

Professor Pietro Martins de Oliveira

# AL GO RIT MOS

do início ao fim

***PACOTE DE EXERCÍCIOS 3:  
ESTRUTURAS DE DECISÃO EM C***

- 1) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba dois números e mostre o maior deles. Caso eles sejam iguais, deve-se mostrar a mensagem "Os números são iguais".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 1 - Caso de teste 1

```
Insira o primeiro número:
1
Insira o segundo número:
2
O maior é: 2
```

Exemplo de execução – Exercício 1 - Caso de teste 2

```
Insira o primeiro número:
2
Insira o segundo número:
1
O maior é: 2
```

Exemplo de execução – Exercício 1 - Caso de teste 3

```
Insira o primeiro número:
1
Insira o segundo número:
1
Os números são iguais
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 1 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float n1, n2;
04     printf("Insira o primeiro número:\n");
05     scanf("%f", &n1);
06     printf("Insira o segundo número:\n");
07     scanf("%f", &n2);
08     if(n1 > n2){
09         printf("O maior é: %f.\n", n1);
10     }
11     if(n2 > n1){
12         printf("O maior é: %f.\n", n2);
13     }
14     if(n1 == n2){
15         printf("Os números são iguais.\n");
```

```
16     }  
17 }
```

- 2) Desenvolva um algoritmo que receba três números. O algoritmo deve imprimir "Condição satisfeita", na tela, caso o primeiro dado inserido seja maior do que os outros dois (o primeiro não pode ser igual a nenhum). Caso contrário, deve ser impressa a mensagem: "Erro".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

#### Exemplo de execução – Exercício 2 – Caso de teste 1

```
Insira o primeiro número:  
3  
Insira o segundo número:  
2  
Insira o terceiro número:  
1  
Condição satisfeita
```

#### Exemplo de execução – Exercício 2 – Caso de teste 2

```
Insira o primeiro número:  
2  
Insira o segundo número:  
3  
Insira o terceiro número:  
1  
Erro
```

#### Exemplo de execução – Exercício 2 – Caso de teste 3

```
Insira o primeiro número:  
1  
Insira o segundo número:  
2  
Insira o terceiro número:  
3  
Erro
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 2 – Solução

```
01 #include <stdio.h>
02
03 int main(){
04     float n1, n2, n3;
05     printf("Insira o primeiro número:\n");
06     scanf("%f", &n1);
07     printf("Insira o segundo número:\n");
08     scanf("%f", &n2);
09     printf("Insira o terceiro número:\n");
10     scanf("%f", &n3);
11     if((n1 > n2) && (n1 > n3)){
12         printf("Condição satisfeita.\n");
13     } else {
14         printf("Erro.\n");
15     }
16 }
```

- 3) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba um número inteiro e verifique se esse número é par ou ímpar.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, no qual você encontra uma simulação da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 3 - Caso de teste 1

Insira um número:

2

2 é par.

Exemplo de execução – Exercício 3 - Caso de teste 2

Insira um número:

3

3 é ímpar.

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 3 – Solução

```
01 #include <stdio.h>
02
03 int main(){
04     int num, r;
05     printf("Insira um número inteiro:\n");
06     scanf("%d", &num);
```

```
07  
08     r = num % 2;  
09  
10     if(r == 0){  
11         printf("O número %d é par.\n", num);  
12     }  
13     if (r != 0){  
14         printf("O número %d é ímpar.\n", num);  
15     }  
16 }
```

- 4) Desenvolva um algoritmo que receba dois números, calcule e mostre a multiplicação entre eles, se ambos forem iguais. Caso o primeiro seja maior que o segundo, mostre a subtração do primeiro pelo segundo. Caso contrário, mostre a soma entre os dois.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 4 – Caso de teste 1

```
Insira o primeiro valor:  
2  
Insira o segundo valor:  
2  
Multiplicação: 4
```

Exemplo de execução – Exercício 4 – Caso de teste 2

```
Insira o primeiro valor:  
2  
Insira o segundo valor:  
1  
Subtração: 1
```

Exemplo de execução – Exercício 4 – Caso de teste 3

