

Lista de exercícios Estrutura de Dados 1

Nome: Richard Henry Hartmann – RGM: 29701961

Curso: Ciência da Computação – Turma: 3B

`richardhartmann2@gmail.com`

Universidade da Cidade de São Paulo (UNICID) - Rua Cesário Galeno, 448/475 São Paulo – SP – Brasil – CEP: 03071-000

***Abstract.** This project aims to answer a list of exercises in the question Structures of Data I, through the knowledge acquired in the classroom, and creativity and imagination in solving problems. With the help of the complementary material provided by Professor Juliano Ratusznei, it is possible to understand the requirements of the exercise, divide it into parts and arrive at a logical result that satisfies the problem.*

Resumo. Este trabalho visa responder a uma lista de exercícios na questão Estruturas de Dados I, através do conhecimento adquirido em sala de aula, e criatividade e imaginação na resolução de problemas. Com a ajuda do material complementar fornecido pelo professor Juliano Ratusznei, é possível entender os requisitos do exercício, dividi-lo em partes e chegar a um resultado lógico que satisfaça o problema.

Descrição do Exercício 1 a ser solucionado.

1. Escreva um programa para escrever quando inicializado “Olá aluno hoje é sexta-feira”

Resolução do Exercício 1 – Codificação:

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

int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    // aqui o programa irá imprimir a frase.
    printf("Olá aluno hoje é sexta-feira");
    system("pause");
    return 0;
}
```

Execução do Exercício 1 – Resolução

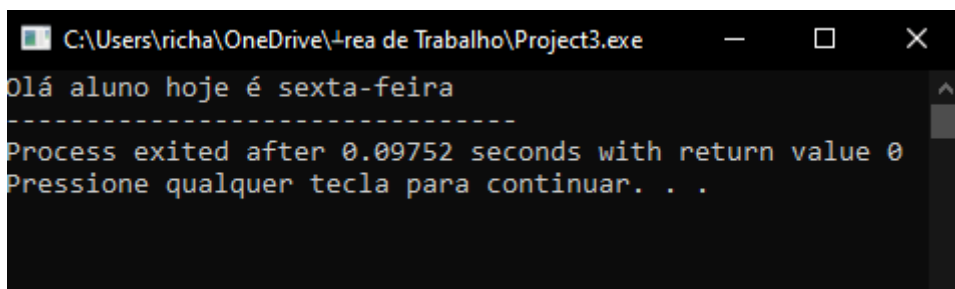


Figura 1: tela da solução do exercício 1.

Descrição da aprendizagem obtida através da problemática:

Através do exercício, é compreendido o funcionamento da função “*printf*” como ferramenta de impressão ou exibição na tela do terminal uma lista formatada de números, caracteres, strings e entre outros.

Descrição do Exercício 2 a ser solucionado.

2. Dado a entrada via teclado do nome de usuário (seu nome) escrever na tela “Bem-vindo nome de usuário”

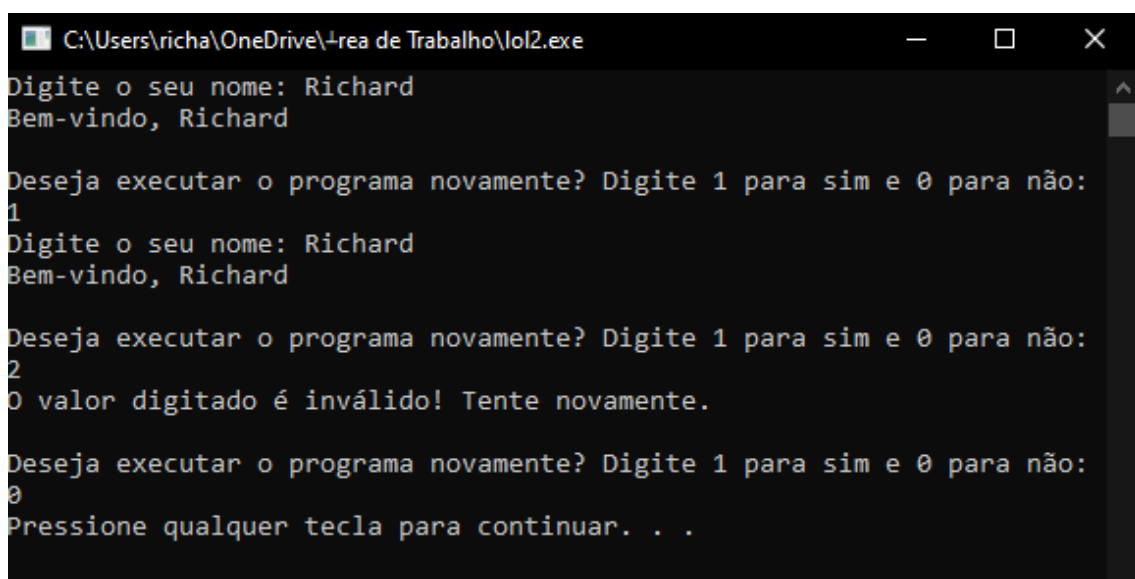
Resolução do Exercício 2 – Codificação:

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

int main(void){
    setlocale(LC_ALL, "Portuguese");
    int i = 1; // criação da variável i para o primeiro loop de repetição do programa
    int sair;
    while (i == 1) { // primeiro loop de repetição do programa
        char nome[99];
        printf("Digite o seu nome: ");
        scanf("%s", &nome); // aqui o usuário dará entrada no nome
        printf("Bem-vindo, %s\n\n", nome); // aqui o programa ira printar o nome do usuário
        i = 0;
        int aux = 1; // criação da variável auxiliar para o segundo loop de repetição do programa
        while (aux == 1) { // segundo loop de repetição do programa
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
            scanf("%d", &sair);
            aux = 0;
            if (sair == 1) {
                i = 1;
            } else if (sair == 0) {
                i = 0;
                break;
            }
        }
    }
}
```

```
    } else {  
        printf("O valor digitado é inválido! Tente novamente. \n\n");  
        aux = 1;  
    }  
    i = 1;  
}  
}  
system("pause");  
return 0;  
}
```

Execução do Exercício 2 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\lol2.exe  
Digite o seu nome: Richard  
Bem-vindo, Richard  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite o seu nome: Richard  
Bem-vindo, Richard  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
2  
O valor digitado é inválido! Tente novamente.  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
0  
Pressione qualquer tecla para continuar. . .
```

Figura 2: tela da solução do exercício 2.

Descrição da aprendizagem obtida através da problemática:

Através do exercício, são compreendidas diversas funções, das quais sendo: estruturas de repetição (*while*, a partir da linha 6), usadas com o propósito de realizar a execução do algoritmo repetidas vezes, vetor de caracteres com posições definidas (*nome[99]*, linha 7), leitura e gravação de dados digitados pelo usuário (*scanf*, a partir da linha 9) e estruturas condicionais (*if* e *else*, a partir da linha 17). Em seguida, o programa pergunta ao usuário se ele deseja executar o programa novamente. O laço principal do programa é um *while* com a condição *i == 1*, o que significa que ele será

executado enquanto i for igual a 1. Dentro desse laço, o programa solicita ao usuário que digite um valor lógico e verifica se o valor é válido (entre 0 e 1). Se o valor for inválido, o programa imprime uma mensagem de erro e solicita que o usuário digite novamente. Caso contrário, o programa calcula o valor lógico oposto correspondente (ou seja, se o valor digitado foi 0, imprime 1, e vice-versa).

Descrição do Exercício 3 a ser solucionado.

3. Dado de entrada um numeral escrever na tela “O número digitado foi: numeral”

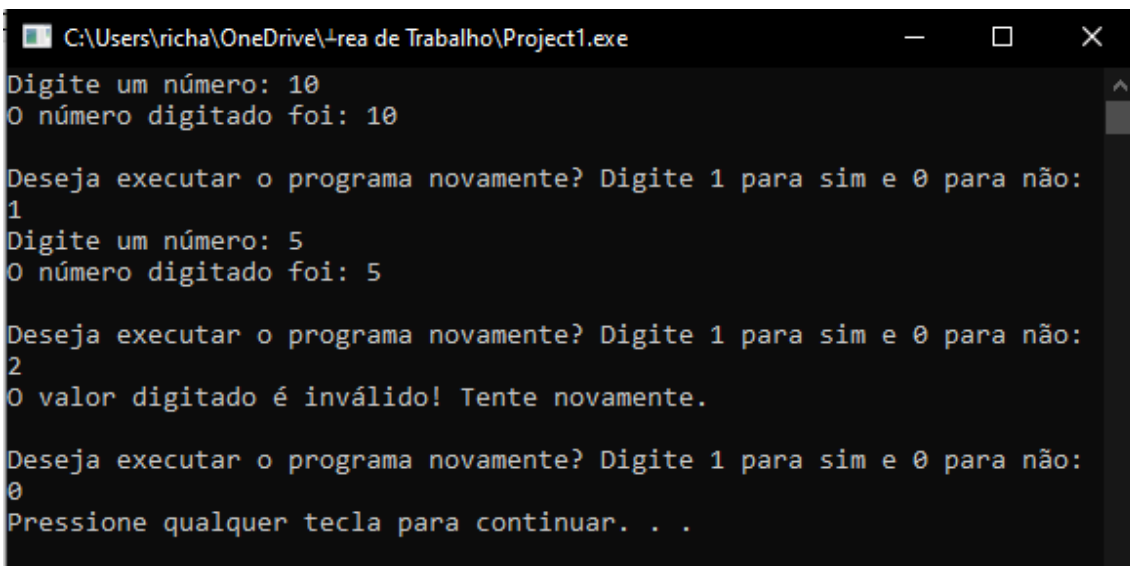
Resolução do Exercício 3 – Codificação:

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

int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    int i=1, sair;
    float num;
    while (i==1){
        printf("Digite um número: ");
        scanf("%f", &num);
        printf("O número digitado foi: %.2f\n\n", num);
        i = 0;
        int aux = 1;
        while (aux == 1) {
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
            scanf("%d", &sair);
            aux = 0;
            if (sair == 1) {
                i = 1;
```

```
} else if (sair == 0) {  
    i = 0;  
    break;  
} else {  
    printf("O valor digitado é inválido! Tente novamente. \n\n");  
    aux = 1;  
}  
i = 1;  
}  
}  
system("pause");  
return 0;  
}
```

Execução do Exercício 3 – Resolução



```
C:\Users\richa\OneDrive\+rea de Trabalho\Project1.exe  
Digite um número: 10  
O número digitado foi: 10  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite um número: 5  
O número digitado foi: 5  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
2  
O valor digitado é inválido! Tente novamente.  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
0  
Pressione qualquer tecla para continuar. . .
```

Figura 3: tela da solução do exercício 3.

