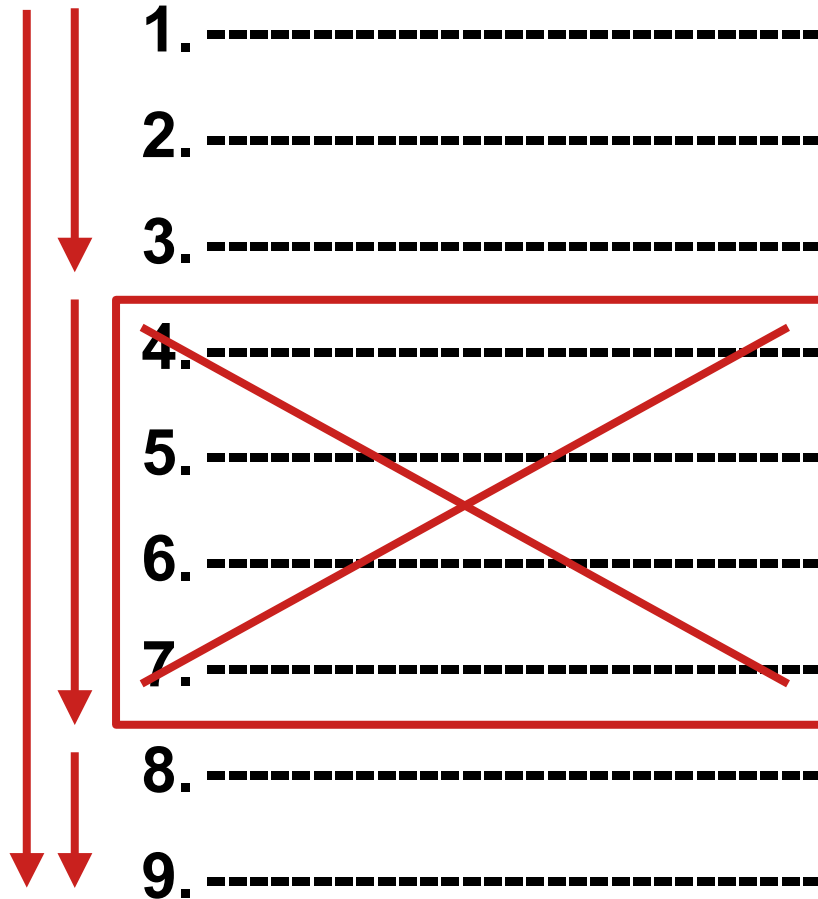




# **Estruturas de Controle de Fluxo**

## **Instrução Condicional (if - else)**

# Estruturas de Controle de Fluxo



# Estruturas de Controle de Fluxo

## 1. Instrução condicional

```
if (<expressão_lógica>)  
{  
    <instrução1>  
}
```

# Estruturas de Controle de Fluxo

## 1. Instrução condicional

if (<expressão\_lógica>)