```
Insira o primeiro valor:  
1  
Insira o segundo valor:  
2  
Soma: 3
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 4 – Solução

```
01 #include <stdio.h>
02
03 int main(){
04     float n1, n2;
05
06     printf("Insira o primeiro valor:\n");
07     scanf("%f", &n1);
08     printf("Insira o segundo valor:\n");
09     scanf("%f", &n2);
10
11     if (n1 == n2){
12         printf("Multiplicação: %f.\n", n1*n2);
13     }
14     if (n1 > n2){
15         printf("Subtração: %f.\n", n1-n2);
16     }
17     if (n1 < n2){
18         printf("Soma: %f.\n", n1+n2);
19     }
20 }
```

- 5) Desenvolva um algoritmo que simule uma calculadora. Você deve dar a opção de o usuário escolher entre: 1 - Somar; 2 - Subtrair; 3 - Multiplicar; 4 - Dividir. O usuário só conseguirá processar dois números inteiros por vez.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

#### Exemplo de execução – Exercício 5 - Caso de teste 1

```
Digite 1 para somar;
Digite 2 para subtrair;
Digite 3 para multiplicar;
Digite 4 para dividir;
1
Insira o primeiro valor:
1
Insira o segundo valor:
2
Soma: 3
```

#### Exemplo de execução – Exercício 5 - Caso de teste 2

```
Digite 1 para somar;
Digite 2 para subtrair;
Digite 3 para multiplicar;
Digite 4 para dividir;
2
Insira o primeiro valor:
```



```
2
Insira o segundo valor:
1
Subtração: 1
```

#### Exemplo de execução – Exercício 5 - Caso de teste 3

```
Digite 1 para somar;
Digite 2 para subtrair;
Digite 3 para multiplicar;
Digite 4 para dividir;
3
Insira o primeiro valor:
2
Insira o segundo valor:
2
Multiplicação: 4
```

#### Exemplo de execução – Exercício 5 - Caso de teste 4

```
Digite 1 para somar;
Digite 2 para subtrair;
Digite 3 para multiplicar;
Digite 4 para dividir;
4
Insira o primeiro valor:
4
Insira o segundo valor:
2
Divisão: 2
```

#### Exemplo de execução – Exercício 5 - Caso de teste 5

```
Digite 1 para somar;
Digite 2 para subtrair;
Digite 3 para multiplicar;
Digite 4 para dividir;
4
Insira o primeiro valor:
1
Insira o segundo valor:
0
Divisão por zero.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 5 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float n1, n2, r;
04     int op;
05     printf("Digite 1 para somar;\n");
```

```
06 printf("Digite 2 para subtrair;\n");
07 printf("Digite 3 para multiplicar;\n");
08 printf("Digite 4 para dividir;\n");
09 scanf("%d", &op);
10 printf("Insira o primeiro valor:\n");
11 scanf("%f", &n1);
12 printf("Insira o segundo valor:\n");
13 scanf("%f", &n2);
14 if(op == 1){
15     r = n1 + n2;
16     printf("Soma: %f\n", r);
17 }
18 if(op == 2){
19     r = n1 - n2;
20     printf("Subtração: %f\n", r);
21 }
22 if(op == 3){
23     r = n1 * n2;
24     printf("Multiplicação: %f\n", r);
25 }
26 if(op == 4){
27     if(n2 != 0){
28         r = n1 / n2;
29         printf("Divisão: %f\n", r);
30     } else {
31         printf("Divisão por zero.\n");
32     }
33 }
34 }
```

- 6) Desenvolva um algoritmo que peça para que o usuário informe a base e a altura de um retângulo, e um terceiro número inteiro "op". Caso o usuário escolha "op" igual a 0, calcule e mostre o perímetro do retângulo. Caso o usuário insira um valor 1 para "op", calcule e mostre a área do retângulo. Se o usuário inserir um valor diferente de 0 e 1 para "op", mostrar a mensagem "Opção inválida.".

