

Lista de exercícios Estrutura de Repetição 1
Nome: Matheus Lourenço Soares
Curso: Desenvolvimento de Sistemas - Turma B
matheus.soares135@etec.sp.gov.br

Escola Técnica Estadual da cidade de São Paulo (ETEC) - R. Ulisses Cruz -
Belenzinho, São Paulo - SP - Brasil - CEP: 03077-000

Abstract. This work aims to answer the list of exercises in the Programming and Algorithms Technique subject, through knowledge obtained during classes, and our creativity and imagination when solving problems. With the help of the complementary material provided by professor Juliano Ratusznei, it was possible to understand what the exercise required.

Resumo. Este trabalho tem por objetivo responder a lista de exercícios da matéria de Técnica de programação e algoritmos, através de conhecimentos obtidos durante as aulas, e nossa criatividade e imaginação na hora de resolver problemas. Com o auxílio do material complementar disponibilizado pelo professor Juliano Ratusznei, foi possível entender o que o exercício pedia.

São Paulo - 2024

Descrição do Exercício 1 a ser solucionado.

Escreva um programa que imprima os números de 0 a 999.

Resolução do Exercício 1 - Codificação:

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

int main(void) {
    setlocale(LC_ALL, "Portuguese");

    //VARIÁVEIS
    int i = 0;

    //IMPRESSÃO DOS NÚMEROS ATÉ 999
    for (i = 0; i <= 999; i++){
        printf("%d\n", i);
    }
}
```

Execução do Exercício 1 - Resolução

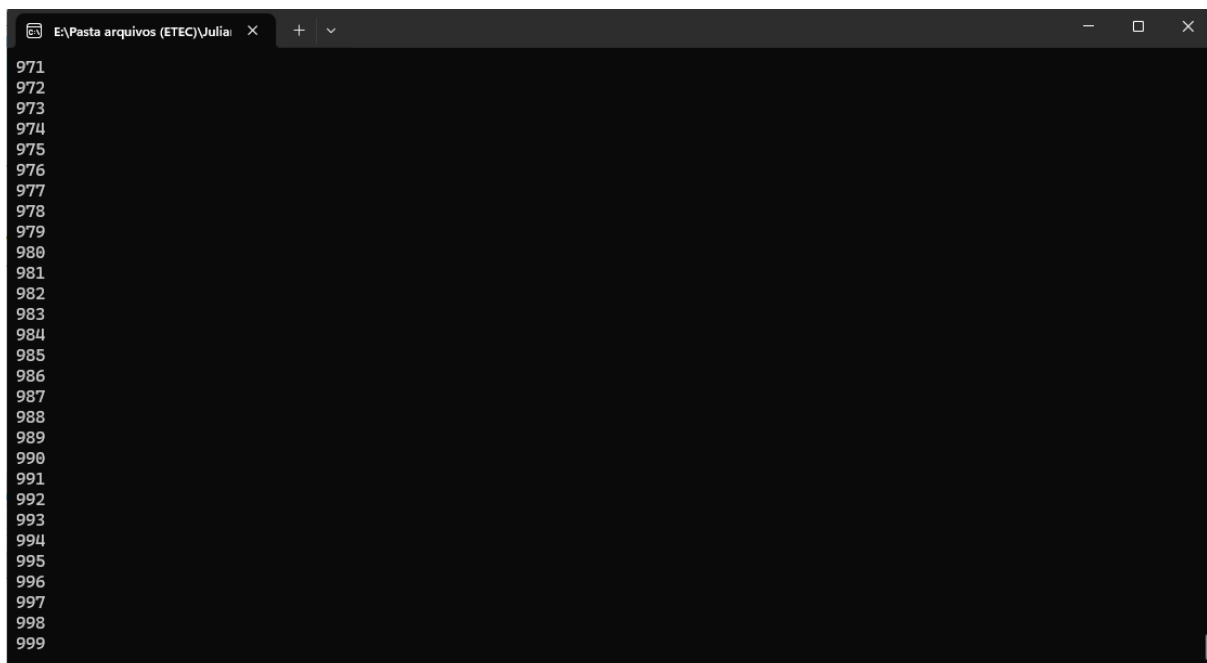
A screenshot of a Windows File Explorer window. The title bar shows the path "E:\Pasta arquivos (ETEC)\Julia:". The main area displays a list of files, each named with a four-digit number from 971 to 999. The files are listed in a single column, with each number on a new line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

Figura 1: tela da solução do exercício 1

Descrição do Exercício 2 a ser solucionado.

Escreva um programa que imprima na tela os números de 1 a 500 que são múltiplos de 5.

Resolução do Exercício 2 - Codificação:

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

void main(void) {

    //VARIÁVEIS
    int i = 5;

    //MÚLTIPLOS DE 5
    while (i <= 500) {
        printf("%d ", i);
        i = i + 5;
    }
}
```

Execução do Exercício 2 - Resolução

```
E:\Pasta arquivos (ETEC)\Julia \ x + v
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175
180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325
330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475
480 485 490 495 500
-----
Process exited after 0.03562 seconds with return value 4
Pressione qualquer tecla para continuar. . . |
```

Figura 2: tela de solução do exercício N

Descrição do Exercício 3 a ser solucionado.

