que leia uma letra que pode ser 'F' ou 'J' e mostra a mensagem "pessoa física", "pessoa jurídica" ou "tipo de pessoa invalido", conforme o caso.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  {
3      ...
4  }
5  else if(condicoes)
6  {
7      ...
8  }
9  else
10 {
11     ...
12 }
```

Vamos praticar

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char pessoa;
6     scanf("%c", &pessoa);
7
8     if( pessoa=='F' )
9     {
10        printf("pessoa fisica\n");
11    }
12    else if( pessoa=='J' )
13    {
14        printf("pessoa juridica\n");
15    }
16    else
17    {
18        printf("tipo de pessoa invalido\n");
19    }
20    return 0;
21 }
22
```

Vamos praticar

Você deve fazer um programa que leia um valor real e apresente uma mensagem dizendo em qual dos seguintes intervalos ($[0,25]$, $(25,50]$, $(50,75]$) este valor se encontra. Se o valor não estiver em nenhum destes intervalos, deverá ser impressa a mensagem “Fora de intervalo”.

1. ()
2. ! (inverso)
3. * / % (resto)
4. + -
5. < > <= >=
6. == (igual) != (diferente)
7. && (e)
8. || (ou)
9. =

```
1  if(condicoes)
2  { ... }
3  else if(condicoes)
4  { ... }
5  else
6  { ... }
7
```

Vamos praticar

```
1 float A;
2 scanf("%f", &A);
3
4 if(A >= 0 && A <= 25)
5 {
6     printf("Intervalo [0,25]\n");
7 }
8 else if(A > 25 && A <= 50)
9 {
10    printf("Intervalo (25,50]\n");
11 }
12 else if(A > 50 && A <= 75)
13 {
14    printf("Intervalo (50,75]\n");
15 }
16 else
17 {
18    printf("Fora de intervalo\n");
19 }
20
```

Vamos praticar

1. Leia quatro números (N1, N2, N3, N4), correspondente às quatro notas de um aluno.
2. Calcule a média com pesos 2, 3, 4 e 1, respectivamente, para cada uma destas notas e mostre esta média acompanhada pela mensagem "Media: ".
3. Se esta média for maior ou igual a 7.0, imprima a mensagem "Aluno aprovado.".
4. Se a média calculada for inferior a 5.0, imprima a mensagem "Aluno reprovado.".
5. Se a média calculada for um valor entre 5.0 e 6.9, inclusive estas, o programa deve imprimir a mensagem "Aluno em exame.".
 - a. No caso do aluno estar em exame, leia um valor correspondente à nota do exame obtida pelo aluno.
 - b. Recalcule a média (some a pontuação do exame com a média anteriormente calculada e divida por 2).
 - c. Imprima a mensagem "Aluno aprovado."(caso a média final seja 5.0 ou mais) ou "Aluno reprovado.", (caso a média tenha ficado 4.9 ou menos).
 - d. Apresente na última linha uma mensagem "Media final: "seguido da média final para esse aluno.

Vamos praticar

```
1 float n1, n2, n3, n4, media;
2 scanf("%f%f%f%f", &n1, &n2, &n3, &n4);
3
4 media = (n1*2 + n2*3 + n3*4 + n4)/10;
5 printf("Media: %.1f\n", media);
6
7 if(media >= 7)
8     printf("Aluno aprovado.\n");
9 else if(media < 5)
10    printf("Aluno reprovado.\n");
11 else if(media >= 5.0 && media <= 6.9)
12 {
13     printf("Aluno em exame.\n");
14     scanf("%f", &n1);
15
16     media = (media + n1)/2;
17     if(m>=5)
18         printf("Aluno aprovado.\n");
19     else
20         printf("Aluno reprovado.\n");
21
22     printf("Media final: %.1f\n", media);
23 }
24
```

Vamos praticar

- Raiz quadrada de um número: use a função **sqrt()**
- Não esqueça de incluir o `math.h`
- Para compilar o código usando o `math.h`
- `gcc nome_arquivo.c -lm`
- Valor absoluto de um número: use a função **abs()**;
- Não esqueça de incluir o `stdlib.h`
- Exercícios do beecrowd (antigo URI):
 - `beecrowd.com.br`
 - 1035, 1036, 1037, 1038, 1040, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1050, 1051, 1052, 1060, 1061 (obs. formato da entrada), 1064, 1065, 1066
- Exercícios no MOJ: em breve