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

#### Exemplo de execução – Exercício 6 - Caso de teste 1

```
Insira a base:
2
Insira a altura:
2
Opções de processamento:
Insira 0 para calcular o perímetro;
Insira 1 para calcular a área.
```



```
0
Perímetro: 8
```

#### Exemplo de execução – Exercício 6 - Caso de teste 2

```
Insira a base:
2
Insira a altura:
2
Opções de processamento:
Insira 1 para calcular o perímetro;
Insira 2 para calcular a área.
1
Área: 4
```

#### Exemplo de execução – Exercício 6 - Caso de teste 3

```
Insira a base:
2
Insira a altura:
2
Opções de processamento:
Insira 0 para calcular o perímetro;
Insira 1 para calcular a área.
2
Opção inválida.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segure aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 6 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float base, altura, calc;
04     int op;
05     printf("Insira a base:\n");
06     scanf("%f", &base);
07     printf("Insira a altura:\n");
08     scanf("%f", &altura);
09     printf("Opções de processamento:\n");
10     printf("Insira 0 para calcular o perímetro;\n");
11     printf("Insira 1 para calcular a área.\n");
12     scanf("%d", &op);
13     if ((op == 0) || (op == 1)){
14         if(op == 0){
15             calc = 2*altura + 2*base;
16             printf("Perímetro: %f.\n", calc);
17         }
18         if(op == 1){
19             calc = altura*base;
20             printf("Área: %f.\n", calc);
21         }
22     } else {
```

```
23     printf("Opção inválida.\n");  
24 }  
25 }
```

- 7) (Adaptado de ASCENCIO e CAMPOS, 2008) A nota final de um estudante é calculada a partir de três notas atribuídas respectivamente a um trabalho de laboratório, a uma avaliação semestral e a um exame final. A média das três notas mencionadas anteriormente obedece aos pesos a seguir:

Nota	Peso
Trabalho de laboratório	2
Avaliação semestral	3
Exame final	4

Faça um programa que receba as três notas, calcule e mostre a média ponderada e o conceito que segue a tabela abaixo:

Média ponderada	Conceito
8,0 ~ 10,0	A
7,0 ~ 8,0	B
6,0 ~ 7,0	C
5,0 ~ 6,0	D
0,0 ~ 5,0	E

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 7 - Caso de teste 1

Insira a nota do trabalho:

8

Insira a nota da avaliação:

8

Insira a nota do exame:

8

Média ponderada: 8.00

Conceito A

Exemplo de execução – Exercício 7 - Caso de teste 2

Insira a nota do trabalho:

7

Insira a nota da avaliação:

7

Insira a nota do exame:

7

Média ponderada: 7.00

### Conceito B

#### Exemplo de execução – Exercício 7 - Caso de teste 3

Insira a nota do trabalho:

6

Insira a nota da avaliação:

6

Insira a nota do exame:

6

Média ponderada: 6.00

Conceito C

#### Exemplo de execução – Exercício 7 - Caso de teste 4

Insira a nota do trabalho:

5

Insira a nota da avaliação:

5

Insira a nota do exame:

5

Média ponderada: 5.00

Conceito D

#### Exemplo de execução – Exercício 7 - Caso de teste 5

Insira a nota do trabalho:

0

Insira a nota da avaliação:

0

Insira a nota do exame:

0

Média ponderada: 0.00

Conceito E

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 7 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float trab, aval, exam, med;
04     printf("Insira a nota do trabalho:\n");
05     scanf("%f", &trab);
06     printf("Insira a nota da avaliação:\n");
07     scanf("%f", &aval);
08     printf("Insira a nota do exame:\n");
09     scanf("%f", &exam);
10     med = (trab*2 + aval*3 + exam*5)/10;
11     printf("Média ponderada: %.2f\n", med);
12     if ((med >= 8.0) && (med <= 10.0)){
13         printf("Conceito A.\n");
14     }
15     if ((med >= 7.0) && (med < 8.0)){
```

```
16     printf("Conceito B.\n");
17 }
18 if ((med >= 6.0) && (med < 7.0)){
19     printf("Conceito C.\n");
20 }
21 if ((med >= 5.0) && (med < 6.0)){
22     printf("Conceito D.\n");
23 }
24 if ((med >= 0.0) && (med < 5.0)){
25     printf("Conceito E.\n");
26 }
27 }
```

- 8) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba três notas de um aluno, calcule e mostre a média aritmética e a mensagem que segue a tabela abaixo. Para alunos de exame, calcule e mostre a nota que deverá ser tirada no exame para aprovação, considerando que a média no exame é 6,0.

Média aritmética	Mensagem
0,0 ~ 3,0	Reprovado
3,0 ~ 7,0	Exame
7,0 ~ 10,0	Aprovado

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

#### Exemplo de execução – Exercício 8 - Caso de teste 1