Descrição da aprendizagem obtida através da problemática:

Através do exercício, como visto no anterior, são usadas as mesmas estruturas de repetição e condicionais para que haja uma melhor dinâmica no funcionamento do algoritmo, porém, nesse em questão, é feita a leitura a partir da função *scanf* nos valores

inteiros digitados pelo usuário. O programa pergunta ao usuário se ele deseja executar o programa novamente. O laço principal do programa é um *while* com a condição $i == 1$, o que significa que ele será executado enquanto i for igual a 1. Dentro desse laço, o programa solicita ao usuário que digite um valor lógico e verifica se o valor é válido (entre 0 e 1). Se o valor for inválido, o programa imprime uma mensagem de erro e solicita que o usuário digite novamente. Caso contrário, o programa calcula o valor lógico oposto correspondente (ou seja, se o valor digitado foi 0, imprime 1, e vice-versa).

Descrição do Exercício 4 a ser solucionado.

4. Dado de entrada um valor lógico escrever na tela “O valor oposto é: valor lógico”

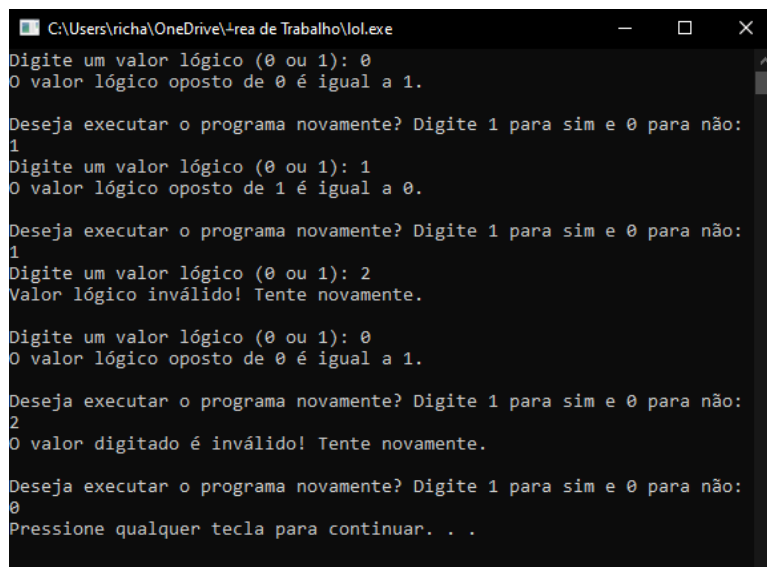
Resolução do Exercício 4 – Codificação:

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

int main(void){
    setlocale(LC_ALL, "Portuguese");
    int num, sair, i = 1;
    while (i == 1) {
        printf("Digite um valor lógico (0 ou 1): ");
        scanf("%d", &num);
        if (num < 0 || num > 1) {
            printf("Valor lógico inválido! Tente novamente.\n\n");
        } else {
            i = 0;
            if (num == 1) {
                printf("O valor lógico oposto de %d é igual a 0.\n\n", num);
            } else {
                printf("O valor lógico oposto de %d é igual a 1.\n\n", num);
            }
        }
    }
}
```

```
}  
  
int aux = 1;  
  
while (aux == 1){  
    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
    \n");  
  
    scanf("%d", &sair);  
  
    if (sair == 1){  
        aux = 0;  
  
        i = 1;  
  
    }  
  
    else if(sair == 0){  
        aux = 0;  
  
        i = 0;  
  
        break;  
  
    }  
  
    else{  
        printf("O valor digitado é inválido! Tente novamente. \n\n");  
  
    }  
  
    }  
  
    }  
  
    system("pause");  
  
    return 0;  
  
}
```

Execução do Exercício 4 – Resolução (próxima página).



```
C:\Users\richa\OneDrive\Área de Trabalho\lol.exe
Digite um valor lógico (0 ou 1): 0
O valor lógico oposto de 0 é igual a 1.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite um valor lógico (0 ou 1): 1
O valor lógico oposto de 1 é igual a 0.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite um valor lógico (0 ou 1): 2
Valor lógico inválido! Tente novamente.

Digite um valor lógico (0 ou 1): 0
O valor lógico oposto de 0 é igual a 1.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
2
O valor digitado é inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
0
Pressione qualquer tecla para continuar. . .
```

Figura 4: tela da solução do exercício 4.

Descrição da aprendizagem obtida através da problemática:

Este programa tem como objetivo pedir para o usuário digitar um valor lógico (0 ou 1), verificar se o valor digitado é válido e, em seguida, imprimir o valor lógico oposto correspondente. Em seguida, o programa pergunta ao usuário se ele deseja executar o programa novamente. O laço principal do programa é um *while* com a condição $i == 1$, o que significa que ele será executado enquanto i for igual a 1. Dentro desse laço, o programa solicita ao usuário que digite um valor lógico e verifica se o valor é válido (entre 0 e 1). Se o valor for inválido, o programa imprime uma mensagem de erro e solicita que o usuário digite novamente. Caso contrário, o programa calcula o valor lógico oposto correspondente (ou seja, se o valor digitado foi 0, imprime 1, e vice-versa).

Descrição do Exercício 5 a ser solucionado.

5. Dado de entrada dois valores numéricos fazer a subtração e apresentar ao usuário.

Resolução do Exercício 5 – Codificação:

```
#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

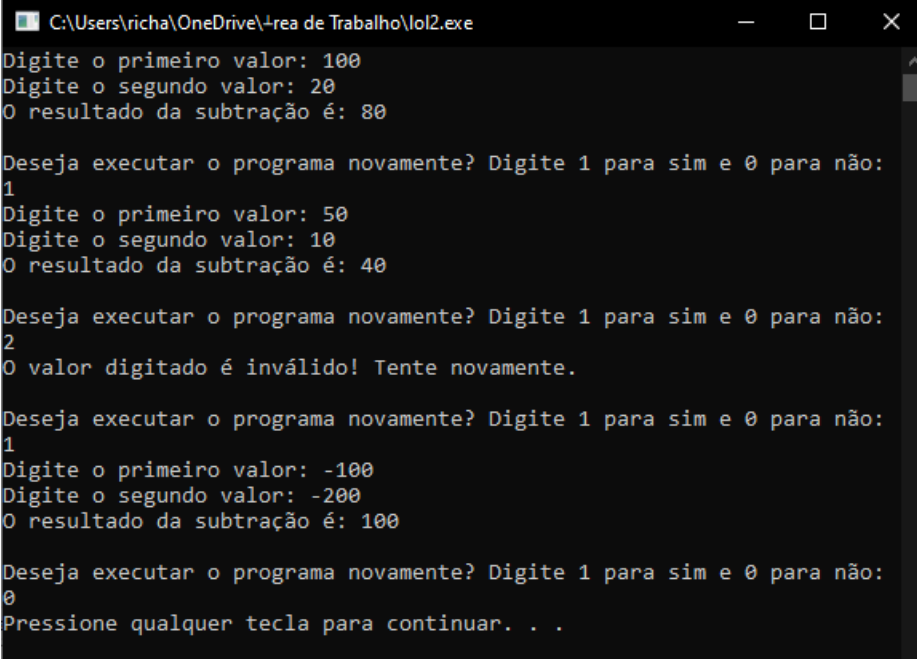
int main(void){

    setlocale(LC_ALL, "Portuguese");
```

```
int i = 1, sair;
float num1, num2, resultado;
while (i==1){
    printf("Digite o primeiro valor: ");
    scanf("%f", &num1);
    printf("Digite o segundo valor: ");
    scanf("%f", &num2);
    resultado = (num1-num2);
    printf("O resultado da subtração de %.2f por %.2f é igual a %.2f\n\n", num1, num2,
resultado);
        int aux = 1;
        while (aux == 1){
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
\n");
            scanf("%d", &sair);
            if (sair == 1){
                aux = 0;
                i = 1;
            }
            else if(sair == 0){
                aux = 0;
                i = 0;
                break;
            }
            else{
                printf("O valor digitado é inválido! Tente novamente. \n\n");
            }
        }
    }
}
system("pause");
```

```
return 0;  
}
```

Execução do Exercício 5 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\lol2.exe  
Digite o primeiro valor: 100  
Digite o segundo valor: 20  
O resultado da subtração é: 80  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite o primeiro valor: 50  
Digite o segundo valor: 10  
O resultado da subtração é: 40  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
2  
O valor digitado é inválido! Tente novamente.  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite o primeiro valor: -100  
Digite o segundo valor: -200  
O resultado da subtração é: 100  
  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
0  
Pressione qualquer tecla para continuar. . .
```

Figura 5: tela da solução do exercício 5.

Descrição da aprendizagem obtida através da problemática:

Este programa realiza a subtração entre dois números (*num1* e *num2*) e imprime o resultado na tela. Após cada cálculo, o usuário é questionado se deseja executar o programa novamente. Se o usuário digitar "1", o programa é executado novamente. Caso contrário, o programa é encerrado. O laço "*while (i == 1)*" é responsável por garantir que o programa seja executado pelo menos uma vez, pois a variável "*i*" é inicializada com o valor 1. Já o laço "*while (aux == 1)*" é responsável por verificar se o valor dado como entrada é válido e assim executar o programa novamente. Enquanto a variável "*aux*" for igual a 1, o usuário será questionado e o programa continuará executando. Se o usuário digitar um valor diferente de 0 ou 1, o programa informa que o valor é inválido e solicita que o usuário digite novamente.

Descrição do Exercício 6 a ser solucionado.