Escreva um programa que imprima na tela os números de 1 a 500 que são múltiplos de 5 e de 3 (os dois ao mesmo tempo).

Resolução do Exercício 3 - Codificação:

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

void main(void) {

//VARIAVEIS
int i = 5, i2 = 3;

//MULTIPLICOS DE 5
while (i <= 500) {
    printf("%d ", i);
    i = i + 5;
```

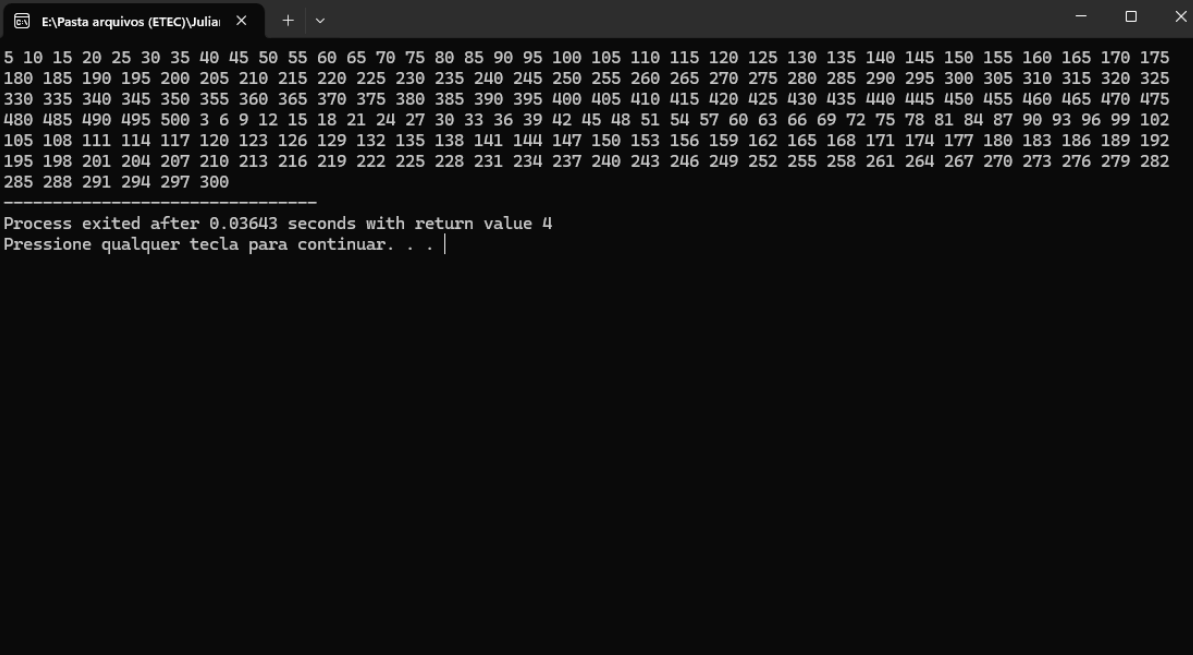
```

}

//MULTIPLICOS DE 3
while (i2 <= 300) {
    printf("%d ", i2);
    i2 = i2 + 3;
}
}

```

Execução do Exercício 3 - Resolução