```
Insira a primeira nota:
7
Insira a segunda nota:
7
Insira a terceira nota:
7
Média: 7
Aprovado! Parabéns.
```

#### Exemplo de execução – Exercício 8 - Caso de teste 2

```
Insira a primeira nota:
3
Insira a segunda nota:
3
Insira a terceira nota:
3
Média: 3
Exame.
É preciso tirar: 9
```

#### Exemplo de execução – Exercício 8 - Caso de teste 3

```
Insira a primeira nota:
0
Insira a segunda nota:
0
Insira a terceira nota:
0
Média: 0
Reprovado.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 8 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float n1, n2, n3, med, exam;
04     printf("Insira a primeira nota:\n");
05     scanf("%f", &n1);
06     printf("Insira a segunda nota:\n");
07     scanf("%f", &n2);
08     printf("Insira a terceira nota:\n");
09     scanf("%f", &n3);
10     med = (n1+n2+n3)/3;
11     printf("Média: %.1f.\n", med);
12     if((med >= 0.0) && (med < 3.0)){
13         printf("Reprovado.\n");
14     }
15     if((med >= 3.0) && (med < 7.0)){
16         printf("Exame.\n");
17         exam = 12 - med;
18         printf("É preciso tirar %.1f no exame.\n", exam);
19     }
20     if((med >= 7.0) && (med <= 10.0)){
21         printf("Aprovado! Parabéns.\n");
22     }
23 }
```

- 9) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba três números distintos e mostre-os em ordem crescente.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 9 - Caso de teste 1

Insira o primeiro número:

1

Insira o segundo número:

2

Insira o terceiro número:

3

1 - 2 - 3

Exemplo de execução – Exercício 9 - Caso de teste 2

Insira o primeiro número:

1

Insira o segundo número:

3

Insira o terceiro número:

2

1 - 2 - 3

Exemplo de execução – Exercício 9 - Caso de teste 3

Insira o primeiro número:

2

Insira o segundo número:

1

Insira o terceiro número:

3

1 - 2 - 3

Exemplo de execução – Exercício 9 - Caso de teste 4

Insira o primeiro número:

2

Insira o segundo número:

3

Insira o terceiro número:

1

1 - 2 - 3

Exemplo de execução – Exercício 9 - Caso de teste 5

Insira o primeiro número:

3

Insira o segundo número:

1

Insira o terceiro número:

2

1 - 2 - 3

Exemplo de execução – Exercício 9 - Caso de teste 6

Insira o primeiro número:

3

Insira o segundo número:

2

Insira o terceiro número:

1

1 - 2 - 3

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 9 – Solução

```
01 #include <stdio.h>
02 int main() {
03     float n1, n2, n3;
04     printf("Insira o primeiro número:\n");
05     scanf("%f", &n1);
06     printf("Insira o segundo número:\n");
07     scanf("%f", &n2);
08     printf("Insira o terceiro número:\n");
09     scanf("%f", &n3);
10     if((n1 < n2) && (n1 < n3)) {
11         if(n2 < n3) {
12             printf("%f - %f - %f.\n", n1, n2, n3);
13         } else {
14             printf("%f - %f - %f.\n", n1, n3, n2);
15         }
16     }
17     if((n2 < n1) && (n2 < n3)) {
18         if(n1 < n3) {
19             printf("%f - %f - %f.\n", n2, n1, n3);
20         } else {
21             printf("%f - %f - %f.\n", n2, n3, n1);
22         }
23     }
24     if((n3 < n1) && (n3 < n2)) {
25         if(n1 < n2) {
26             printf("%f - %f - %f.\n", n3, n1, n2);
27         } else {
28             printf("%f - %f - %f.\n", n3, n2, n1);
29         }
30     }
31 }
```

- 10)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba três números obrigatoriamente em ordem crescente e um quarto número que não siga esta regra. Mostre, em seguida, os quatro números em ordem não-crescente.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).



#### Exemplo de execução - Exercício 10 - Caso de teste 1