6. Escreva um programa que imprima os 10 primeiros números negativos.

Resolução do Exercício 6 – Codificação:

```
#include <stdio.h>

#include <stdlib.h>

int main(void){

    int i;

    for (i=-1; i> -11; i--){

        printf("%d\n", i);

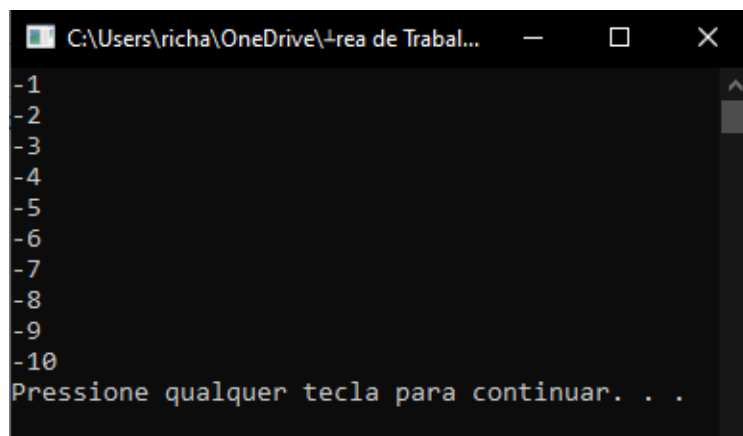
    }

    system("pause");

    return 0;

}
```

Execução do Exercício 6 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabal...
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
Pressione qualquer tecla para continuar. . .
```

Figura 6: tela da solução do exercício 6.

Descrição da aprendizagem obtida através da problemática:

Este programa usa um laço de repetição *for* para imprimir na tela uma sequência decrescente de números inteiros, iniciando em -1 e terminando em -10. A variável *i* é inicializada em -1 e, a cada iteração, é decrementada em 1 até atingir o valor -10. Dentro do corpo do laço, é utilizada a função *printf* para imprimir o valor atual de *i* na tela, seguido por uma quebra de linha. O programa finaliza com a chamada da função *system("pause")*, que pausa a execução do programa até que o usuário pressione uma tecla, e retorna 0 para indicar que a execução do programa foi concluída com sucesso.

Descrição do Exercício 7 a ser solucionado.

7. Dado um número verificar se ele é maior que 10.

Resolução do Exercício 7 – Codificação:

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

int main(void) {
    setlocale(LC_ALL, "Portuguese");
    int i = 1, sair;
    float num = 0;
    while (i == 1) {
        printf("Digite um número para verificar se o mesmo é maior do que 10: ");
        scanf("%f", &num);
        if (num > 10) {
            printf("\nO número digitado é maior do que 10.\n");
        } else if (num == 10) {
            printf("\nO número digitado é igual a 10.\n");
        } else {
            printf("\nO número digitado é menor do que 10.\n");
        }
        int aux = 1;
        while (aux == 1) {
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
            scanf("%d", &sair);
            if (sair == 1) {
                aux = 0;
            } else if (sair == 0) {
```

```
    aux = 0;

    i = 0;

} else {

    printf("O valor digitado é inválido! Tente novamente. \n\n");

}

}

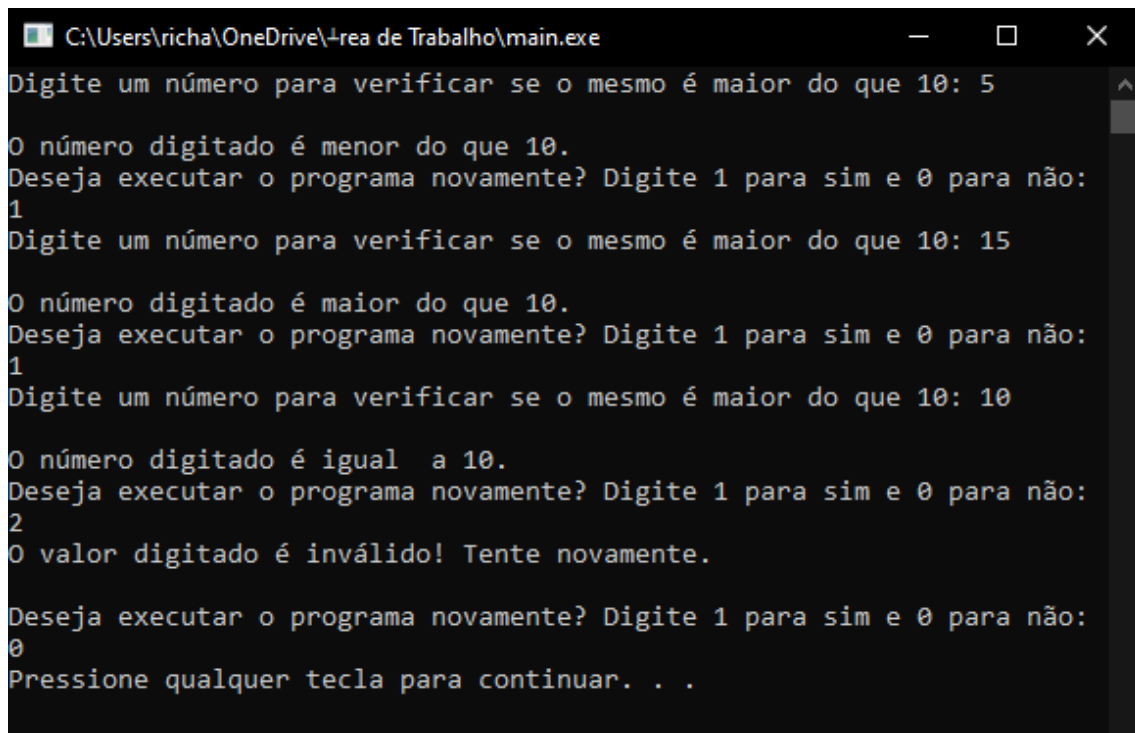
}

system("pause");

return 0;

}
```

Execução do Exercício 7 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite um número para verificar se o mesmo é maior do que 10: 5
O número digitado é menor do que 10.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite um número para verificar se o mesmo é maior do que 10: 15
O número digitado é maior do que 10.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite um número para verificar se o mesmo é maior do que 10: 10
O número digitado é igual a 10.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
2
O valor digitado é inválido! Tente novamente.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
0
Pressione qualquer tecla para continuar. . .
```

Figura 7: tela da solução do exercício 7.

Descrição da aprendizagem obtida através da problemática:

Este programa em C solicita que o usuário insira um número e verifica se o número é maior, igual ou menor do que 10. Em seguida, o programa pergunta ao usuário se deseja

executar o programa novamente e repete o processo se o usuário inserir 1 para "sim" ou termina o programa se o usuário inserir 0 para "não". O programa começa definindo a localidade para "Portuguese" com a função `setlocale`. Em seguida, declara as variáveis *i* e *sair* como inteiros e *num* como um número real (*float*).

Descrição do Exercício 8 a ser solucionado.

8. Dado um número verificar se ele é positivo ou negativo.

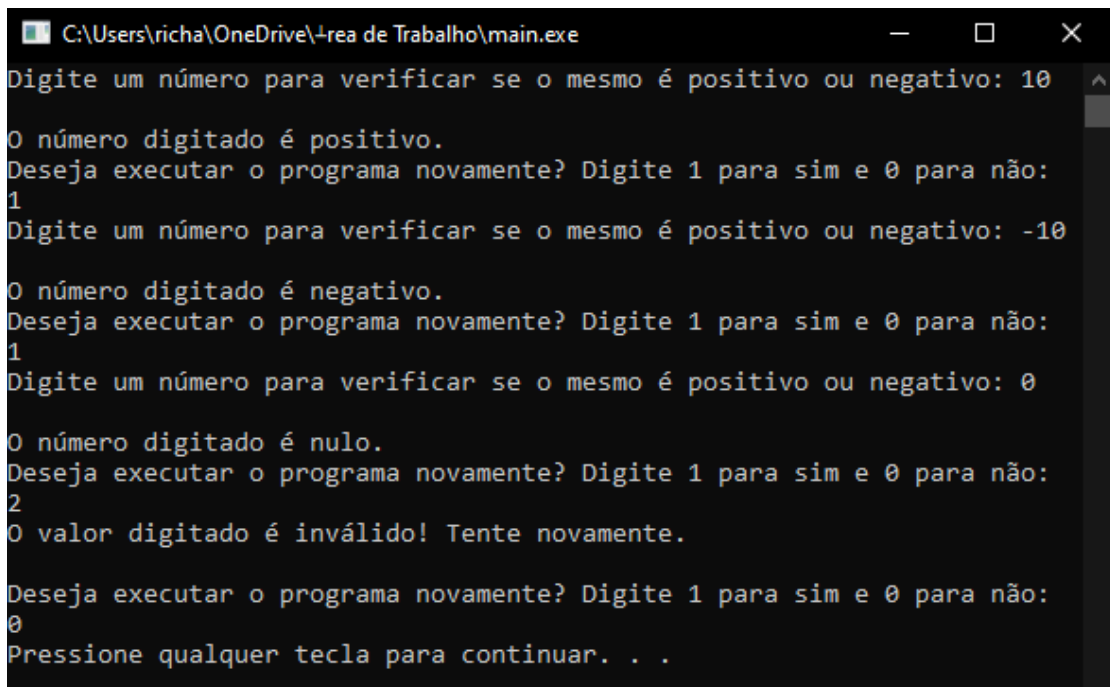
Resolução do Exercício 8 – Codificação:

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

int main(void) {
    setlocale(LC_ALL, "Portuguese");
    int i = 1, sair;
    float num = 0;
    while (i == 1) {
        printf("Digite um número para verificar se o mesmo é positivo ou negativo: ");
        scanf("%f", &num);
        if (num > 0) {
            printf("\nO número digitado é positivo.\n");
        } else if (num == 0) {
            printf("\nO número digitado é nulo.\n");
        } else {
            printf("\nO número digitado é negativo.\n");
        }
        int aux = 1;
        while (aux == 1) {
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
            scanf("%d", &sair);
```

```
if (sair == 1) {  
    aux = 0;  
} else if (sair == 0) {  
    aux = 0;  
    i = 0;  
} else {  
    printf("O valor digitado é inválido! Tente novamente. \n\n");  
}  
}  
}  
system("pause");  
return 0;  
}
```

Execução do Exercício 8 – Resolução



```
C:\Users\richa\OneDrive\Trabalho\main.exe  
Digite um número para verificar se o mesmo é positivo ou negativo: 10  
O número digitado é positivo.  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite um número para verificar se o mesmo é positivo ou negativo: -10  
O número digitado é negativo.  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
1  
Digite um número para verificar se o mesmo é positivo ou negativo: 0  
O número digitado é nulo.  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
2  
O valor digitado é inválido! Tente novamente.  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:  
0  
Pressione qualquer tecla para continuar. . .
```

Figura 8: tela da solução do exercício 8.