<instrução1>

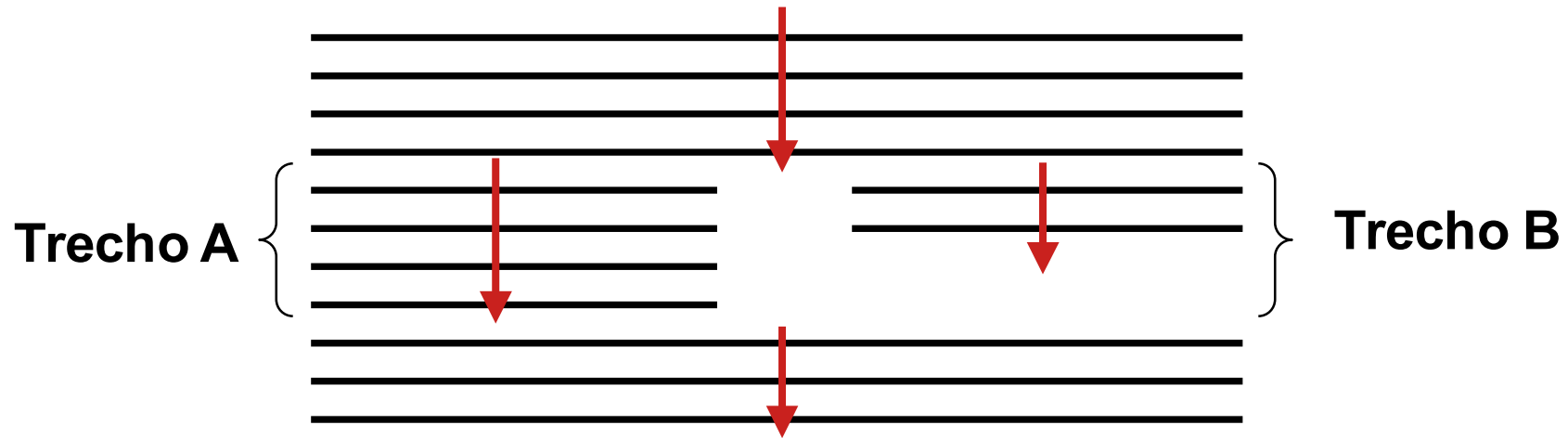


# Estruturas de Controle de Fluxo

## 1. Instrução condicional

```
if (<expressão_lógica>)  
{  
    <instrução1>  
    <instrução2>  
    ...  
}
```

# Estruturas de Controle de Fluxo



# Estruturas de Controle de Fluxo

## 1. Instrução condicional

```
if (<expressão_lógica>)  
{  
    <instrução1>  
}  
else  
{  
    <instrução2>  
}
```

# Estruturas de Controle de Fluxo

## 1. Instrução condicional

if (<expressão\_lógica>)

<instrução1>

else

<instrução2>

# Estruturas de Controle de Fluxo

```
if (<expressão_lógica> )
{
    <instrução1>
    ...
    <instruçãon>
}
else
{
    <instrução1>
    ...
    <instruçãom>
}
```

# Estruturas de Controle de Fluxo

```
#include <stdio.h>
int main()
{
    float n1, n2, n3, n4, media;
    printf ("Entre com a primeira nota do aluno: ");
    scanf ("%f",&n1);
    printf ("\nEntre com a segunda nota do aluno: ");
    scanf ("%f",&n2);
    printf ("\nEntre com a terceira nota do aluno: ");
    scanf ("%f",&n3);
    printf ("\nEntre com a quarta nota do aluno: ");
    scanf ("%f",&n4);
    media=(n1+n2+n3+n4)/4;
    if (media>=7)
        printf ("\nAluno aprovado!");
    else
        printf ("\nAluno reprovado!");
}
```

Exemplo

# Estruturas de Controle de Fluxo

## 1. Instrução condicional

### Exercício 1:

Construa um programa, na linguagem C, que classifique uma pessoa em uma determinada faixa etária. O programa recebe através do teclado a idade do usuário, caso esta seja inferior a 12 anos é classificado como criança, se a idade for de 12 a 18 anos é classificado como adolescente e se for superior a 18 anos é classificado como adulto. O programa retornará no monitor o resultado da classificação.

```
#include <stdio.h>
int main()
{
    int idade;
    printf ("Entre com a sua idade: ");
    scanf ("%d", &idade);
    if (idade<12)
        printf ("\nVoce eh uma criança.\n");
    if (idade>=12 && idade<=18)
        printf ("\nVoce eh um adolescente.\n");
    if (idade>18)
        printf ("\nVoce eh um adulto.\n");
}
```



```
#include <stdio.h>
int main()
{
    int idade;
    printf ("Entre com a sua idade: ");
    scanf ("%d", &idade);
    if (idade<12)
        printf ("\nVoce eh uma criança.\n");
    else
        if (idade>=12 && idade<=18)
            printf ("\nVoce eh um adolescente.\n");
        else
            printf ("\nVoce eh um adulto.\n");
}
```