```
Insira o primeiro número:  
10  
Insira o segundo número:  
20  
Insira o terceiro número:  
30  
Insira o quarto número:  
5  
30 - 20 - 10 - 5
```

#### Exemplo de execução - Exercício 10 - Caso de teste 2

```
Insira o primeiro número:  
10  
Insira o segundo número:  
20  
Insira o terceiro número:  
30  
Insira o quarto número:  
15  
30 - 20 - 15 - 10
```

#### Exemplo de execução - Exercício 10 - Caso de teste 3

```
Insira o primeiro número:  
10  
Insira o segundo número:  
20  
Insira o terceiro número:  
30  
Insira o quarto número:  
25  
30 - 25 - 20 - 10
```

#### Exemplo de execução - Exercício 10 - Caso de teste 4

```
Insira o primeiro número:  
10  
Insira o segundo número:  
20  
Insira o terceiro número:  
30  
Insira o quarto número:  
35  
35 - 30 - 20 - 10
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO - Exercício 10 - Solução

```
01 #include <stdio.h>
02 int main() {
03     float n1, n2, n3, n4;
04     printf("Insira o primeiro número:\n");
05     scanf("%f", &n1);
06     printf("Insira o segundo número:\n");
07     scanf("%f", &n2);
08     printf("Insira o terceiro número:\n");
09     scanf("%f", &n3);
10     printf("Insira o quarto número:\n");
11     scanf("%f", &n4);
12
13     if(n4 > n3){
14         printf("%f - %f - %f - %f.\n", n4, n3, n2, n1);
15     }
16     if((n4 > n2) && (n4 <= n3)){
17         printf("%f - %f - %f - %f.\n", n3, n4, n2, n1);
18     }
19     if((n4 > n1) && (n4 <= n2)){
20         printf("%f - %f - %f - %f.\n", n3, n2, n4, n1);
21     }
22     if((n4 <= n1)){
23         printf("%f - %f - %f - %f.\n", n3, n2, n1, n4);
24     }
25 }
```

- 11) Desenvolva um algoritmo que peça ao usuário que informe os coeficientes  $a$ ,  $b$  e  $c$  de uma equação de segundo grau:  $ax^2 + bx + c$ . Com base na Fórmula de Bhaskara, calcule e mostre as raízes da respectiva equação de segundo grau.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 11 - Caso de teste

```
Insira o coeficiente a
1
Insira o coeficiente b
-5
Insira o coeficiente c
6
Raiz x': 3
Raiz x'': 2
```

Exemplo de execução – Exercício 11 - Caso de teste 2

```
Insira o coeficiente a
2
Insira o coeficiente b
```

```
1
Insira o coeficiente c
2
Raízes fora do domínio dos números reais.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 11 – Solução