Descrição da aprendizagem obtida através da problemática:

O programa solicita ao usuário que digite um número e, em seguida, verifica se o número é positivo, negativo ou nulo. Se o número for maior que 0, o programa exibe a mensagem "O número digitado é positivo." Se o número for menor que 0, o programa exibe a mensagem "O número digitado é negativo." E se o número for igual a 0, o programa exibe a mensagem "O número digitado é nulo." Em seguida, o programa pergunta se o usuário deseja executar o programa novamente ou sair. Se o usuário digitar 1, o programa reinicia e pede que o usuário digite outro número. Se o usuário digitar 0, o programa encerra.

Descrição do Exercício 9 a ser solucionado.

9. Dados dois números diferentes dividir o maior pelo menor valor.

Resolução do Exercício 9 – Codificação

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

int main(void) {
    setlocale(LC_ALL, "Portuguese");
    int i = 1, sair;
    float num1, num2, maior, menor, resultado;
    while (i == 1) {
        printf("Digite o primeiro número a ser dividido pelo maior valor: \n");
        scanf("%f", &num1);
        printf("Digite o segundo número a ser dividido pelo maior valor: \n");
        scanf("%f", &num2);
        if (num1 > num2){
            maior = num1;
            menor = num2;
        }
        else{
```

```
    maior = num2;

    menor = num1;
}

resultado = maior/menor;

printf("O resultado da divisão de %.2f por %.2f é igual a %.2f.\n", maior, menor,
resultado);

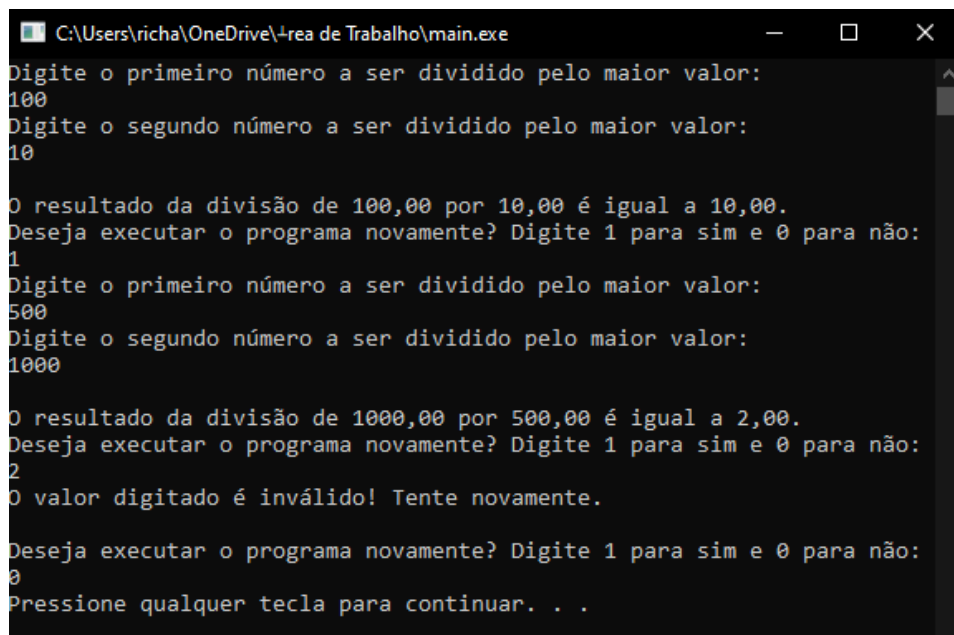
int aux = 1;
while (aux == 1) {
    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
    scanf("%d", &sair);
    if (sair == 1) {
        aux = 0;
    } else if (sair == 0) {
        aux = 0;
        i = 0;
    } else {
        printf("O valor digitado é inválido! Tente novamente. \n\n");
    }
}

}

system("pause");

return 0;
}
```

Execução do Exercício 9 – Resolução (próxima página).



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite o primeiro número a ser dividido pelo maior valor:
100
Digite o segundo número a ser dividido pelo maior valor:
10

O resultado da divisão de 100,00 por 10,00 é igual a 10,00.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite o primeiro número a ser dividido pelo maior valor:
500
Digite o segundo número a ser dividido pelo maior valor:
1000

O resultado da divisão de 1000,00 por 500,00 é igual a 2,00.
Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
2
O valor digitado é inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
0
Pressione qualquer tecla para continuar. . .
```

Figura 9: tela da solução do exercício 9.

Descrição da aprendizagem obtida através da problemática:

A princípio, este programa recebe dois números, compara os valores de *num1* e *num2* e armazena o maior na variável maior e o menor na variável menor, e por fim, divide o maior pelo menor, exibindo o resultado da divisão na tela. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 10 a ser solucionado.

10. Dado a formula $Y = X + 5$ escrever o valor de Y quando dado uma entrada de X.

Resolução do Exercício 10 – Codificação:

```
#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

int main(void) {

    setlocale(LC_ALL, "Portuguese");

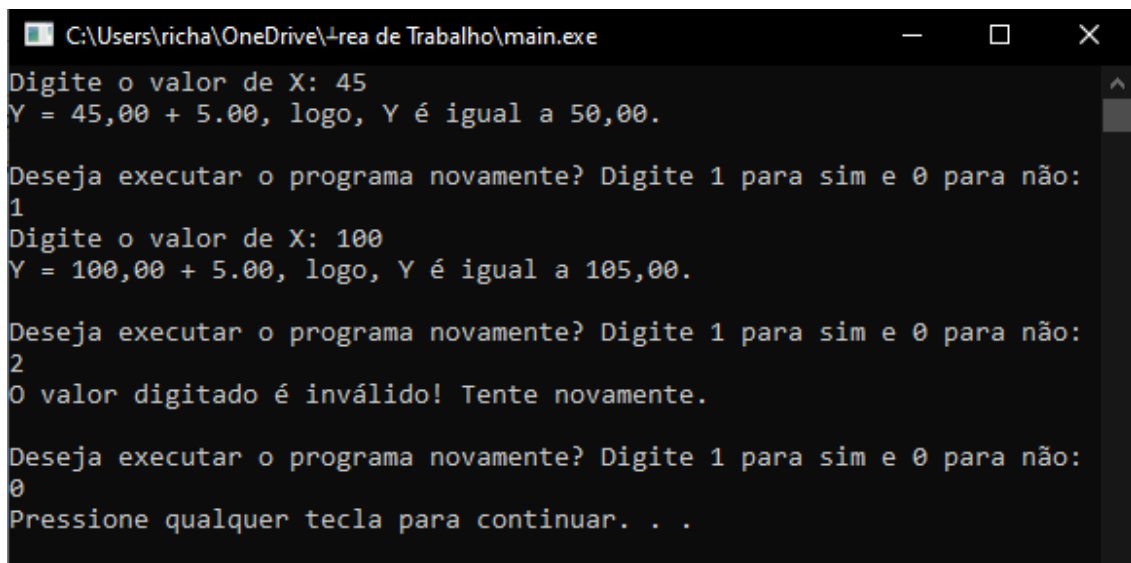
    int i = 1, sair;

    float x, y;

    while (i == 1) {
```

```
printf("Digite o valor de X: ");
scanf("%f", &x);
y = x + 5;
printf("Y = %.2f + 5.00, logo, Y é igual a %.2f.\n\n", x, y);
int aux = 1;
while (aux == 1) {
    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");
    scanf("%d", &sair);
    if (sair == 1) {
        aux = 0;
    } else if (sair == 0) {
        aux = 0;
        i = 0;
    } else {
        printf("O valor digitado é inválido! Tente novamente. \n\n");
    }
}
system("pause");
return 0;
}
```

Execução do Exercício 10 – Resolução (próxima página).