# Estruturas de Controle de Fluxo


## 1. Instrução condicional (continuação)

### Exercício 2:

Construa um programa em C que receba através da entrada padrão um valor inteiro pertencente ao intervalo  $[0,255]$  e a posição do bit que se deseja verificar se está ativo, o programa deve indicar na saída padrão o resultado desta verificação.

```
#include <stdio.h>
int main ()
{
    unsigned char desloca, byte, aux=1;
    printf("\nDigite um numero pertencente ao intervalo [0,255]: ");
    scanf("%hhd", &byte);
    printf("\nDigite a posicao do bit a testar entre 0 e 7: ");
    scanf("%hhd", &desloca);
    aux = aux << desloca;
    byte = byte & aux;
byte = byte >> desloca;
    if (byte = 1)
        printf("\n 0 bit testado esta ativo.\n");
    else
        printf("\n 0 bit testado nao esta ativo.\n");
}
```

```
#include <stdio.h>
int main ()
{
    unsigned char desloca, byte, aux=1;
    printf("\nDigite um numero pertencente ao intervalo [0,255]: ");
    scanf("%hhd", &byte);
    printf("\nDigite a posicao do bit a testar entre 0 e 7: ");
    scanf("%hhd", &desloca);
    aux = aux << desloca;
    byte = byte & aux;
    if (byte)
        printf("\n 0 bit testado esta ativo.\n");
    else
        printf("\n 0 bit testado nao esta ativo.\n");
}
```

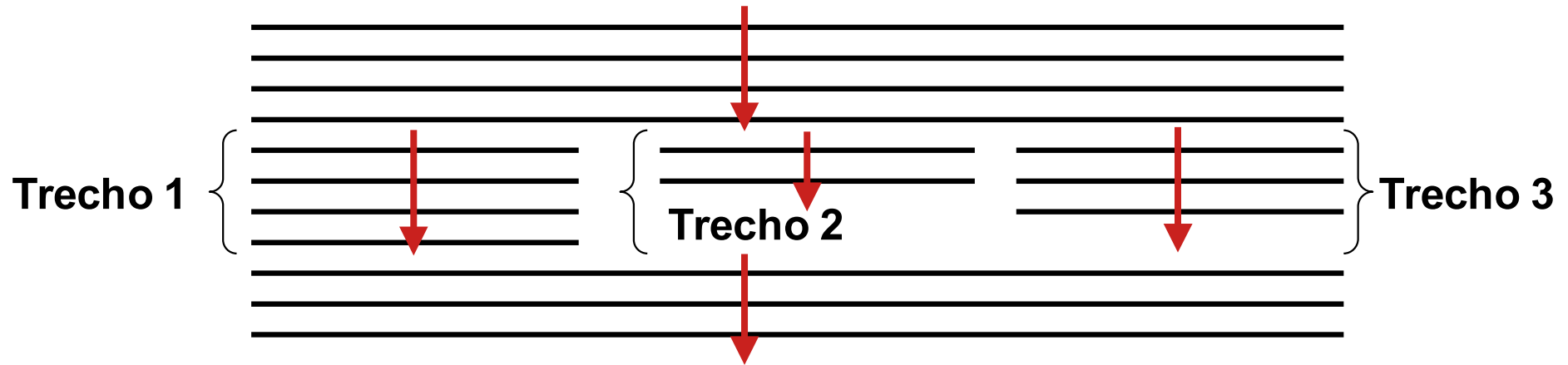


# **Estruturas de Controle de Fluxo**

## **Comando de Seleção Múltipla (switch)**

# Estruturas de Controle de Fluxo

## 2. Comando de seleção múltipla



# Estruturas de Controle de Fluxo

## 2. Comando de seleção múltipla

```
switch (<identificador_variável_escalar>
{
    case <constante1> : <instrução1> ou <instruções1>
                        break;
    case <constante2> : <instrução2> ou <instruções2>
                        break;
    . . .
    case <constanten> : <instruçãon> ou <instruçõesn>
                        break;
    default:
                        <instruçãon+1> ou <instruçõesn+1>
}
```

# Estruturas de Controle de Fluxo

## 2. Comando de seleção múltipla

Exemplo:

