

Lista de exercícios para For, If/Else

Minicurso de C

Março de 2025

Sumário

1	Exercícios de <i>for</i>	3
1.1	Exercício 1	3
1.2	Exercício 2	3
1.3	Exercício 3	3
1.4	Exercício 4	3
1.5	Exercício 5	3
2	Exercícios de <i>If/Else</i>	4
2.1	Exercício 1	4
2.2	Exercício 2	4
2.3	Exercício 3	4
2.4	Exercício 4	4
2.5	Exercício 5	5
3	Respostas para os exercícios da seção 1	6
3.1	1.1	6
3.2	1.2	6
3.3	1.3	6
3.4	1.4	7
3.5	1.5	8
4	Respostas para os exercícios da seção 2	8
4.1	2.1	8
4.2	2.2	9
4.3	2.3	10
4.4	2.4	11
4.5	2.5	12

1 Exercícios de *for*

1.1 Exercício 1

Escreva um programa em C, utilizando o *for*, que print os 10 primeiros números naturais.

1.2 Exercício 2

Escreva um programa em C que, dado um número inteiro fornecido pelo usuário, imprima a tabuada completa desse número (de 1 a 10).

Exemplo de saída para o número 2:

$$\begin{array}{rcl} 2 \times 1 & = & 2 \\ 2 \times 2 & = & 4 \\ 2 \times 3 & = & 6 \\ \vdots & & \vdots \\ 2 \times 10 & = & 20 \end{array}$$

1.3 Exercício 3

Escreva um programa em C que, dado uma quantidade inteira *num* definida pelo usuário, retorne todos os números ímpares de 1 à *num*. Em seguida, o programa deve imprimir a soma de todos os números primos printados anteriormente.

Saída esperada para *num* = 10

1, 3, 5, 7, 9, 11, 13, 15, 17, 19
Soma dos *num* primeiros números ímpares: 100

1.4 Exercício 4

DESAFIO

Escreva um programa em C que, dado um *range* definido pelo usuário, retorne todos os números primos contidos nesse intervalo.

Exemplo de entrada/saída:

Digite o início do intervalo: 10
Digite o fim do intervalo: 30
Números primos no intervalo [10, 30]: 11, 13, 17, 19, 23, 29

1.5 Exercício 5

DESAFIO

Escreva um programa em C que imprima uma pirâmide alfabética, onde o número de níveis é definido pelo usuário. A pirâmide deve ser centralizada e seguir o padrão abaixo.

Exemplo de saída para 6 níveis:

```
      A
    A B A
  A B C B A
A B C D C B A
A B C D E D C B A
A B C D E F E D C B A
```

2 Exercícios de *If/Else*

2.1 Exercício 1

Escreva um programa em C que ordene em ordem crescente 3 números fornecidos pelo usuário.

2.2 Exercício 2

Escreva um programa em C que retorne as duas raízes **reais** de uma equação quadrática.

2.3 Exercício 3

Escreva um programa em C que classifique um triângulo de acordo com os lados fornecidos pelo usuário.

Exemplo de saída

```
Triângulo isósceles!
```

2.4 Exercício 4

DESAFIO

Escreva um programa em C que calcule o desconto de uma compra, com base nos seguintes dados de entrada:

- **Valor da compra** (tipo float)
- **Tipo de cliente:**
 - 1 = Normal
 - 2 = VIP
 - 3 = Funcionário
- **Forma de pagamento:**
 - 1 = Débito
 - 2 = Crédito
 - 3 = Pix

Regras de Desconto

Tipo de Cliente	Forma de Pagamento	Desconto
Normal	Débito	5%
	Crédito	Sem desconto
	Pix	7%
VIP	Débito	10% + 5% extra se valor > R\$500
	Crédito	8%
	Pix	12%
Funcionário	Todos	15% + 10% extra se valor > R\$300

Tabela 1: Tabela de descontos conforme cliente e forma de pagamento

Saída Esperada

O programa deve exibir:

- Valor original da compra
- Valor do desconto aplicado
- Valor final após o desconto

Exemplo de Cálculo

Para um **cliente VIP** com compra de **R\$600** no **débito**:

$$\text{Desconto padrão} = 600 \times 0,10 = \text{R\$60}$$

$$\text{Desconto extra} = 600 \times 0,05 = \text{R\$30}$$

$$\text{Desconto total} = \text{R\$90}$$

$$\text{Valor final} = 600 - 90 = \text{R\$510}$$

2.5 Exercício 5

DESAFIO

Escreva um programa em C para inserir a tarifa unitária de eletricidade e calcular a conta total de eletricidade de acordo com as condições fornecidas.