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite o valor de X: 45
Y = 45,00 + 5.00, logo, Y é igual a 50,00.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite o valor de X: 100
Y = 100,00 + 5.00, logo, Y é igual a 105,00.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
2
0 valor digitado é inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
0
Pressione qualquer tecla para continuar. . .
```

Figura 10: tela da solução do exercício 10.

Descrição da aprendizagem obtida através da problemática:

Esse programa solicita ao usuário que insira um valor para x , calcula y como $x + 5$ e imprime o valor de y na tela. A partir deste programa, é compreendido o uso de operações aritméticas em um algoritmo. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 11 a ser solucionado.

11. Dado a formula $media = n1+n2+n3/3$ escrever o valor do resultado ao usuário quando ele digitar o valor de $n1$, $n2$ e $n3$.

Resolução do Exercício 11 – Codificação:

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

int main(void) {
    setlocale(LC_ALL, "Portuguese");
    int i = 1, sair;
    float n1, n2, n3, media;
    while (i == 1) {
        printf("Digite o primeiro valor a ser calculado a media: \n");
```

```
scanf("%f", &n1);

printf("Digite o segundo valor a ser calculado a media: \n");

scanf("%f", &n2);

printf("Digite o terceiro valor a ser calculado a media: \n");

scanf("%f", &n3);

media = (n1 + n2 + n3) / 3;

printf("\nA média dos valores de %.2f, %.2f e %.2f é igual a %.2f.\n\n", n1,
      n2, n3, media);

int aux = 1;

while (aux == 1) {

    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: \n");

    scanf("%d", &sair);

    if (sair == 1) {

        aux = 0;

    } else if (sair == 0) {

        aux = 0;

        i = 0;

    } else {

        printf("O valor digitado é inválido! Tente novamente. \n\n");

    }

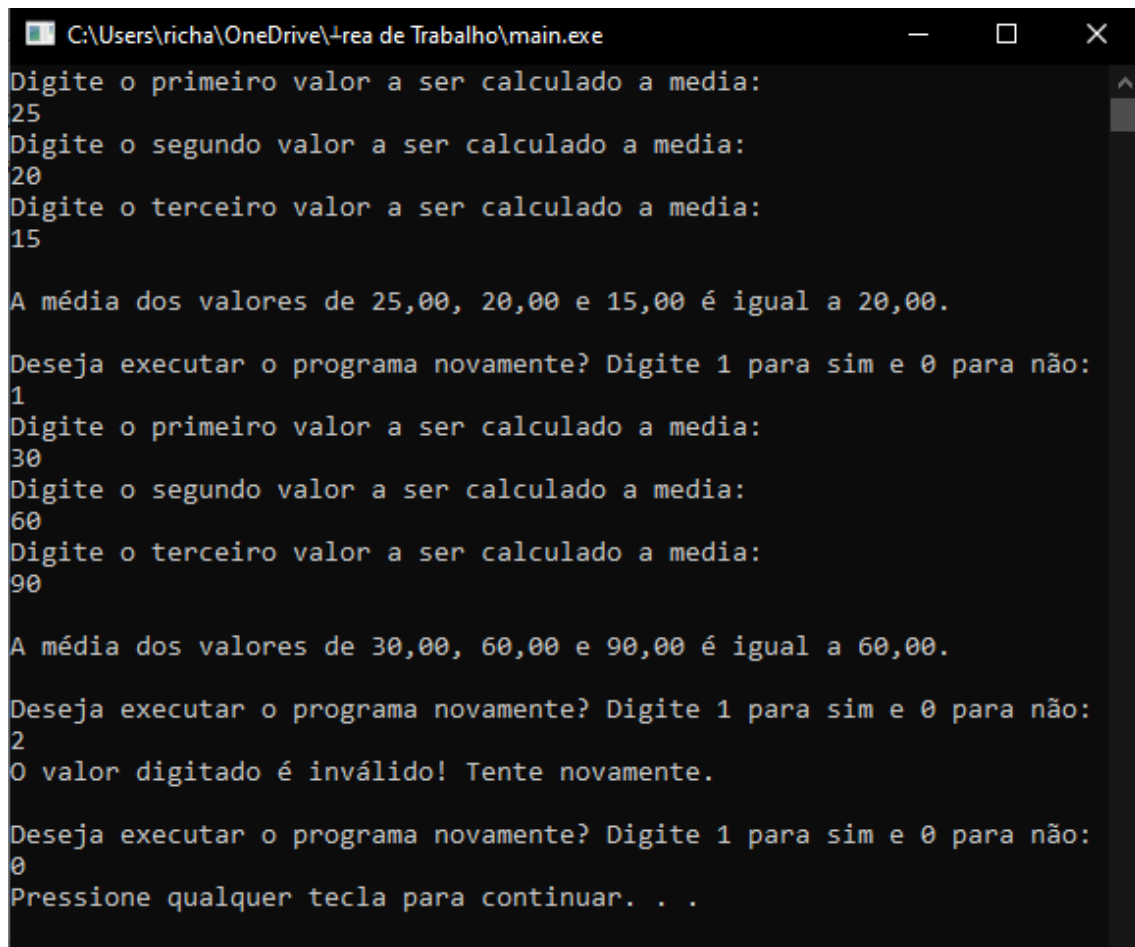
}

system("pause");

return 0;

}
```

Execução do Exercício 11 – Resolução (próxima página).



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite o primeiro valor a ser calculado a media:
25
Digite o segundo valor a ser calculado a media:
20
Digite o terceiro valor a ser calculado a media:
15

A média dos valores de 25,00, 20,00 e 15,00 é igual a 20,00.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
1
Digite o primeiro valor a ser calculado a media:
30
Digite o segundo valor a ser calculado a media:
60
Digite o terceiro valor a ser calculado a media:
90

A média dos valores de 30,00, 60,00 e 90,00 é igual a 60,00.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
2
O valor digitado é inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não:
0
Pressione qualquer tecla para continuar. . .
```

Figura 11: tela da solução do exercício 11.

Descrição da aprendizagem obtida através da problemática:

Este programa em C solicita ao usuário três valores, calcula a média desses valores e a exibe na tela. As variáveis `n1`, `n2`, `n3` e `media` são definidas para armazenar os valores inseridos pelo usuário e a média calculada. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 12 a ser solucionado.

12. Dado a formula $A/5 = B/(A+2)$ calcular o valor de B dado uma entrada de A.

Resolução do Exercício 12 – Codificação:

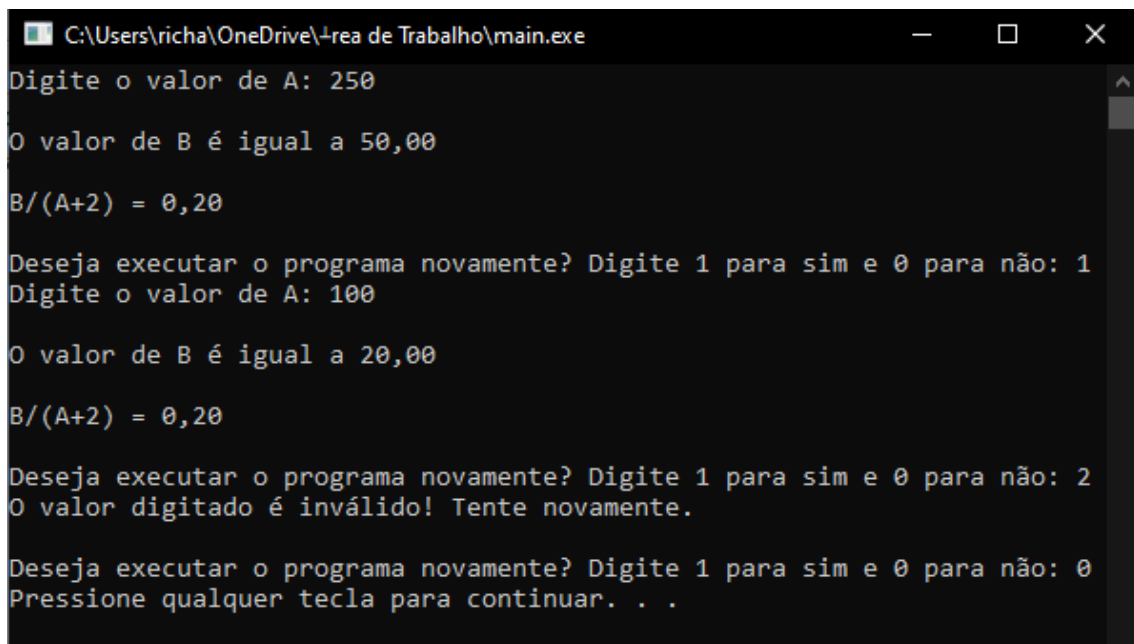
```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>
```

```
int main(void){
    setlocale(LC_ALL, "Portuguese");
    int i = 1, sair;
    float num1, num2, resultado;
    while (i==1){
        float a, b, resultado;
        printf("Digite o valor de A: ");
        scanf("%f", &a);
        b = a / 5;
        printf("\nO valor de B é igual a %0.2f\n", b);
        resultado = b / (a + 2);
        printf("\nB/(A+2) = %0.2f\n\n", resultado);
        int aux = 1;
        while (aux == 1){
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: ");
            scanf("%d", &sair);
            if (sair == 1){
                aux = 0;
                i = 1;
            }
            else if(sair == 0){
                aux = 0;
                i = 0;
                break;
            }
            else{
                printf("O valor digitado é inválido! Tente novamente. \n\n");
            }
        }
    }
}
```



```
}  
system("pause");  
return 0;  
}
```

Execução do Exercício 12 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe  
Digite o valor de A: 250  
O valor de B é igual a 50,00  
B/(A+2) = 0,20  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1  
Digite o valor de A: 100  
O valor de B é igual a 20,00  
B/(A+2) = 0,20  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 2  
O valor digitado é inválido! Tente novamente.  
Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 0  
Pressione qualquer tecla para continuar. . .
```

Figura 12: tela da solução do exercício 12.

Descrição da aprendizagem obtida através da problemática:

Este programa solicita que o usuário digite um valor para a variável a , calcula o valor de b como a divisão de a por 5, e em seguida calcula o resultado como a divisão de b por $a+2$. O resultado é exibido na tela e o usuário tem a opção de executar o programa novamente ou não.

Descrição do Exercício 13 a ser solucionado.

13. Escreva um programa que dados 15 números, imprima seus quadrados, obs: você não precisa necessariamente imprimir no final.