```

E:\Pasta arquivos (ETEC)\Julia\ x + v
5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95 100 105 110 115 120 125 130 135 140 145 150 155 160 165 170 175
180 185 190 195 200 205 210 215 220 225 230 235 240 245 250 255 260 265 270 275 280 285 290 295 300 305 310 315 320 325
330 335 340 345 350 355 360 365 370 375 380 385 390 395 400 405 410 415 420 425 430 435 440 445 450 455 460 465 470 475
480 485 490 495 500 3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60 63 66 69 72 75 78 81 84 87 90 93 96 99 102
105 108 111 114 117 120 123 126 129 132 135 138 141 144 147 150 153 156 159 162 165 168 171 174 177 180 183 186 189 192
195 198 201 204 207 210 213 216 219 222 225 228 231 234 237 240 243 246 249 252 255 258 261 264 267 270 273 276 279 282
285 288 291 294 297 300
-----
Process exited after 0.03643 seconds with return value 4
Pressione qualquer tecla para continuar. . . |

```

Figura 3: tela de solução do exercício 3

Descrição do Exercício 4 a ser solucionado.

Escreva um programa que some os números de 1 a 100.

Resolução do Exercício 4 - Codificação:

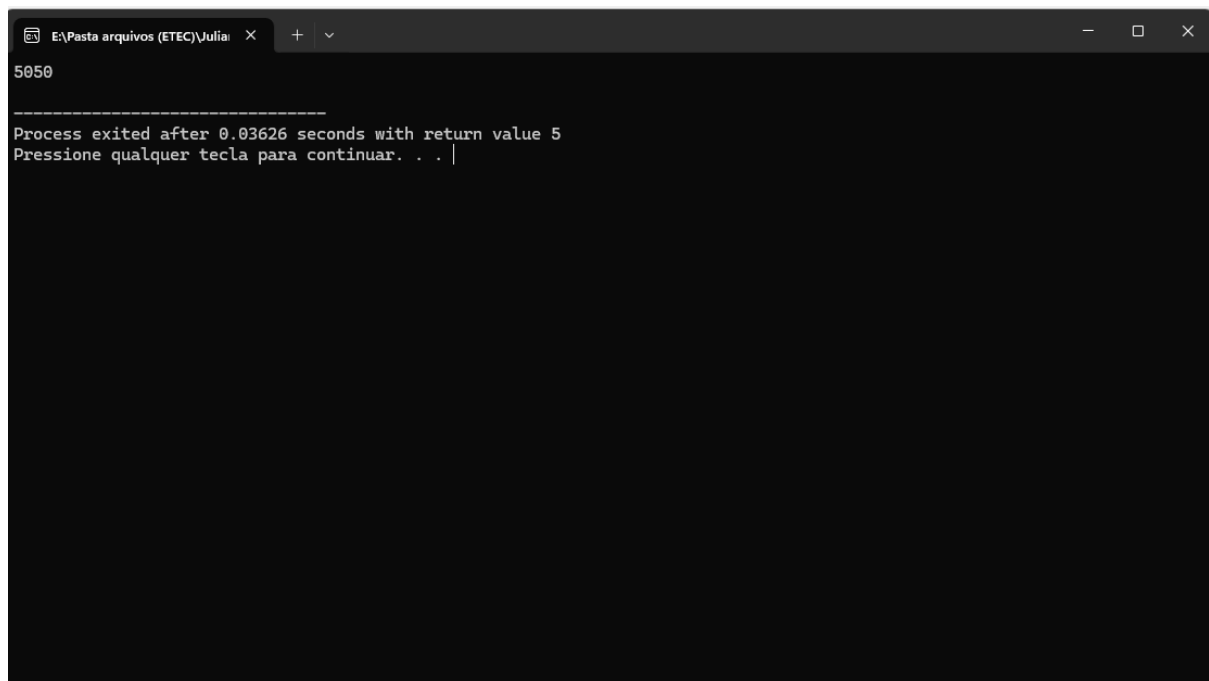
```

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

```

```
void main(void) {  
  
    //VARIÁVEIS  
    int i, i2 = 0 ;  
  
    //CÁLCULO  
    for (i = 0; i <= 100; i++){  
        i2 += i;  
    }  
  
    //IMPRESSÃO DO RESULTADO  
    printf("%d\n", i2);  
}
```

Execução do Exercício 4 - Resolução