O programa a seguir ao receber um número inteiro pertencente ao intervalo [1, 12] retorna o nome do mês correspondente ao valor fornecido.



```
#include <stdio.h>
int main()
{
    int mes;
    printf("\nEntre com um valor inteiro de 1 a 12 : ");
    scanf("%d",&mes);
    switch (mes)
    {
        case 1: printf ("0 %dº mes do calendario eh Janeiro", mes);
                break;
        case 2: printf ("0 %dº mes do calendario eh Fevereiro", mes);
                break;
        case 3: printf ("0 %dº mes do calendario eh Marco", mes);
                break;
        case 4: printf ("0 %dº mes do calendario eh Abril ", mes);
                break;
        case 5: printf ("0 %dº mes do calendario eh Maio", mes);
                break;
    }
}
```

```
case 6: printf ("0 %dº mes do calendario eh Junho", mes);  
        break;  
case 7: printf ("0 %dº mes do calendario eh Julho", mes);  
        break;  
case 8: printf ("0 %dº mes do calendario eh Agosto", mes);  
        break;  
case 9: printf ("0 %dº mes do calendario eh Setembro", mes);  
        break;  
case 10: printf ("0 %dº mes do calendario eh Outubro", mes);  
         break;  
case 11: printf ("0 %dº mes do calendario eh Novembro", mes);  
         break;  
case 12: printf ("0 %dº mes do calendario eh Dezembro", mes);  
         break;  
default: printf ("Nao existe um mes que corresponda a %s",  
                 "este valor.");  
}  
}
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a,b,c;
```

```
    printf("\nEntre com o valor de a : ");
```

```
    scanf("%d",&a);
```

```
    switch (a)
```

```
    {
```

```
        case 1:
```

```
            b=2;
```

```
            break;
```

```
        case 2:
```

```
        {
```

```
            c=3;
```

```
            b=a*c;
```

```
        }
```

```
            break;
```

```
        case 3 :
```

```
            c=a;
```

```
        default:
```

```
            b=500;
```

```
    }
```

```
    printf("\nvalor de a = %d    valor de b = %d    valor de c = %d\n",a,b,c);
```

```
}
```

# Exercício 3

**Determine:**

**Saídas para a==1**

valor de a = 1 valor de b = 2 valor de c = ?

**Saídas para a==2**

valor de a = 2 valor de b = 6 valor de c = 3

**Saídas para a==3**

valor de a = 3 valor de b = 500 valor de c = 3

**Saídas para a==11**

valor de a = 11 valor de b = 500 valor de c = ?

# Estruturas de Controle de Fluxo


## 2. Comando de seleção múltipla

**Exercício 4:** Construa um programa, na linguagem C, que receba dois inteiros e utilizando a estrutura de controle de fluxo *switch*, ofereça ao usuário um menu com as opções de soma, subtração, multiplicação e divisão. Após o usuário fazer sua opção o programa deve efetuar a operação aritmética selecionada tendo como operandos os valores fornecidos, apresentando o resultado na saída padrão.



```
#include <stdlib.h>
#include <stdio.h>
int main()
{
    int opcao,n1,n2,resultado;
    printf("\nEntre com o primeiro valor inteiro: ");
    scanf("%d",&n1);
    printf("\nEntre com o segundo valor inteiro: ");
    scanf("%d",&n2);
    printf("Digite:\n1 - para somar os inteiros\n2 - para subtrair os inteiros\n");
    printf("3 - para multiplicar os inteiros\n4 - para dividir os inteiros\n0opcao: ");
    scanf("%d",&opcao);
```

```
switch (opcao)
{
    case 1 : resultado=n1+n2;
            break;
    case 2 : resultado=n1-n2;
            break;
    case 3 : resultado=n1*n2;
            break;
    case 4 : resultado=n1/n2;
            break;
    default: printf("Opcao invalida!");
            exit(1);
}
printf("\n0 resultado da operacao foi %d\n",resultado);
}
```



#include <stdlib.h>

```
switch (opcao)
{
    case 1 : resultado=n1+n2;
            break;
    case 2 : resultado=n1-n2;
            break;
    case 3 : resultado=n1*n2;
            break;
    case 4 : if (n2)
                resultado=n1/n2;
            else
            {
                printf("Divisao por zero!");
                exit(2);
            }
            break;
    default: printf("Opcao invalida!");
            exit(1);
}
printf("\n0 resultado da operacao foi %d\n",resultado);
}
```



```
#include <stdlib.h>
#include <stdio.h>

int main() {
    int n1,n2,resultado;
    char opcao;
    printf("\nEntre com o primeiro valor inteiro: ");
    scanf("%d",&n1);
    printf("\nEntre com o segundo valor inteiro: ");
    scanf("%d",&n2);
    printf("Digite:\n"+"+" para somar os inteiros\n\n"-" para subtrair os inteiros\n\n"* para multiplicar os inteiros\n\n");
    printf("\n"/ para dividir os inteiros\n\n0pcao: ");
    scanf("%c",&opcao);
```

```
switch (opcao) {
    case '+' : resultado=n1+n2;
               break;
    case '-' : resultado=n1-n2;
               break;
    case '*' : resultado=n1*n2;
               break;
    case '/' : if (n2)
                resultado=n1/n2;
               else {
                   printf("Divisao por zero!");
                   exit(2);
               }
               break;
    default :  printf("Opcao invalida!");
               exit(1);
}
printf("\n0 resultado da operacao foi %d\n",resultado);
}
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    float f;
    char c;
    printf ("Entre com um real: ");
    scanf ("%f",&f);
    printf ("Entre com um caractere: ");
    fflush (stdin); /*para limpar o buffer*/
    scanf ("%c",&c);
    printf("\n|%c|%d\n",c,c);
}
```