```
01 int main() {
02     int a, b, c, delta;
03     float x1, x2;
04
05     printf("Insira o coeficiente a:\n");
06     scanf("%d", &a);
07     printf("Insira o coeficiente b:\n");
08     scanf("%d", &b);
09     printf("Insira o coeficiente c:\n");
10     scanf("%d", &c);
11
12     delta = b*b - 4*a*c;
13
14     if(delta < 0) {
15         printf("Raízes fora do domínio dos números reais.\n");
16     } else {
17         x1 = (-b + sqrt(delta))/(2*a);
18         x2 = (-b - sqrt(delta))/(2*a);
19         printf("Raiz x': %f.\n", x1);
20         printf("Raiz x'': %f.\n", x2);
21     }
22 }
```

- 12) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba quatro valores, I, A, B e C. I é um valor inteiro e positivo e A, B e C são valores reais e distintos. Escreva os números A, B e C obedecendo à tabela a seguir. Supor que o valor digitado para I seja sempre um valor válido, ou seja, 1, 2 ou 3.

Valor de I	Forma de escrever
1	A, B e C em ordem crescente
2	A, B e C em ordem decrescente
3	O maior fica entre os outros dois números

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em

branco correspondem a dados informados pelo usuário (operações de entrada – scanf, gets ou fgets).

Exemplo de execução – Exercício 12 - Caso de teste 1

Insira o primeiro número (A):

1

Insira o segundo número (B):

2

Insira o terceiro número (C):

3

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 2

Insira o primeiro número (A):

1

Insira o segundo número (B):

3

Insira o terceiro número (C):

2

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 3

Insira o primeiro número (A):

2

Insira o segundo número (B):

1

Insira o terceiro número (C):

3

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 4

Insira o primeiro número (A):

2

Insira o segundo número (B):

3

Insira o terceiro número (C):

1

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 5

Insira o primeiro número (A):

3

Insira o segundo número (B):

1

Insira o terceiro número (C):

2

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 6

Insira o primeiro número (A):

3

Insira o segundo número (B):

2

Insira o terceiro número (C):

1

Insira a opção (I):

1

1 - 2 - 3

Exemplo de execução – Exercício 12 - Caso de teste 7

Insira o primeiro número (A):

1

Insira o segundo número (B):

2

Insira o terceiro número (C):

3

Insira a opção (I):

2

3 - 2 - 1

Exemplo de execução – Exercício 12 - Caso de teste 8

Insira o primeiro número (A):

1

Insira o segundo número (B):

3

Insira o terceiro número (C):

2

Insira a opção (I):

2

3 - 2 - 1

Exemplo de execução – Exercício 12 - Caso de teste 9

Insira o primeiro número (A):

2

Insira o segundo número (B):

1

Insira o terceiro número (C):

3

Insira a opção (I):

2

3 - 2 - 1

Exemplo de execução – Exercício 12 - Caso de teste 10

Insira o primeiro número (A):

```
2
Insira o segundo número (B):
3
Insira o terceiro número (C):
1
Insira a opção (I):
2
3 - 2 - 1
```

#### Exemplo de execução – Exercício 12 - Caso de teste 11

```
Insira o primeiro número (A):
3
Insira o segundo número (B):
1
Insira o terceiro número (C):
2
Insira a opção (I):
2
3 - 2 - 1
```

#### Exemplo de execução – Exercício 12 - Caso de teste 12

```
Insira o primeiro número (A):
3
Insira o segundo número (B):
2
Insira o terceiro número (C):
1
Insira a opção (I):
2
3 - 2 - 1
```

#### Exemplo de execução – Exercício 12 - Caso de teste 13

```
Insira o primeiro número (A):
1
Insira o segundo número (B):
2
Insira o terceiro número (C):
3
Insira a opção (I):
3
1 - 3 - 2
```

#### Exemplo de execução – Exercício 12 - Caso de teste 14

```
Insira o primeiro número (A):
1
Insira o segundo número (B):
3
Insira o terceiro número (C):
2
Insira a opção (I):
3
1 - 3 - 2
```

Exemplo de execução – Exercício 12 - Caso de teste 15

Insira o primeiro número (A):

2

Insira o segundo número (B):

1

Insira o terceiro número (C):

3

Insira a opção (I):

3

2 - 3 - 1

Exemplo de execução – Exercício 12 - Caso de teste 16

Insira o primeiro número (A):

2

Insira o segundo número (B):

3

Insira o terceiro número (C):

1

Insira a opção (I):

3

2 - 3 - 1

Exemplo de execução – Exercício 12 - Caso de teste 17

Insira o primeiro número (A):

3

Insira o segundo número (B):

1

Insira o terceiro número (C):

2

Insira a opção (I):

3

1 - 3 - 2

Exemplo de execução – Exercício 12 - Caso de teste 18

Insira o primeiro número (A):

3

Insira o segundo número (B):

2

Insira o terceiro número (C):

1

Insira a opção (I):

3

2 - 3 - 1

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 12 – Solução