```
E:\Pasta arquivos (ETEC)\Julia > 5050  
-----  
Process exited after 0.03626 seconds with return value 5  
Pressione qualquer tecla para continuar. . . |
```

Figura 4: tela de solução do exercício 4

Descrição do Exercício 5 a ser solucionado.

Escreva um programa que calcule a soma dos números pares entre 25 e 200.

Resolução do Exercício 5 - Codificação:

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

void main (void){
    setlocale(LC_ALL, "portuguese");

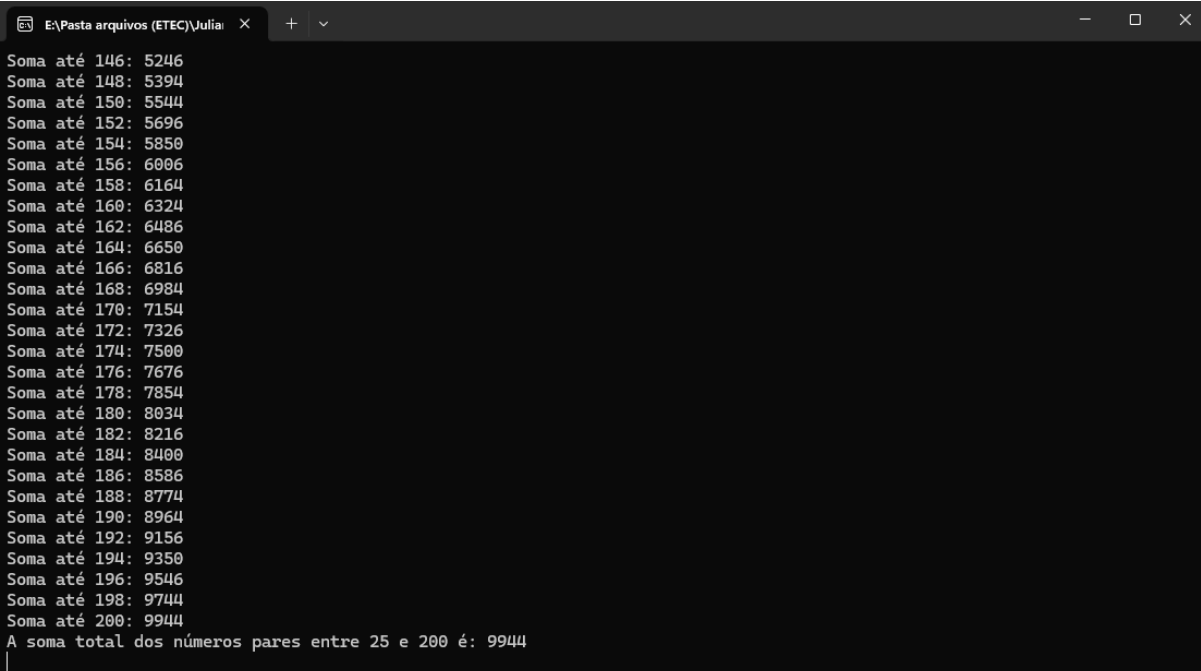
    int n = 0;
    int i;

    for (i = 26; i <= 200; i-=1){
        n += i;
    }

    printf("a soma dos números pares de 25 até 200 é: %d\n", n);

}
```

Execução do Exercício 5 - Resolução



```
E:\Pasta arquivos (ETEC)\Julia:
Soma até 146: 5246
Soma até 148: 5394
Soma até 150: 5544
Soma até 152: 5696
Soma até 154: 5850
Soma até 156: 6006
Soma até 158: 6164
Soma até 160: 6324
Soma até 162: 6486
Soma até 164: 6650
Soma até 166: 6816
Soma até 168: 6984
Soma até 170: 7154
Soma até 172: 7326
Soma até 174: 7500
Soma até 176: 7676
Soma até 178: 7854
Soma até 180: 8034
Soma até 182: 8216
Soma até 184: 8400
Soma até 186: 8586
Soma até 188: 8774
Soma até 190: 8964
Soma até 192: 9156
Soma até 194: 9350
Soma até 196: 9546
Soma até 198: 9744
Soma até 200: 9944
A soma total dos números pares entre 25 e 200 é: 9944
```

Figura 5: tela de solução do exercício 5

Descrição do Exercício 6 a ser solucionado.

Escreva um programa que imprime os primos até um número digitado.

Resolução do Exercício 6 - Codificação:

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

void main (void){
    setlocale(LC_ALL, "Portuguese");

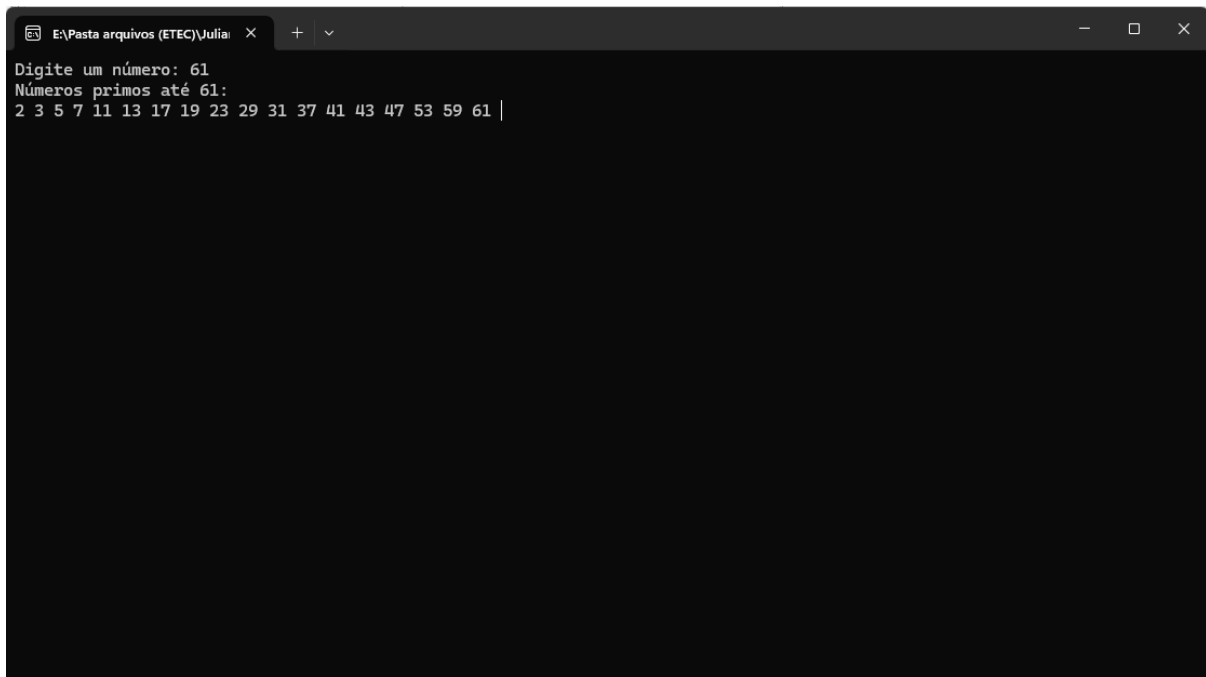
    int limite, i, primo;
    printf("Digite um número: ");
    scanf("%d", &limite);