**Não deve ser utilizado!  
Por quê?**

**Solução adequada!**

```
#include <stdio.h>
int main()
{
    float f;
    char c;
    printf ("Entre com um real: ");
    scanf ("%f",&f);
    printf ("Entre com um caractere: ");
    setbuf (stdin, NULL); /*para limpar o buffer*/
    scanf ("%c",&c);
    printf("\n|%c|%d\n",c,c);
}
```

```
#include <stdlib.h>
#include <stdio.h>

int main()
{
    int n1,n2,resultado;
    char opcao;
    printf("\nEntre com o primeiro valor inteiro: ");
    scanf("%d",&n1);
    printf("\nEntre com o segundo valor inteiro: ");
    scanf("%d",&n2);
    printf("Digite:\n"+"+" - para somar os inteiros\n\n"-\" para subtrair os inteiros\n");
    printf("\n"*\" para multiplicar os inteiros\n\n"/\" para dividir os inteiros\n\n0pcao: ");
    setbuf (stdin, NULL);
    scanf("%c",&opcao);
    . . .
}
```



# **Estruturas de Controle de Fluxo**

## **Laços de Repetição (while)**

# Estruturas de Controle de Fluxo

Para facilitar a compreensão/utilização das próximas estruturas de controle de fluxo a serem estudadas, vamos analisar o exemplo a seguir.

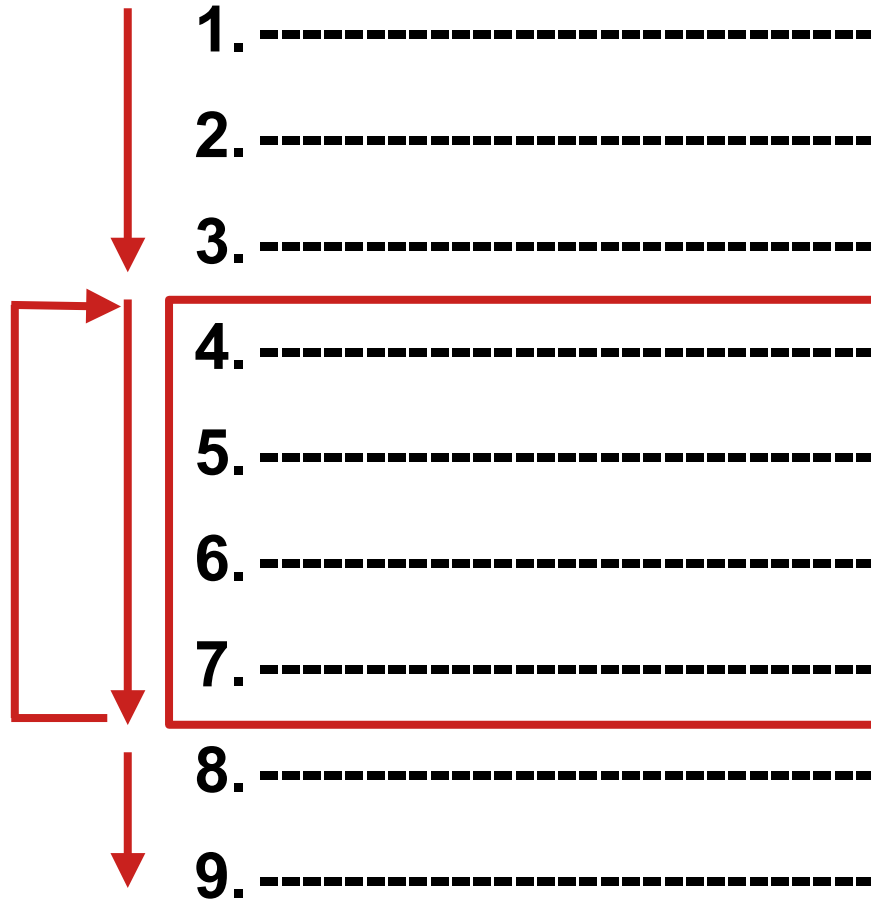
Em uma determinada disciplina foram aplicadas duas avaliações com notas variando de zero a dez, e a nota final foi calculada com base na média aritmética entre as notas obtidas pelo aluno. O programa a seguir, escrito na linguagem C, recebe as notas de quatro alunos e determina qual foi a maior nota final obtida.

```
#include <stdio.h>
int main() {
    float n1A1, n2A1, n1A2, n2A2, n1A3, n2A3, n1A4, n2A4,
    media, maiorMedia;
    printf ("Digite a 1ª nota do 1º aluno entre 0 e 10: ");
    scanf ("%f",&n1A1);
    printf ("Digite a 2ª nota do 1º aluno entre 0 e 10: ");
    scanf ("%f",&n2A1);
    maiorMedia = media = (n1A1 + n2A1)/2;
    printf ("Digite a 1ª nota do 2º aluno entre 0 e 10: ");
    scanf ("%f",&n1A2);
    printf ("Digite a 2ª nota do 2º aluno entre 0 e 10: ");
    scanf ("%f",&n2A2);
    media = (n1A2 + n2A2)/2;
    if (maiorMedia<media)
    {
        maiorMedia=media;
    }
}
```



```
printf ("Digite a 1ª nota do 3º aluno entre 0 e 10: ");
scanf ("%f",&n1A3);
printf ("Digite a 2ª nota do 3º aluno entre 0 e 10: ");
scanf ("%f",&n2A3);
media = (n1A3 + n2A3)/2;
if (maiorMedia<media)
    maiorMedia=media;
printf ("Digite a 1ª nota do 4º aluno entre 0 e 10: ");
scanf ("%f",&n1A4);
printf ("Digite a 2ª nota do 4º aluno entre 0 e 10: ");
scanf ("%f",&n2A4);
media = (n1A4 + n2A4)/2;
if (maiorMedia<media)
    maiorMedia=media;
printf("A maior nota final obtida foi %.2f\n", maiorMedia);
}
```