```
01 #include <stdio.h>
02 int main(){
03     float A, B, C;
04     int I;
05     printf("Insira o primeiro número (A):\n");
06     scanf("%f", &A);
07     printf("Insira o segundo número (B):\n");
08     scanf("%f", &B);
09     printf("Insira o terceiro número (C):\n");
10     scanf("%f", &C);
11     printf("Insira a opção (I):\n");
12     scanf("%d", &I);
13     if (I == 1){
14         if((A < B) && (A < C)){
15             if(B < C){
16                 printf("%f - %f - %f.\n", A, B, C);
17             } else {
18                 printf("%f - %f - %f.\n", A, C, B);
19             }
20         }
21         if((B < A) && (B < C)){
22             if(A < C){
23                 printf("%f - %f - %f.\n", B, A, C);
24             } else {
25                 printf("%f - %f - %f.\n", B, C, A);
26             }
27         }
28         if((C < A) && (C < B)){
29             if(A < B){
30                 printf("%f - %f - %f.\n", C, A, B);
31             } else {
32                 printf("%f - %f - %f.\n", C, B, A);
33             }
34         }
35     }
36     if (I == 2){
37         if((A > B) && (A > C)){
38             if(B > C){
39                 printf("%f - %f - %f.\n", A, B, C);
40             } else {
41                 printf("%f - %f - %f.\n", A, C, B);
42             }
43         }
44         if((B > A) && (B > C)){
45             if(A > C){
46                 printf("%f - %f - %f.\n", B, A, C);
47             } else {
48                 printf("%f - %f - %f.\n", B, C, A);
49             }
50         }
51     }
```

```
56         }
57     }
58     if((C > A) && (C > B)){
59         if(A > B){
60             printf("%f - %f - %f.\n", C, A, B);
61         } else {
62             printf("%f - %f - %f.\n", C, B, A);
63         }
64     }
65 }
66 }
67 if(I == 3){
68     if ((A > B) && (A > C)){
69         printf("%f - %f - %f.\n", B, A, C);
70     }
71     if ((B > A) && (B > C)){
72         printf("%f - %f - %f.\n", A, B, C);
73     }
74     if ((C > A) && (C > B)){
75         printf("%f - %f - %f.\n", A, C, B);
76     }
77 }
78 }
79 }
80 }
```

**13)** (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que mostre o menu de opções a seguir, receba a opção do usuário e dos dados necessários para executar cada operação. Suponha que o usuário não irá inserir valores inválidos.

Menu de opções:

1. Somar dois números
2. Raiz quadrada de um número

Digite a opção desejada

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 13 - Caso de teste 1

```
--- Menu de Opções ---
1 - Somar dois números;
2 - Raiz quadrada de um número;
1
Insira o primeiro número
2
Insira o segundo número
2
```

Resultado da soma: 4

Exemplo de execução – Exercício 13 - Caso de teste 2

```
--- Menu de Opções ---  
1 - Somar dois números;  
2 - Raiz quadrada de um número;  
2  
Insira o número  
9  
Resultado da raiz quadrada: 3
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 13 – Solução

```
01 #include <stdio.h>  
02 #include <math.h>  
03 int main(){  
04     float n1, n2, calc;  
05     int op;  
06     printf("---- Menu de Opções ----\n");  
07     printf("1 - Somar dois números;\n");  
08     printf("2 - Raiz quadrada de um número.\n");  
09     scanf("%d", &op);  
10     if(op == 1){  
11         printf("Insira o primeiro número:\n");  
12         scanf("%f", &n1);  
13         printf("Insira o segundo número:\n");  
14         scanf("%f", &n2);  
15         calc = n1 + n2;  
16         printf("Resultado da soma: %f.\n", calc);  
17     }  
18     if(op == 2){  
19         printf("Insira o número:\n");  
20         scanf("%f", &n1);  
21         calc = sqrt(n1);  
22         printf("Resultado da raiz quadrada: %f.\n", calc);  
23     }  
24 }
```

- 14) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba a hora de início de um jogo e a hora final do jogo (cada hora é composta por duas variáveis inteiras: hora e minuto). Calcule e mostre a duração do jogo (horas e minutos) sabendo-se que o tempo máximo de duração do jogo é de 24 horas e que o jogo pode iniciar em um dia e terminar no dia seguinte. Observação: utilizar o formato de hora que vai das 00hr00min até as 23hr29min.