    printf("Números primos até %d:\n", limite);
    for (i = 2; i <= limite; i++) {
        primo = 1;

        int p = 2;

        while (p < i && primo) {
            primo = (i % p);
            p++;
        }
        (primo) && printf("%d ", i);
    }
    getch();
}
```

Execução do Exercício 6 - Resolução



```
E:\Pasta arquivos (ETEC)\Julia > Digite um número: 61
Números primos até 61:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 |
```

Figura 6: tela de solução do exercício 6

Descrição do Exercício 7 a ser solucionado.

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

Resolução do Exercício 7 - Codificação

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

void main(void){
    setlocale (LC_ALL, "portuguese");

    int n, p, i;

    printf("Insira o número que deseja: ");
    scanf("%d", &n);

    p = n/n && n/1;

    if(n <= 1){
```

```

p = 0;
}

else{
for(i = 2; i <= n/2; i++){
if(n % i == 0){
p = 0;
}
}
}

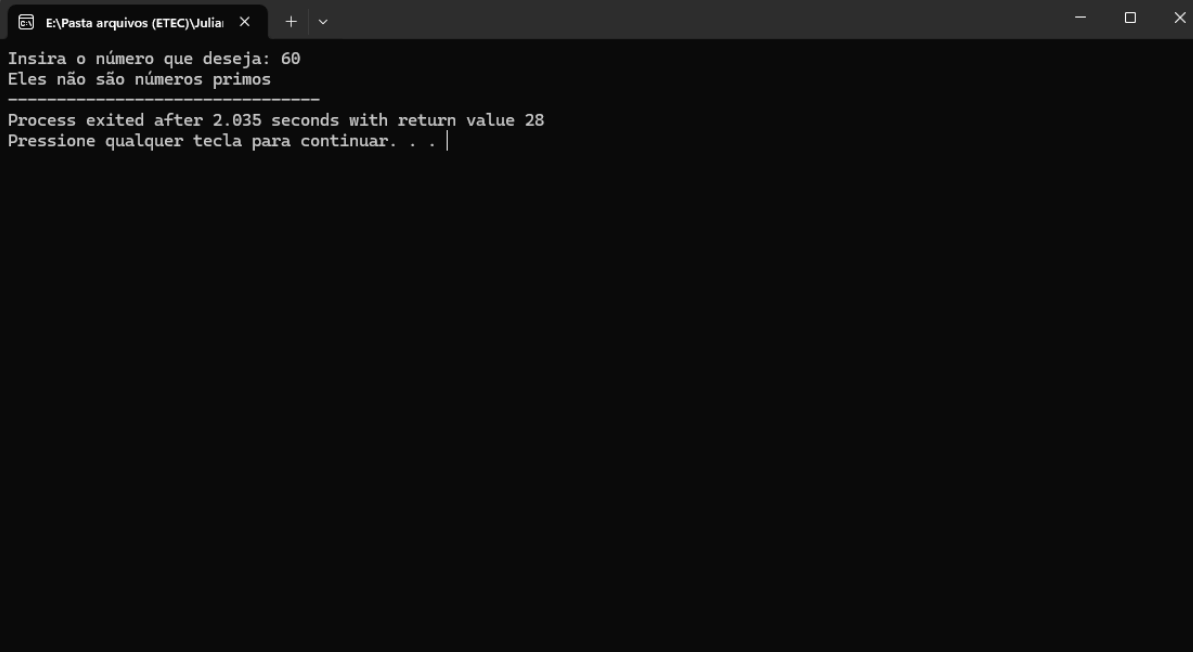
if(p == 1){
printf("Eles são números primos", n);
}

else{
printf("Eles não são números primos ", n);
}

}