Resolução do Exercício 13 – Codificação:

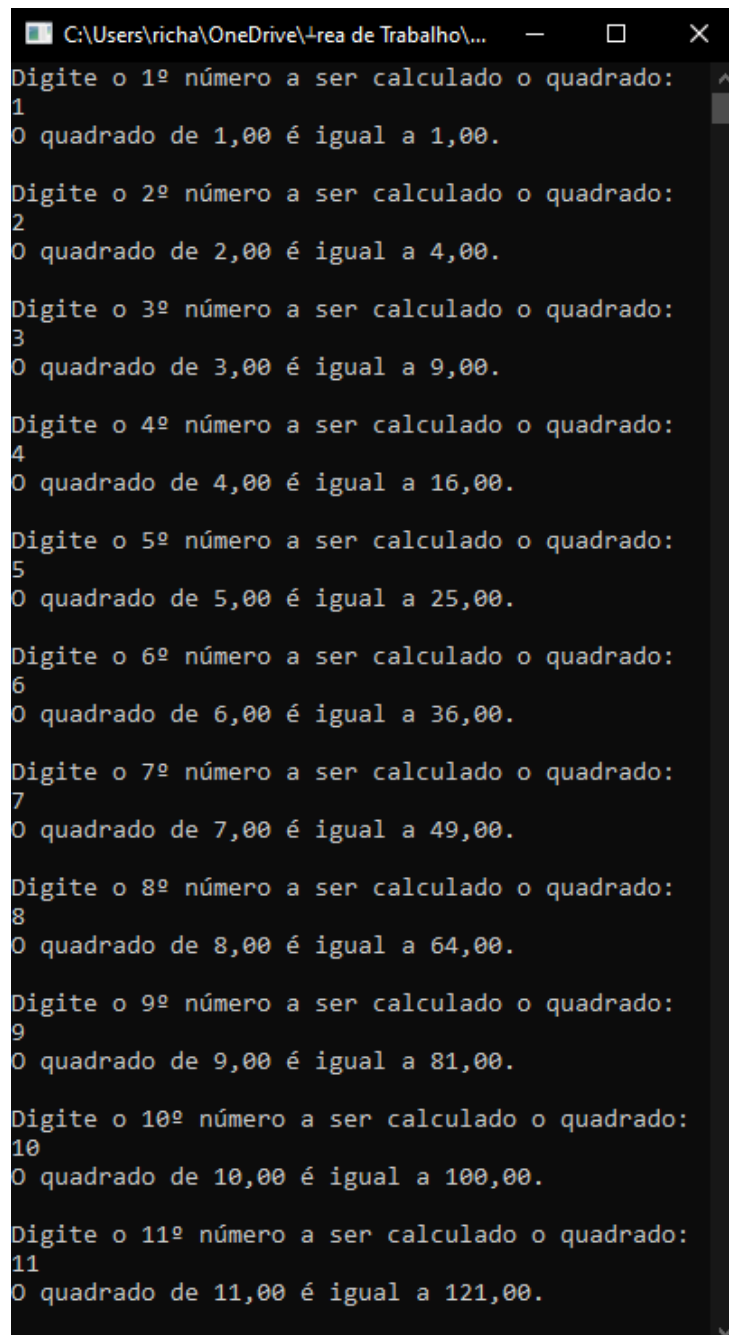
```
#include <stdio.h>  
  
#include <stdlib.h>
```

```
#include <locale.h>

int main(void){
    setlocale(LC_ALL, "Portuguese");
    int a, i = 1, sair;
    float n[15], r[15];
    while (i==1){
        for (a = 1; a <= 15; a++){
            printf("Digite o %dº número a ser calculado o quadrado: \n", a);
            scanf("%f", &n[a]);
            printf("O quadrado de %.2f é igual a %.2f.\n\n", n[a], n[a]*n[a]);
        }
        int aux = 1;
        while (aux == 1){
            printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: ");
            scanf("%d", &sair);
            if (sair == 1){
                aux = 0;
                i = 1;
            }
            else if(sair == 0){
                aux = 0;
                i = 0;
                break;
            }
            else{
                printf("O valor digitado é inválido! Tente novamente. \n\n");
            }
        }
    }
}
```

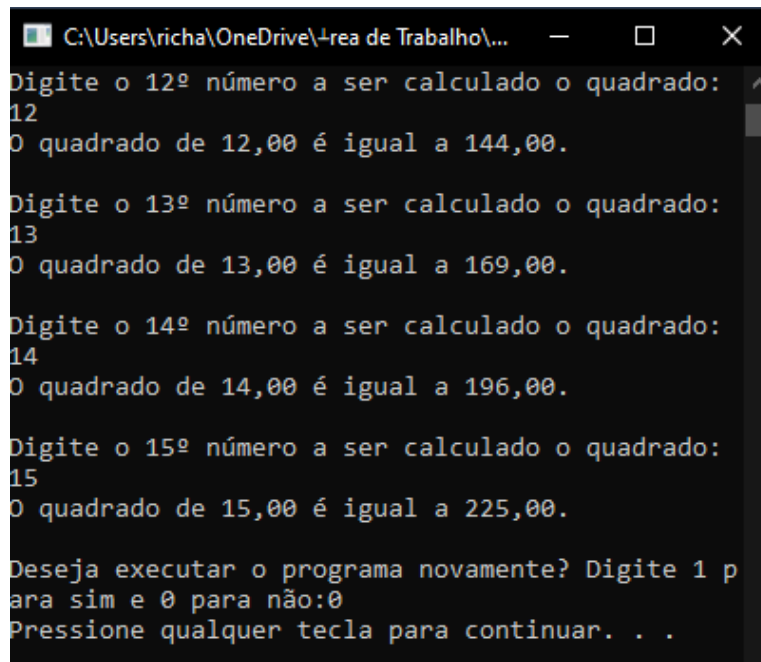
```
}  
system("pause");  
return 0;  
}
```

Execução do Exercício 13 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\...  
Digite o 1º número a ser calculado o quadrado:  
1  
O quadrado de 1,00 é igual a 1,00.  
  
Digite o 2º número a ser calculado o quadrado:  
2  
O quadrado de 2,00 é igual a 4,00.  
  
Digite o 3º número a ser calculado o quadrado:  
3  
O quadrado de 3,00 é igual a 9,00.  
  
Digite o 4º número a ser calculado o quadrado:  
4  
O quadrado de 4,00 é igual a 16,00.  
  
Digite o 5º número a ser calculado o quadrado:  
5  
O quadrado de 5,00 é igual a 25,00.  
  
Digite o 6º número a ser calculado o quadrado:  
6  
O quadrado de 6,00 é igual a 36,00.  
  
Digite o 7º número a ser calculado o quadrado:  
7  
O quadrado de 7,00 é igual a 49,00.  
  
Digite o 8º número a ser calculado o quadrado:  
8  
O quadrado de 8,00 é igual a 64,00.  
  
Digite o 9º número a ser calculado o quadrado:  
9  
O quadrado de 9,00 é igual a 81,00.  
  
Digite o 10º número a ser calculado o quadrado:  
10  
O quadrado de 10,00 é igual a 100,00.  
  
Digite o 11º número a ser calculado o quadrado:  
11  
O quadrado de 11,00 é igual a 121,00.
```

Figura 13-1: tela da solução do exercício 13.



```
C:\Users\richa\OneDrive\Área de Trabalho\...
Digite o 12º número a ser calculado o quadrado:
12
O quadrado de 12,00 é igual a 144,00.

Digite o 13º número a ser calculado o quadrado:
13
O quadrado de 13,00 é igual a 169,00.

Digite o 14º número a ser calculado o quadrado:
14
O quadrado de 14,00 é igual a 196,00.

Digite o 15º número a ser calculado o quadrado:
15
O quadrado de 15,00 é igual a 225,00.

Deseja executar o programa novamente? Digite 1 p
ara sim e 0 para não:0
Pressione qualquer tecla para continuar. . .
```

Figura 13-2: tela da solução do exercício 13.

Descrição da aprendizagem obtida através da problemática:

Esse programa solicita que o usuário digite 15 números e, em seguida, calcula e exibe o quadrado de cada um deles. O programa utiliza um loop *for* para iterar 15 vezes e pedir ao usuário que insira um número em cada iteração. Em seguida, o programa calcula o quadrado do número utilizando a fórmula $n[a]*n[a]$ e imprime o resultado na tela. A partir deste programa, é compreendido o uso de vetores estáticos em variáveis para alocação de valores. O resultado é exibido na tela e o usuário tem a opção de executar o programa novamente ou não.

Descrição do Exercício 14 a ser solucionado.

14. Escreva um programa que dado o primeiro número e a razão de uma sequência, imprima seus dez primeiros termos: você não precisa necessariamente imprimir no final.

Resolução do Exercício 14 – Codificação:

```
#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main(void){
```

```
setlocale(LC_ALL, "Portuguese");

int num, x, r, a, i = 1, sair;

while (i==1){

printf("Digite o primeiro termo da progressão: ");

scanf("%d", &num);

printf("Digite a razão da progressão: ");

scanf("%d", &r);

    for (x = 0; x < 10; x++){

printf("%d\t", num);

    num=num+r;

}

printf("\n\n");

int aux=1;

while (aux==1){

    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: ");

    scanf("%d", &sair);

    if (sair == 1){

        aux = 0;

    }

    else if (sair == 0){

        aux = 0;

        i = 0;

    }

    else{

        printf ("Valor inválido! Tente novamente.\n\n");

        aux = 1;

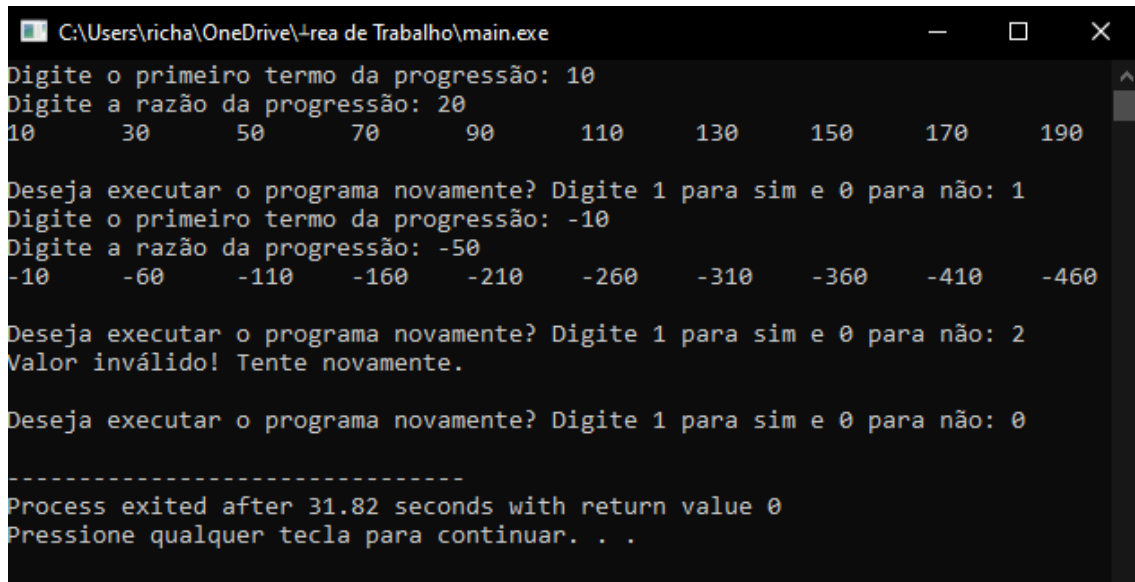
    }

}

}
```

```
}
```

Execução do Exercício 14 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite o primeiro termo da progressão: 10
Digite a razão da progressão: 20
10    30    50    70    90    110    130    150    170    190

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite o primeiro termo da progressão: -10
Digite a razão da progressão: -50
-10   -60   -110  -160  -210  -260  -310  -360  -410  -460

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 2
Valor inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 0

-----
Process exited after 31.82 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 14: tela da solução do exercício 14.

Descrição da aprendizagem obtida através da problemática:

O programa acima implementa uma sequência numérica em progressão aritmética. O usuário é solicitado a fornecer o primeiro termo da sequência e a razão da progressão. Utilizando um loop de repetição *for*, o programa calcula e exibe os 10 primeiros termos da sequência. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 15 a ser solucionado.

15. Escreva um programa que imprima na tela o fatorial de um programa recebido

Ex: $5! = 5 \times 4 \times 3 \times 2 \times 1$

Ex: $0! = 1$

Resolução do Exercício 15 – Codificação:

