

My Project

AUTHOR
Versão 1.0

Sumário

Table of contents

Índice dos Arquivos

Lista de Arquivos

Esta é a lista de todos os arquivos e suas respectivas descrições:

C:/Users/Igor/Documents/testedoc/teste/brief.c	3
---	---

Arquivos

Referência do Arquivo

C:/Users/Igor/Documents/testedoc/teste/brief.c

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
#include <stdbool.h>
```

Funções

- void **verify** (const char *str)
Verifica e imprime se o caractere é uma letra ou um número.
- void **sequencia** (int valor)
Imprime os números em sequência.
- void **sequenciaInv** (int valor)
Função que imprime os números em sequência na ordem inversa.
- void **valoresAlternados** (int valor)
Função que imprime os números de forma alternada.
- bool **VerificaPrimo** (int num)
Função que verifica os números primos de 1 ao valor.
- void **primo** (int valor)
Imprime os números primos de 1 até o valor
- bool **VerificaPerfeito** (int num)
Função que verifica os números perfeitos.
- void **perfeito** (int valor)
Função que imprime os números perfeitos de 1 até o valor.
- int **main** (int argc, char *argv[])
Função principal.

Funções

int main (int argc, char * argv[])

Função principal.

Parâmetros

<i>argc</i>	quantidade de argumentos
<i>argv</i>	argumentos

Valores Retornados

<i>int</i>	retorna 0 se o programa rodar corretamente
------------	--

```

150                                     {
151     if (argc != 2){
152         printf("Número de argumentos inválidos!\nTente ./main <valor>: %s\n",
153             argv[0]);
154         return 1;
155     }
156     int valor = atoi(argv[1]);
157     if (valor <= 0){
158         printf("O valor deve ser maior que zero.\n");
159         return 1;
160     }
161     if (valor > 1000){
162         printf("Pra que tudo isso? Ta inventando arte.\n");
163     }
164
165     verify(argv[1]);
166     sequencia(valor);
167     sequenciaInv(valor);
168     valoresAlternados(valor);
169     primo(valor);
170     perfeito(valor);
171
172     return 0;
173 }

```

void perfeito (int valor)

Função que imprime os números perfeitos de 1 até o valor.

Parâmetros

<i>valor</i>	valores que serão impressos
--------------	-----------------------------

```

135                                     {
136     printf("Números perfeitos até %d:\n", valor);
137     for (int i = 1; i <= valor; i++) {
138         if (VerificaPerfeito(i)) {
139             printf("%d \n", i);
140         }
141     }
142 }

```

void primo (int valor)

Imprime os números primos de 1 até o valor

Parâmetros

<i>valor</i>	valores que serão impressos
--------------	-----------------------------

```

102                                     {
103     printf("Números primos até %d:\n", valor);
104     for (int i = 2; i <= valor; i++){
105         if (VerificaPrimo(i)){

```

```

106         printf("%d \n", i);
107     }
108 }
109 }

```

void sequencia (int valor)

Imprime os números em sequência.

Parâmetros

<i>valor</i>	valores que serão impressos em sequência
--------------	--

```

49     {
50     printf("Sequência até %d:\n", valor);
51     for (int i = 1; i <= valor; i++){
52         printf("%d \n", i);
53     }
54 }

```

void sequencialnv (int valor)

Função que imprime os números em sequência na ordem inversa.

Parâmetros

<i>valor</i>	valores que serão impressos em sequência na ordem inversa
--------------	---

```

60     {
61     printf("Sequência até %d na ordem inversa:\n", valor);
62     for (int i = valor; i >= 1; i--){
63         printf("%d \n", i);
64     }
65 }

```

void valoresAlternados (int valor)

Função que imprime os números de forma alternada.

Parâmetros

<i>valor</i>	valores que serão impressos de forma alternada
--------------	--

```

71     {
72     printf("Valores alternados até %d:\n", valor);
73     int inicio = 1;
74     int fim = valor;
75     while (inicio <= valor){
76         printf("%d ", inicio);
77         if (inicio != fim){
78             printf("%d \n", fim);
79         }
80         inicio++;
81         fim--;
82     }
83 }

```

bool VerificaPerfeito (int num)

Função que verifica os números perfeitos.

Parâmetros

<i>num</i>	valores que serão verificados se são perfeitos retval int: se o número for perfeito
------------	---

```
117     {
118     if (num <= 1) return false;
119     int soma = 1;
120     for (int i = 2; i * i <= num; i++) {
121         if (num % i == 0) {
122             soma += i;
123             if (i != num / i) {
124                 soma += num / i;
125             }
126         }
127     }
128     return soma == num;
129 }
```

bool VerificaPrimo (int num)

Função que verifica os números primos de 1 ao valor.

Parâmetros

<i>num</i>	valores que serão verificados se são números primos retval true: se o número for primo
------------	--

```
90     {
91     if (num <= 1) return false;
92     for (int i = 2; i * i <= num; i++) {
93         if (num % i == 0) return false;
94     }
95     return true;
96 }
```

void verify (const char * str)

Verifica e imprime se o caractere é uma letra ou um número.

Parâmetros

<i>str</i>	string a ser analisada
------------	------------------------

Valores Retornados

<i>void</i>	
-------------	--

```
12     {
13     bool num = true;
14     bool alfaNum = true;
15
16     // \0 é o chamado terminador nulo e determina o fim de uma string
17     if (str[0] == '\0') {
18         printf("O valor é nulo.\n");
19         return;
20     }
21
22     for (int i = 0; str[i] != '\0'; i++) {
23         //isdigit é uma função que verifica se o caractere é um dígito
24         if (!isdigit(str[i])) {
25             num = false;
26         }
27         //isalnum é uma função que verifica se o caractere é alfanumérico
28         if (!isalnum(str[i])) {
29             alfaNum = false;
30         }
31     }
```

```
32
33     if (num) {
34         printf("Número.\n");
35     }
36     else if (alfaNu) {
37         printf("Alfanumérico.\n");
38     }
39     else {
40         printf("NULO.\n");
41     }
42
43 }
```

Sumário

INDEX