```

Execução do Exercício 7 - Resolução

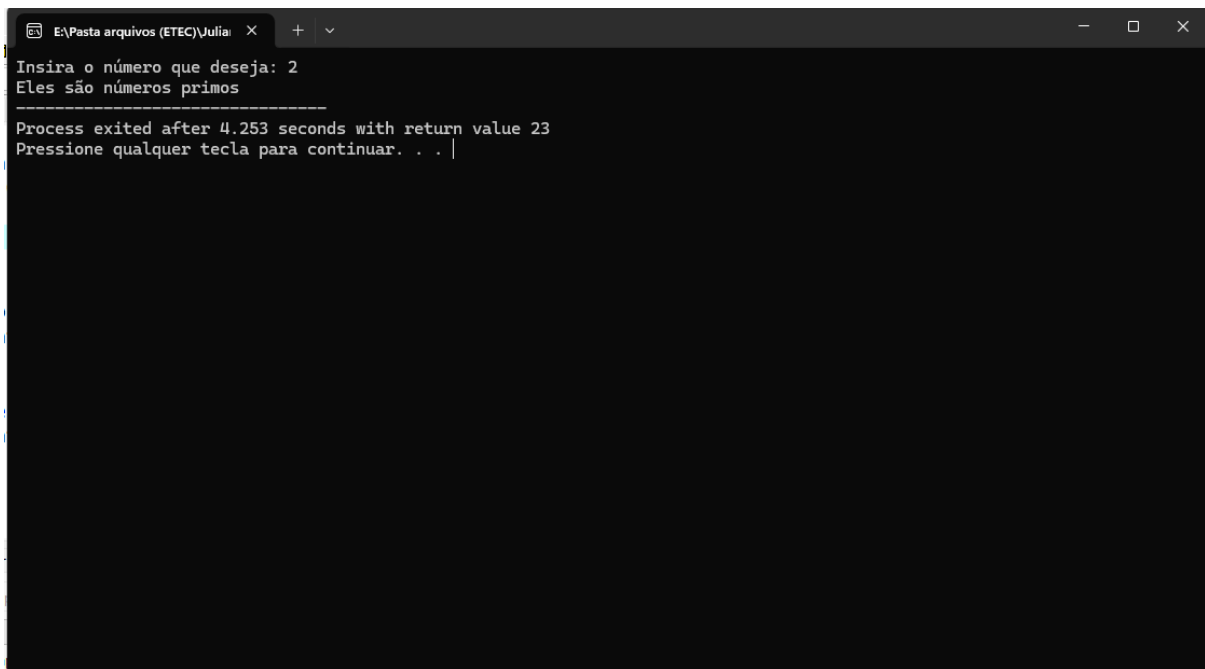


```

E:\Pasta arquivos (ETEC)\Julia
Insira o número que deseja: 60
Eles não são números primos
-----
Process exited after 2.035 seconds with return value 28
Pressione qualquer tecla para continuar. . . |

```

Figura 7: tela de solução do exercício 7



```
E:\Pasta arquivos (ETEC)\Julia: X + | v
Insira o número que deseja: 2
Eles são números primos
-----
Process exited after 4.253 seconds with return value 23
Pressione qualquer tecla para continuar. . . |
```

Figura 8: tela de solução do exercício 7

Descrição do Exercício 8 a ser solucionado.

Escreva um programa onde o usuário diz quantos números quer digitar, em seguida solicite a ele que digite todos os números e diga qual o maior número daqueles digitados.

Resolução do Exercício 8 - Codificação

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

void main(void) {
    setlocale(LC_ALL, "Portuguese");

    int n1, n2, i = 0, maior;

    printf("Insira o número de vezes que deseja digitar: ");
    scanf("%d", &n1);

    if (n1 > 0) {
```

```

printf("Digite o número: ");
scanf("%d", &n2);
maior = n2;