```
#include <locale.h>

#include <stdio.h>

#include <stdlib.h>

int main() {
```

```
setlocale(LC_ALL, "Portuguese");

int aux, i, n, r, saida, aux1 = 1;

while (aux1 == 1) {
    n = 1, r = 1;

    printf("Digite o número a ser calculado o fatorial: ");

    scanf("%d", &n);

    if (n < 0) {
        printf("O valor digitado é inválido! Tente novamente. \n\n");
    } else if (n == 0) {
        printf("0! = 1\n");

        printf("O fatorial de 0 é igual a 1.\n\n");
    } else {
        aux = n;

        printf("%d! = ", n);

        for (i = n; i >= 1; i--) {
            r = (r * i);

            printf("%d", i);

            if (i != 1) {
                printf(" * ");
            }
        }

        printf(" = %d", r);

        printf("\nO fatorial de %d é igual a %d.\n\n", aux, r);
    }

    int aux2 = 1;

    while (aux2 == 1) {
        printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para "
            "não: ");

        scanf("%d", &saida);
    }
}
```

```

    aux2 = 0;
    if (saida == 1) {
        continue;
    } else if (saida == 0) {
        aux1 = 0;
    } else {
        printf("O valor digitado é inválido! Tente novamente. \n\n");
        aux2 = 1;
    }
}
}
return 0;
}

```

Execução do Exercício 15 – Resolução

```

C:\Users\richa\OneDrive\Trabalho\main.exe
Digite o número a ser calculado o fatorial: 5
5! = 5 * 4 * 3 * 2 * 1 = 120
O fatorial de 5 é igual a 120.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite o número a ser calculado o fatorial: 10
10! = 10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1 = 3628800
O fatorial de 10 é igual a 3628800.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite o número a ser calculado o fatorial: 0
0! = 1
O fatorial de 0 é igual a 1.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 2
O valor digitado é inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 0
-----
Process exited after 13.35 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

Figura 15: tela da solução do exercício 15.

Descrição da aprendizagem obtida através da problemática:

O programa em questão lê um número inteiro digitado pelo usuário e calcula o seu fatorial. Caso o número digitado seja negativo, o programa exibe uma mensagem de erro. Caso o número seja igual a zero, o programa exibe o resultado 1 para o fatorial. Caso o número seja positivo, utilizando um loop de repetição *for*, o programa calcula o fatorial e exibe o resultado. A partir deste programa, é compreendido o uso de *recursividade* em C. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 16 a ser solucionado.

16. Escreva um programa que dado um número, ele diz se é um número primo ou não.

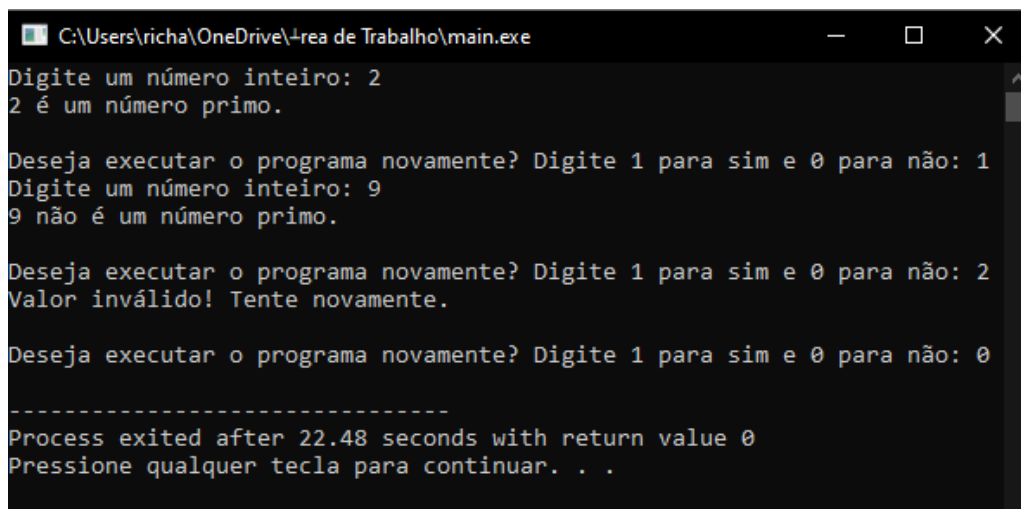
Resolução do Exercício 16 – Codificação:

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

int main() {
    setlocale(LC_ALL, "Portuguese");
    int n, x, sair, primo = 0, i = 1;
    while (i == 1) {
        n = 0, primo = 0;
        printf("Digite um número inteiro: ");
        scanf("%d", &n);
        for (x = 2; x <= n / 2; ++x) {
            if (n % x == 0) {
                primo = 1;
                break;
            }
        }
        if (n == 1) {
```

```
printf("1 não é um número primo.\n\n");
} else {
    if (primo == 0)
        printf("%d é um número primo.\n\n", n);
    else
        printf("%d não é um número primo.\n\n", n);
}
int aux = 1;
while (aux == 1) {
    printf("Deseja executar o programa novamente? Digite 1 para sim e 0 para não: ");
    scanf("%d", &sair);
    if (sair == 1) {
        aux = 0;
    } else if (sair == 0) {
        aux = 0;
        i = 0;
    } else {
        printf("Valor inválido! Tente novamente.\n\n");
        aux = 1;
    }
}
}
return 0;
}
```

Execução do Exercício 16 – Resolução (próxima página).



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Digite um número inteiro: 2
2 é um número primo.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite um número inteiro: 9
9 não é um número primo.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 2
Valor inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 0

-----
Process exited after 22.48 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 16: tela da solução do exercício 16.

Descrição da aprendizagem obtida através da problemática:

Este programa recebe um número inteiro digitado pelo usuário e verifica se ele é um número primo ou não. O programa, então, entra em um loop *for* que começa em 2 e termina em $n/2$. Ele verifica se n é divisível por algum número x no intervalo de 2 a $n/2$. Se n é divisível por algum número x , a variável *primo* é definida como 1 e o loop *for* é interrompido com o comando *break*. Após o loop *for*, o programa verifica se o número digitado é igual a 1. Se for, imprime "*1 não é um número primo*". Caso contrário, ele verifica o valor da variável *primo*. Se *primo* for igual a 0, o número digitado é um número primo e a mensagem "*n é um número primo*" é exibida na tela. Caso contrário, o número digitado não é um número primo e a mensagem "*n não é um número primo*" é exibida. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 17 a ser solucionado.

17. Escreva um programa que imprime todos os números primos positivos a partir do zero até um número digitado.

Resolução do Exercício 17 – Codificação:

```
#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main() {

    setlocale(LC_ALL, "Portuguese");
```

```
int n, x, j, primo, sair, i=1;

while (i==1){
printf("Digite um número inteiro positivo: ");

scanf("%d", &n);

printf("Os números primos de 0 a %d são: ", n);

for (x = 2; x <= n; x++) {

    primo = 0;

    for (j = 2; j <= x/2; j++) {

        if (x % j == 0) {

            primo = 1;

            break;

        }

    }

    if (primo == 0) {

        printf("%d ", x);

    }

}

int aux = 1;

while (aux == 1){

    printf("\n\nDeseja executar o programa novamente? Digite 1 para sim e 0 para não:");

    scanf("%d", &sair);

    if (sair == 1) {

        aux = 0;

    }

    else if (sair == 0){

        aux = 0;

        i = 0;

    }

    else{
```

```

    printf("Valor inválido! Tente novamente.\n\n");

    aux = 1;

}

}

}

return 0;

}

```

Execução do Exercício 17 – Resolução

```

C:\Users\richa\OneDrive\Área de Trabalho\main.exe
Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite um número inteiro positivo: 100
Os números primos de 0 a 100 são: 2 3 5 7 11 13 17 19 23 29 31 37 41 43
47 53 59 61 67 71 73 79 83 89 97

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 1
Digite um número inteiro positivo: 500
Os números primos de 0 a 500 são: 2 3 5 7 11 13 17 19 23 29 31 37 41 43
47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241
251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353
359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461
463 467 479 487 491 499

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 2
Valor inválido! Tente novamente.

Deseja executar o programa novamente? Digite 1 para sim e 0 para não: 0
-----
Process exited after 13.7 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

Figura 17: tela da solução do exercício 17.

Descrição da aprendizagem obtida através da problemática:

Este programa recebe um número inteiro positivo do usuário e imprime todos os números primos de 0 até esse número. O programa começa definindo variáveis e o loop *while* principal. Dentro do loop *while* principal, o programa pede ao usuário para digitar um número inteiro positivo. Em seguida, o programa usa um loop *for* para iterar de 2 até o número fornecido pelo usuário. Dentro do loop *for*, o programa usa outro loop *for* para verificar se cada número é primo. O programa define a variável "*primo*" como 0

antes do loop *for* interno, o que significa que assume que o número atual é primo a princípio. Em seguida, o programa verifica se o número atual é divisível por qualquer número entre 2 e a metade desse número (porque um número não pode ser divisível por um número maior que sua metade). Se o número atual for divisível por qualquer número dentro desse intervalo, o programa define a variável "*primo*" como 1 e sai do loop *for* interno. Fora do loop *for* interno, o programa verifica se a variável "*primo*" foi definida como 0. Se sim, isso significa que o número atual é primo e o programa imprime esse número. Em seguida, pergunta ao usuário se deseja executar o programa novamente ou não.

