Introdução

Este documento detalha o projeto "Operações Numéricas", um programa em C que oferece um menu interativo para realizar diversas operações matemáticas com números naturais. O objetivo é demonstrar a implementação de funções, passagem de parâmetros por referência e manipulação de entrada/saída de dados.

Subprogramas Implementados

Protótipos das Funções

```
int apresentarMenu(void);
void verificarPrimo(void);
int mdc(int a, int b);
void mdcUsuario(void);
void mmc(void);
void media10(void);
void eCapicua(void);
int inverterNumero(int num);
void inverterNumeroUsuario(void);
void verificarPrimosIntervalo(void);
int pedirNumeroNatural(char str[]);
int ePrimo(int n);
void trocar(int *a, int *b);
```

Descrição das Funções

Função	Descrição
`apresentarMenu()`	Exibe o menu principal de opções ao utilizador e retorna a opção escolhida.
`verificarPrimo()`	Solicita um número ao utilizador e informa se o mesmo é primo ou não, utilizando a função `ePrimo()`.
`mdc(int a, int b)`	Calcula e retorna o Máximo Divisor Comum (MDC) de dois números inteiros `a` e `b` utilizando o algoritmo de Euclides.

Função	Descrição
`mdcUsuario()`	Solicita dois números naturais ao utilizador, calcula o MDC usando `mdc()` e exibe o resultado.
`mmc()`	Solicita dois números naturais ao utilizador, calcula o Mínimo Múltiplo Comum (MMC) utilizando a relação com o MDC e exibe o resultado.
`media10()`	Solicita 10 números naturais ao utilizador, calcula a média e exibe o resultado.
`eCapicua()`	Solicita um número natural ao utilizador e verifica se o mesmo é uma capicua usando `inverterNumero()`.
`inverterNumero(int num)`	Inverte os algarismos de um número inteiro `num` e retorna o número invertido.
`inverterNumeroUsuario()`	Solicita um número natural ao utilizador, inverte-o usando `inverterNumero()` e exibe o resultado.
`verificarPrimosIntervalo()`	Solicita um intervalo (dois números naturais) ao utilizador e exibe todos os números primos contidos nesse intervalo, utilizando `ePrimo()`.
`pedirNumeroNatural(char str[])`	Solicita ao utilizador um número natural (inteiro não negativo) e valida a entrada, repetindo a solicitação caso o número seja inválido.
`ePrimo(int n)`	Verifica se um número inteiro `n` é primo, retornando 1 se for primo e 0 caso contrário.
`trocar(int *a, int *b)`	Troca os valores de duas variáveis inteiras `a` e `b` através da passagem de parâmetros por referência.

Passagem de Parâmetros por Referência

A passagem de parâmetros por referência foi utilizada na função `trocar(int *a, int *b)`. Esta função recebe os endereços de memória de duas variáveis inteiras. Ao manipular `*a` e `*b`, a função modifica diretamente os valores das variáveis originais na função chamadora (`mdc` neste caso), permitindo que a ordem dos números seja ajustada conforme necessário para o algoritmo do MDC.

Exemplos de Testes Realizados

A seguir, são apresentados alguns exemplos de testes com valores válidos e inválidos para as diferentes funcionalidades do programa.

Opção 1: Calcular o MDC

Entrada a	Entrada b	Saída Esperada	Saída Obtida
12	18	O máximo divisor comum de 12 e 18 é 6	O máximo divisor comum de 12 e 18 é 6
0	5	0 é inválido, deve ser um número natural (para 'a'); 5 é inválido, deve ser um número natural (para 'b'); O máximo divisor comum de 0 e 5 é 5 (se 0 for tratado como múltiplo)	0 é inválido, deve ser um número natural (para 'a'); ... (solicitação repetida)
7	11	O máximo divisor comum de 7 e 11 é 1	O máximo divisor comum de 7 e 11 é 1

Opção 2: Calcular o MMC

Entrada a	Entrada b	Saída Esperada	Saída Obtida
4	6	O mínimo múltiplo comum de 4 e 6 é 12	O mínimo múltiplo comum de 4 e 6 é 12
0	10	0 é inválido, deve ser um número natural (para 'a'); ... (solicitação repetida)	0 é inválido, deve ser um número natural (para 'a'); ... (solicitação repetida)
3	5	O mínimo múltiplo comum de 3 e 5 é 15	O mínimo múltiplo comum de 3 e 5 é 15

Opção 3: Calcular a média de 10 números

Entradas	Saída Esperada	Saída Obtida
1, 2, 3, 4, 5, 6, 7, 8, 9, 10	A média dos 10 números é: 5	A média dos 10 números é: 5
-1 (primeira entrada)	-1 é inválido, deve ser um número natural (solicitação repetida)	-1 é inválido, deve ser um número natural (solicitação repetida)

Opção 4: Inverter os algarismos de um número inteiro

Entrada	Saída Esperada	Saída Obtida
12345	O número invertido é: 54321	O número invertido é: 54321
100	O número invertido é: 1	O número invertido é: 1
-5	-5 é inválido, deve ser um número natural (solicitação repetida)	-5 é inválido, deve ser um número natural (solicitação repetida)

Opção 5: Verificar se um número inteiro é uma capicua

Entrada	Saída Esperada	Saída Obtida
121	121 é capicua	121 é capicua
123	123 não é capicua	123 não é capicua
22	22 é capicua	22 é capicua

Opção 6: Verificar se um número inteiro é primo

Entrada	Saída Esperada	Saída Obtida
7	7 é primo	7 é primo
10	10 Não é primo	10 Não é primo
1	1 é primo (considerando 1 como primo para este contexto)	1 é primo

Entrada	Saída Esperada	Saída Obtida
0	0 é inválido, deve ser um número natural (solicitação repetida)	0 é inválido, deve ser um número natural (solicitação repetida)

Opção 7: Mostrar todos os números primos num intervalo

Entrada a	Entrada b	Saída Esperada	Saída Obtida
1	10	Os números primos entre 1 e 10 são: 1 2 3 5 7	Os números primos entre 1 e 10 são: 1 2 3 5 7
20	30	Os números primos entre 20 e 30 são: 23 29	Os números primos entre 20 e 30 são: 23 29
10	12	Os números primos entre 10 e 12 são: Não há	Os números primos entre 10 e 12 são: Não há

Conclusão

O programa "Operações Numéricas" demonstra a aplicação de conceitos fundamentais de programação em C, incluindo a estruturação de código com funções, tratamento de entrada do utilizador com validação, e a passagem de parâmetros por referência para otimizar certas operações. A modularização do código em subprogramas facilita a manutenção e a reutilização.