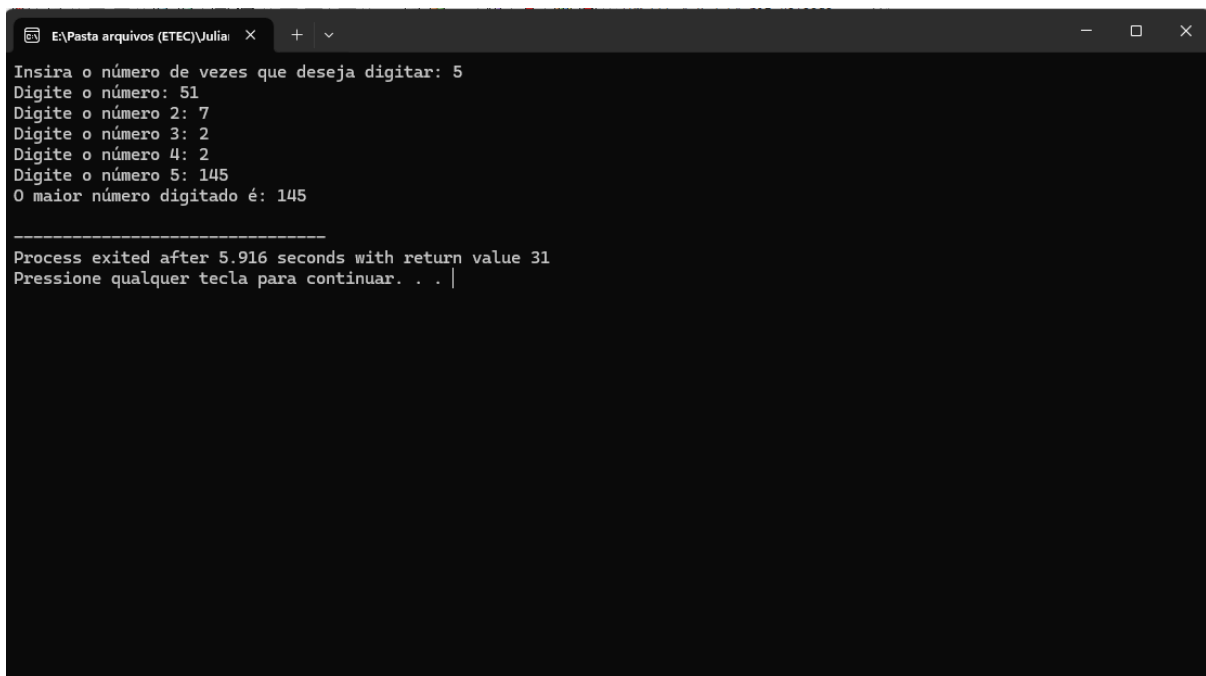
for (i = 1; i < n1; i++) {
    printf("Digite o número %d: ", i + 1);
    scanf("%d", &n2);

    if (n2 > maior) {
        maior = n2;
    }
}

printf("O maior número digitado é: %d\n", maior);
} else {
    printf("O número de vezes deve ser maior que 0.\n");
}
}

```

Execução do Exercício 8 - Resolução



```

E:\Pasta arquivos (ETEC)\Julia: x + v
Insira o número de vezes que deseja digitar: 5
Digite o número: 51
Digite o número 2: 7
Digite o número 3: 2
Digite o número 4: 2
Digite o número 5: 145
O maior número digitado é: 145

-----
Process exited after 5.916 seconds with return value 31
Pressione qualquer tecla para continuar. . . |

```

Figura 9: tela de solução do exercício 8

Descrição do Exercício 9 a ser solucionado.

Faça um programa que imprima a progressão aritmética de dois números. A entrada é a razão e o limite superior.

Ex: limite=20/ razão=5 -> 0 5 10 15 20

Resolução do Exercício 9 - Codificação

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

void main (void){
    setlocale(LC_ALL, "portuguese");
    int soma, n;
    {
        scanf("%d", &n) ;

        for(soma = 0; n > 0; n = n - 1)
        {
            soma = soma + n;
        }

        printf("\n%d", soma);
    }
}
```

Execução do Exercício 9 - Resolução

```
E:\Pasta arquivos (ETEC)\Julia: X + v
Insira o número de vezes que deseja digitar: 5
Digite o número: 51
Digite o número 2: 7
Digite o número 3: 2
Digite o número 4: 2
Digite o número 5: 145
O maior número digitado é: 145

-----
Process exited after 5.916 seconds with return value 31
Pressione qualquer tecla para continuar. . . |
```

Figura 10: tela de solução do exercício 9

Descrição do Exercício 10 a ser solucionado.

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 10- - Codificação

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

void main (void){
    setlocale(LC_ALL, "Portuguese");

    int numero, fatorial = 1, i;

    printf("Digite um número para calcular o fatorial: ");
    scanf("%d", &numero);