Faixa de Unidades	Preço por Unidade (R\$)	Observação
1 – 50	0,50	Preço base
51 – 150	0,75	Acréscimo nas próximas 100 unidades
151 – 250	1,20	Acréscimo nas próximas 100 unidades
Acima de 250	1,50	Preço máximo por unidade
Total	+20%	Sobretaxa adicional sobre o valor total

Tabela 2: Estrutura de Preços por Unidade

3 Respostas para os exercícios da seção 1

3.1 1.1

```
1 #include <stdio.h>
2
3 int main() {
4     int i;
5
6     printf("Os 10 primeiros numeros naturais sao:\n");
7
8     for (i = 1; i <= 10; i++) {
9         printf("%d ", i);
10    }
11
12    printf("\n");
13    return 0;
14 }
```

Listing 1: Imprime os 10 primeiros números naturais.

3.2 1.2

```
1 #include <stdio.h>
2
3 void main() {
4     int j, n;
5
6     printf("Digite o numero (para calcular a tabuada): ");
7     scanf("%d", &n);
8     printf("\n");
9
10    for (j = 1; j <= 10; j++) {
11        printf("%d X %d = %d \n", n, j, n * j);
12    }
13 }
```

Listing 2: Programa que calcula a tabuada

3.3 1.3

```
1 #include <stdio.h>
2
3 #include <stdio.h>
4
5 int main() {
6     int num, sum = 0, i;
7
8     printf("Digite a quantidade de numeros desejados: ");
9     scanf("%d", &num);
```

```

10
11     printf("\n");
12
13     printf("Os numeros impares de 1 ate %d sao: ", num);
14
15     for(i = 1; i <= num; i++){
16         printf("%d, ", 2*i - 1);
17         sum += 2*i - 1;
18     }
19     printf("\n\n");
20     printf("Soma dos primeiros %d numeros impares: %d", num, sum);
21     return 0;
22 }

```

Listing 3: Programa que identifica e soma todos os n-esimos numeros primos

3.4 1.4

```

1  #include <stdio.h>
2
3  int main() {
4      int num, i, ctr, stno, enno;
5      // Ctr = 0 -> numero primo
6      // Ctr = 1 -> numero nao primo
7      printf("Digite o inicio do intervalo: ");
8      scanf("%d", &stno);
9
10     printf("Digite o fim do intervalo: ");
11     scanf("%d", &enno);
12
13     printf("Numeros primos entre %d e %d:\n", stno, enno);
14
15     for(num = stno; num <= enno; num++) {
16         ctr = 0;
17
18         (num < 2) ? (ctr = 1) : 0;
19
20
21         for(i = 2; i*i <= num && !ctr; i++) {
22             (num % i == 0) ? (ctr = 1) : 0;
23         }
24
25         (!ctr) && printf("%d ", num);
26     }
27
28     printf("\n");
29     return 0;
30 }
31
32 }

```

Listing 4: Programa para encontrar numeros primos sem if

3.5 1.5

```
1 #include <stdio.h>
2
3 int main(){
4
5     int num_layers;
6     char letter = 'A';
7     int i, j, k;
8
9     printf("Digite o numero de camadas desejado: \n");
10    scanf("%d", &num_layers);
11
12    for(i = 0; i < num_layers; i++){
13        for(j = 0; j <= num_layers - i; j++){
14            printf(" ");
15        }
16        for(k = 0; k <= i; k++){
17            printf("%c ", 'A' + k);
18        }
19        for(k = i - 1; k >= 0; k--){
20            printf("%c ", 'A' + k);
21        }
22        printf("\n");
23    }
24
25 }
```

Listing 5: Programa para desenhar uma estrutura de piramide

4 Respostas para os exerc cios da se  o 2

4.1 2.1

```
1 #include <stdio.h>
2
3 int main(){
4
5     int num1, num2, num3;
6
7     int temp;
8
9     printf("Insira 3 numeros inteiros e positivos: ");
10
11    if(scanf("%d %d %d", &num1, &num2, &num3) != 3 || num1 < 0 ||
        num2 < 0 || num3 < 0){
```



```

12     printf("Falha critica!");
13     return 1;
14 }
15     if(num1 > num2){
16         temp = num1;
17         num1 = num2;
18         num2 = temp;
19     }
20     if(num1 > num3){
21         temp = num1;
22         num1 = num3;
23         num3 = temp;
24     }
25     if(num2 > num3){
26         temp = num2;
27         num2 = num3;
28         num3 = temp;
29     }
30     printf("Em ordem crescente: \n");
31     printf("%d, %d, %d", num1, num2, num3);
32
33     return 0;
34 }