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

#### Exemplo de execução – Exercício 14 - Caso de teste 1

Insira os dados do início:

Horas:

10

Minutos:

15

Insira os dados do fim:

Horas:

11

Minutos:

30

Duração: 1hr e 15min.

#### Exemplo de execução – Exercício 14 - Caso de teste 2

Insira os dados do início:

Horas:

10

Minutos:

15

Insira os dados do fim:

Horas:

12

Minutos:

10

Duração: 1hr e 55min.

#### Exemplo de execução – Exercício 14 - Caso de teste 3

Insira os dados do início:

Horas:

22

Minutos:

30

Insira os dados do fim:

Horas:

01

Minutos:

50

Duração: 3hr e 20min.

#### Exemplo de execução – Exercício 14 - Caso de teste 4

Insira os dados do início:

Horas:

22

Minutos:

30

Insira os dados do fim:

```
Horas:
01
Minutos:
15
Duração: 2hr e 45min.
```

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

#### ESTRUTURAS DE DECISÃO – Exercício 14 – Solução

```
01 #include <stdio.h>
02 int main(){
03     int hr_ini, min_ini, hr_fim, min_fim, hora, minuto;
04     printf("Insira os dados do início:\n");
05     printf("Horas: ");
06     scanf("%d", &hr_ini);
07     printf("Minutos: ");
08     scanf("%d", &min_ini);
09     printf("Insira os dados do fim:\n");
10     printf("Horas: ");
11     scanf("%d", &hr_fim);
12     printf("Minutos: ");
13     scanf("%d", &min_fim);
14     if(min_ini > min_fim){
15         min_fim = min_fim + 60;
16         hr_fim = hr_fim - 1;
17     }
18     if(hr_ini > hr_fim){
19         hr_fim = hr_fim + 24;
20     }
21     minuto = min_fim - min_ini;
22     hora = hr_fim - hr_ini;
23     printf("Duração %dhr e %dmin.\n", hora, minuto);
24 }
```

- 15) (Adaptado de ASCENCIO e CAMPOS, 2008) Faça um programa que receba o salário de um funcionário, calcule e mostre o novo salário desse funcionário, acrescido de bonificação e de auxílio-escola.

Salário	Bonificação
Até R\$ 500,00	5% do salário
Entre R\$ 500,01 e R\$ 1.200,00	12% do salário
Acima de R\$ 1.200,00	Sem bonificação

Salário	Auxílio-escola
Até R\$ 600,00	R\$ 150,00

Mais que R\$ 600,00	R\$ 600,00
---------------------	------------

Quer dicas de como seu algoritmo deveria funcionar? Observe os quadros abaixo, nos quais você encontra simulações da execução do algoritmo. Textos que estão em azul são mensagens geradas pela máquina (operações de saída – `printf` ou `puts`). Já os textos que estão em branco correspondem a dados informados pelo usuário (operações de entrada – `scanf`, `gets` ou `fgets`).

Exemplo de execução – Exercício 15 – Caso de teste 1

Informe o salário:

500

Salário total: 675

Exemplo de execução – Exercício 15 – Caso de teste 2

Informe o salário:

1200

Salário total: 1444

Exemplo de execução – Exercício 15 – Caso de teste 3

Informe o salário:

1500

Salário total: 1600

Antes de verificar o gabarito desta questão, limpe sua consciência: Dedique ao MENOS 10 MINUTOS do seu tempo tentando resolver esse exercício. Caso considere que já tenha tentado o suficiente, segue aí uma solução que funciona:

ESTRUTURAS DE DECISÃO – Exercício 15 – Solução

```
01 #include <stdio.h>
02 int main(){
03     float sal, sal_novo, bonif, aux;
04     printf("Informe o salário:\n");
05     scanf("%f", &sal);
06
07     if(sal <= 500){
08         bonif = sal * 0.05;
09     }
10     if(sal > 500){
11         bonif = sal * 0.12;
12     }
13     if(sal > 1200){
14         bonif = 0;
15     }
16     if(sal <= 600){
17         aux = 150;
18     }
19     if(sal > 600){
20         aux = 100;
21     }
22     sal_novo = sal + bonif + aux;
23     printf("Salário total: %.2f.\n", sal_novo);
```

```
24 }
```