    printf("%d! = ", numero);
```

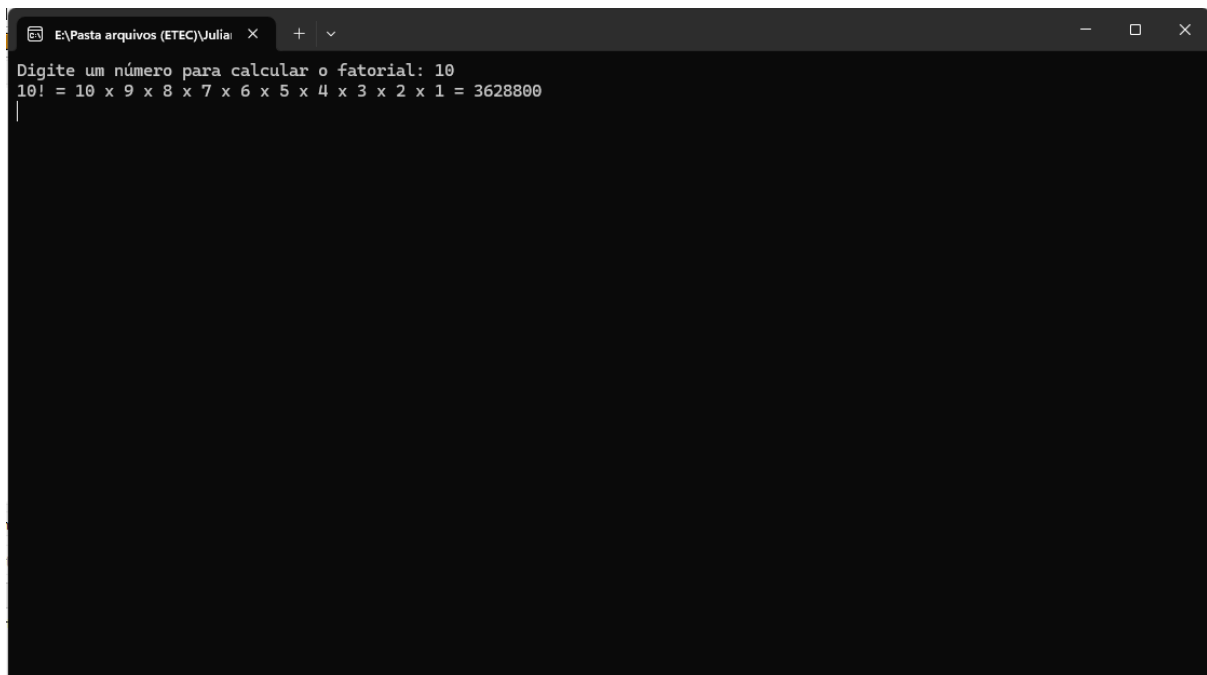
```

for (i = numero; i >= 1; i--) {
    printf("%d", i);

    if (i != 1) {
        printf(" x ");
    }
    fatorial *= i;
}
printf(" = %d\n", fatorial);
getch();
}

```

Execução do Exercício 10 - Resolução



```

E:\Pasta arquivos (ETEC)\Julia: X + v
Digite um número para calcular o fatorial: 10
10! = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 3628800

```

Figura 11: tela de solução do exercício 10

Descrição do Exercício 11 a ser solucionado.

Escreva um programa que imprime na tela a série de FIBONACCI até um número dado. Esta série começa com 1 e 1 e os próximos números são obtidos pela soma dos anteriores. Ex: 1 1 2 3 5 8 13 21 34 55

Resolução do Exercício 11 - Codificação

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

void main(void) {

    setlocale(LC_ALL, "Portuguese");

    int a, b, auxiliar, i, n;

    a = 0;
    b = 1;

    printf("Digite um número: ");
    scanf("%d", &n);
    printf("\nSérie de Fibonacci:\n\n");
    printf("%d\n", b);

    for(i = 0; i < n; i++) {

        auxiliar = a + b;
        a = b;
        b = auxiliar;

        printf("%d\n", auxiliar);
    }
}
```

Execução do Exercício 11 - Resolução


```
E:\Pasta arquivos (ETEC)\Julia X + v
Digite um número: 10
Série de Fibonacci:
1
1
2
3
5
8
13
21
34
55
89

-----
Process exited after 1.952 seconds with return value 10
Pressione qualquer tecla para continuar. . . |
```

Figura 12: tela de solução do exercício 11

Descrição do Exercício 12 a ser solucionado.

A série de RICCI difere da série de FIBONACCI porque os dois números iniciais são dados pelo usuário e os outros termos são gerados a partir da soma dos dois anteriores. Escreva um programa que imprime os 20 primeiros números a partir de duas entradas do usuário

Resolução do Exercício 11 - Codificação

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

void main(void) {

    setlocale(LC_ALL, "Portuguese");

    int n1, n2, i, soma;

    printf("Digite um número: ");
    scanf("%d", &n1);
    printf("Digite um número: ");
```

```

scanf("%d", &n2);
printf("\nSérie de RICCI:\n\n");
printf("%d\n", n2);

for(i = n1; i < 19; i++) {

    soma = n1 + n2;
    n1 = n2;
    n2 = soma;

    printf("%d\n", soma);
}
}

```

Execução do Exercício 12 - Resolução

```

E:\Pasta arquivos (ETEC)\Julia: x + v
Digite um número: 10
Digite um número: 15

Série de RICCI:

15
25
40
65
105
170
275
445
720
1165

-----
Process exited after 3.113 seconds with return value 5
Pressione qualquer tecla para continuar. . . |

```

Descrição da aprendizagem obtida através da problemática:

Com a realização deste exercício, compreendi a relevância das estruturas de repetição "for" e "while". Essas estruturas são fundamentais para simplificar e reduzir a repetição de trechos de código, tornando o desenvolvimento mais eficiente e organizado.