```

Listing 6: Programa que ordena 3 números naturais

4.2 2.2

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5      float a, b, c;
6      float x1, x2;
7      char termo = 'a';
8
9      for (int i = 0; i < 3; i++) {
10         printf("Digite um valor para %c: ", termo);
11         switch (termo) {
12             case 'a':
13                 scanf("%f", &a);
14                 break;
15             case 'b':
16                 scanf("%f", &b);
17                 break;
18             case 'c':
19                 scanf("%f", &c);
20                 break;
21         }
22         termo++;
23     }

```

```

24
25     if(a == 0){
26         printf("a nao pode ser igual a 0!");
27         return 1;
28     }
29
30     float delta = b*b - 4*a*c;
31     if(delta < 0){
32         printf("Sua equacao nao possui valores reais!");
33         return 1;
34     }
35
36     x1 = (-b + sqrt(delta))/(2*a);
37     x2 = (-b - sqrt(delta))/(2*a);
38
39     printf("\n\tS{x1 = %f e x2 = %f}", x1, x2);
40 }

```

Listing 7: Programa que calcula as raízes reais de uma equação do segundo grau.

4.3 2.3

```

1 #include <stdio.h>
2
3 int main() {
4     float a, b, c;
5
6     printf("Digite o comprimento do primeiro lado: ");
7     scanf("%f", &a);
8     printf("Digite o comprimento do segundo lado: ");
9     scanf("%f", &b);
10    printf("Digite o comprimento do terceiro lado: ");
11    scanf("%f", &c);
12
13    if (a + b > c && a + c > b && b + c > a) {
14        if (a == b && b == c) {
15            printf("O tri ngulo      Equil tero.\n");
16        } else if (a == b || b == c || a == c) {
17            printf("O tri ngulo      Is sceles.\n");
18        } else {
19            printf("O tri ngulo      Escaleno.\n");
20        }
21    } else {
22        printf("Os valores fornecidos n o formam um tri ngulo
23              v lido.\n");
24    }
25
26    return 0;
27 }

```

Listing 8: Programa que verifica a validade e o tipo de triangulo com base nos lados

4.4 2.4

```
1 #include <stdio.h>
2
3 int main() {
4     float valorCompra, desconto = 0.0;
5     int tipoCliente, formaPagamento;
6
7     printf("Digite o valor da compra: R$ ");
8     scanf("%f", &valorCompra);
9
10    printf("Digite o tipo de cliente (1-Normal, 2-VIP,
11           3-Funcionario): ");
12    scanf("%d", &tipoCliente);
13
14    printf("Digite a forma de pagamento (1-D bito, 2-Cr dito,
15           3-Pix): ");
16    scanf("%d", &formaPagamento);
17
18    switch (tipoCliente) {
19        case 1:
20            if (formaPagamento == 1)
21                desconto = valorCompra * 0.05;
22            else if (formaPagamento == 3)
23                desconto = valorCompra * 0.07;
24            break;
25
26        case 2:
27            if (formaPagamento == 1) {
28                desconto = valorCompra * 0.10;
29                if (valorCompra > 500)
30                    desconto += valorCompra * 0.05;
31            } else if (formaPagamento == 2)
32                desconto = valorCompra * 0.08;
33            else if (formaPagamento == 3)
34                desconto = valorCompra * 0.12;
35            break;
36
37        case 3:
38            desconto = valorCompra * 0.15;
39            if (valorCompra > 300)
40                desconto += valorCompra * 0.10;
41            break;
42
43        default:
44            printf("Tipo de cliente inv lido!\n");
45            return 1;
46    }
47
48    float valorFinal = valorCompra - desconto;
```

```

48     printf("Valor original: R$ %.2f\n", valorCompra);
49     printf("Desconto aplicado: R$ %.2f\n", desconto);
50     printf("Valor final: R$ %.2f\n", valorFinal);
51
52     return 0;
53 }

```

Listing 9: Programa para cálculo de descontos

4.5 2.5

```

1  #include <stdio.h>
2
3  int main(){
4
5      float preco;
6      int unidade;
7
8      printf("Digite a quantidade de unidades consumidas: ");
9      scanf("%d", &unidade);
10
11     if(unidade <= 50){
12
13         preco = unidade*0.5;
14
15     }else if(unidade <= 150){
16
17         preco = 25 + (unidade - 50)*0.75;
18
19     }else if(unidade <= 250){
20
21         preco = 100 + (unidade - 150)*1.2;
22
23     }else{
24
25         preco = 220 + (unidade - 250)*1.5;
26
27     }
28
29     preco = preco*1.2;
30
31     printf("Conta de energia: R$%.2f", preco);
32     return 0;
33 }

```

Listing 10: Programa que calcula a conta energética com base na quantidade de unidades energéticas consumidas

João Pedro Freire de Albuquerque
João Pedro Freire de Albuquerque
Universidade Federal do Rio Grande do Norte
Data: 6 de maio de 2025