# Estruturas de Controle de Fluxo



# Estruturas de Controle de Fluxo

## 3. Laços de repetição

```
while (<expressão_lógica>)  
{  
    <instrução1>  
    . . .  
    <instruçãon>  
}
```

/\* enquanto o resultado da avaliação da expressão lógica for verdadeiro faça \*/

# Estruturas de Controle de Fluxo

## 3. Laços de repetição

O exemplo visto anteriormente pode ser revisitado e implementado utilizando o laço de repetição *while*, vamos ver o mesmo.

```
#include <stdio.h>
int main()
{
    float n1, n2, media, maiorMedia;
    int contador=1;
    while (contador<=4)
    {
        printf ("Digite a 1ª nota do %dº aluno entre 0 e 10: ", contador);
        scanf ("%f",&n1);
        printf ("Digite a 2ª nota do %dº aluno entre 0 e 10: ", contador);
        scanf ("%f",&n2);
        if (contador==1)
            maiorMedia = (n1 + n2)/2;
        else {
            media = (n1 + n2)/2;
            if (maiorMedia < media)
                maiorMedia = media;
        }
        contador++;
    }
    printf("A maior nota final obtida foi %.2f",maiorMedia);
}
```

# Estruturas de Controle de Fluxo

## 3. Laços de repetição

Talvez algum aluno ainda não tenha se convencido da vantagem de se utilizar estruturas de repetição.

Para evidenciar que às vezes nem ao menos existe alternativa à utilização de estruturas de repetição, resolva o exercício a seguir, que trata-se de uma variação do exemplo anterior.

# Estruturas de Controle de Fluxo

## 3. Laços de repetição

### Exercício 5:

Em uma determinada disciplina foram aplicadas duas avaliações com notas variando de zero a dez, e a nota final de cada aluno foi calculada com base na média aritmética entre as notas obtidas pelo mesmo. Codifique um programa, utilizando a linguagem C, que receba como entrada a quantidade de alunos existentes em uma determinada turma da disciplina em questão e as notas dos alunos. O programa deve exibir na saída padrão a maior nota final obtida na referida turma.