Descrição do Exercício 18 a ser solucionado.

18. Escreva um programa que imprima o seguinte vetor:

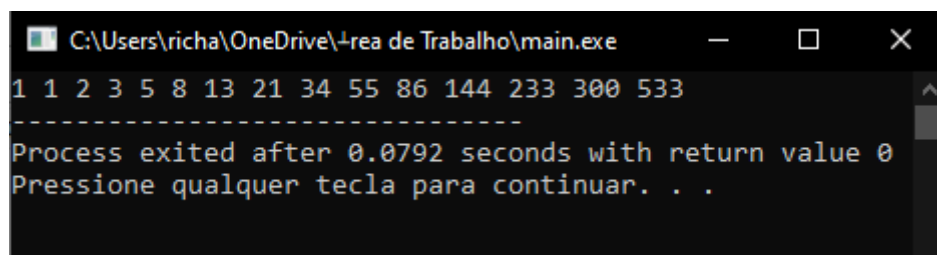
1 1 2 3 5 8 13 21 34 55 86 144 233 300 533

Resolução do Exercício 18 – Codificação:

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

int main() {
    int n[15] = {1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 86, 144, 233, 300, 533}, i;
    for (i=0; i<=14; i++){
        printf("%d ", n[i]);
    }
    return 0;
}
```

Execução do Exercício 18 – Resolução



```
C:\Users\richa\OneDrive\Trabalho\main.exe
1 1 2 3 5 8 13 21 34 55 86 144 233 300 533
-----
Process exited after 0.0792 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Figura 18: tela da solução do exercício 18.

Descrição da aprendizagem obtida através da problemática:

Este programa define um *array* de inteiros chamado "*n*" com 15 elementos e inicializa seus valores com uma série de Fibonacci pré-definida. Em seguida, ele imprime na tela todos os elementos do *array*, separados por um espaço. O resultado final será a impressão da série de Fibonacci definida no *array* "*n*". O uso de vetores é novamente apresentado nesse programa.

Descrição do Exercício 19 a ser solucionado.

19. Escreva um programa que imprima a seguinte matriz:

```
10 11 12 13 14
15 16 17 18 19
20 21 22 23 24
25 26 27 28 29
30 31 32 33 34
```

Resolução do Exercício 19 – Codificação:

```
#include <locale.h>
#include <stdio.h>
#include <stdlib.h>
#define tam 5

int matriz[tam][tam];

int main() {
    int i, j;

    matriz[0][0] = 10;
    matriz[0][1] = 11;
    matriz[0][2] = 12;
    matriz[0][3] = 13;
    matriz[0][4] = 14;

    matriz[1][0] = 15;
    matriz[1][1] = 16;
```

```
matriz[1][2] = 17;
matriz[1][3] = 18;
matriz[1][4] = 19;

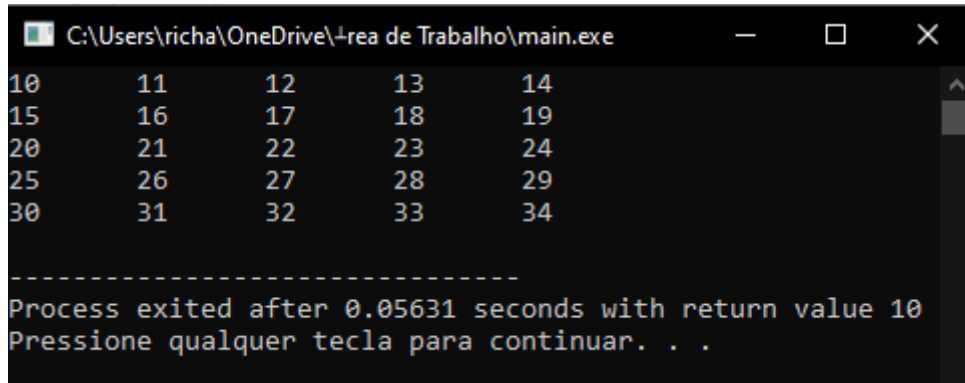
matriz[2][0] = 20;
matriz[2][1] = 21;
matriz[2][2] = 22;
matriz[2][3] = 23;
matriz[2][4] = 24;

matriz[3][0] = 25;
matriz[3][1] = 26;
matriz[3][2] = 27;
matriz[3][3] = 28;
matriz[3][4] = 29;

matriz[4][0] = 30;
matriz[4][1] = 31;
matriz[4][2] = 32;
matriz[4][3] = 33;
matriz[4][4] = 34;

for (i = 0; i < tam; i++) {
    for (j = 0; j < tam; j++) {
        printf("%d\t", matriz[i][j]);
    }
    printf("\n");
}
```


Execução do Exercício 19 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
10      11      12      13      14
15      16      17      18      19
20      21      22      23      24
25      26      27      28      29
30      31      32      33      34

-----
Process exited after 0.05631 seconds with return value 10
Pressione qualquer tecla para continuar. . .
```

Figura 19: tela da solução do exercício 19.

Descrição da aprendizagem obtida através da problemática:

Este programa cria e imprime uma matriz de 5x5 elementos, onde, primeiramente, é definida uma constante *tam* com valor 5, que será usada para determinar o tamanho da matriz. Em seguida, é declarada uma matriz bidimensional de tamanho *tam* * *tam* com o nome *matriz*. Na função *main()*, são atribuídos valores para cada elemento da matriz, usando a notação de índices *[linha][coluna]*. Esses valores são inseridos manualmente, e nesse caso foram preenchidos com números inteiros sequenciais, começando em 10 e terminando em 34. Finalmente, é utilizado um laço de repetição aninhado para imprimir cada elemento da matriz. O laço externo percorre as linhas da matriz, e o laço interno percorre as colunas. A função *printf()* é usada para imprimir cada elemento na tela, com um tabulador (*\t*) separando cada coluna, e uma quebra de linha (*\n*) separando cada linha.

Descrição do Exercício 20 a ser solucionado.

20. Escreva um programa que imprime os seguintes dados na tela

```
1-1 1-2 1-3 1-4 1-5
2-1 2-2 2-3 2-4 2-5
3-1 3-2 3-3 3-4 3-5
4-1 4-2 4-3 4-4 4-5
5-1 5-2 5-3 5-4 5-5
```

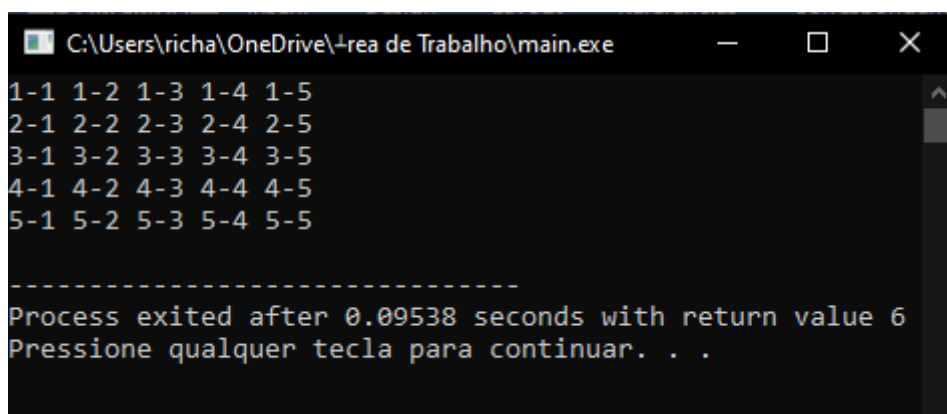
Resolução do Exercício 19 – Codificação:

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

```
int x, y;

int main() {
    for (x = 1; x <= 5; x++) {
        for (y = 1; y <= 5; y++) {
            printf("%d-%d ", x, y);
        }
        printf("\n");
    }
}
```

Execução do Exercício 20 – Resolução



```
C:\Users\richa\OneDrive\Área de Trabalho\main.exe
1-1 1-2 1-3 1-4 1-5
2-1 2-2 2-3 2-4 2-5
3-1 3-2 3-3 3-4 3-5
4-1 4-2 4-3 4-4 4-5
5-1 5-2 5-3 5-4 5-5

-----
Process exited after 0.09538 seconds with return value 6
Pressione qualquer tecla para continuar. . .
```

Figura 20: tela da solução do exercício 20.

Descrição da aprendizagem obtida através da problemática:

Neste programa, o código apresenta dois loops *for* aninhados que vão de 1 até 5, e em cada iteração, é impresso na tela os valores das variáveis *x* e *y*, separados por um hífen e seguidos por um espaço em branco. Após imprimir os valores de *y*, é executado um caractere de nova linha.

Referências

CASAVELLA, Eduardo. Vetores – arrays em linguagem C. 201-. Disponível em: <http://linguagemc.com.br/vetores-ou-arrays-em-linguagem-c/>. Acesso em: 21 mar. 2023.

CASAVELLA, Eduardo. O comando while em C. 201-. Disponível em: <http://linguagemc.com.br/o-comando-while-em-c/>. Acesso em: 21 mar. 2023.

CASAVELLA, Eduardo. Operadores Lógicos em C. 201-. Disponível em: <http://linguagemc.com.br/operadores-logicos-em-c/>. Acesso em: 21 mar. 2023.

GOOGLE. Série de Fibonacci. Disponível em: <https://sites.google.com/a/liesenberg.biz/cjogos/home/materiais-de-apoio/topicos-relativos-a-c/recursao/serie-de-fibonacci>. Acesso em: 19 mar. 2023.

LARA, Silvana Maria Affonso de. Recursão em C. Universidade de São Paulo – São Carlos; Instituto de Ciências Matemáticas e de Computação, pág 3 a 20. (e-book).

PINHO, Márcio Sarroglia. Matrizes e Vetores. 201-. Disponível em: <https://www.inf.pucrs.br/~pinho/LaproI/Vetores/Vetores.htm>. Acesso em: 29 mar. 2023.