

Números Primos

Abrantes Araújo Silva Filho

2023-08-19

Descrição:

Números **primos** são definidos como números inteiros maiores do que 1 cujos únicos fatores são o 1 e o próprio número. Assim, 3 é um número primo pois seus únicos fatores são 1 e o próprio 3. O número 4, por outro lado, é um número **composto** (não é primo), pois é o produto de 2×2 . Neste exercício você escreverá um algoritmo para gerar e imprimir todos os números primos em uma faixa numérica especificada pelo usuário.

Dicas:

A maneira mais fácil para verificar se um número qualquer n é primo, é dividir esse número n por todos os números menores do que ele, começando em 2 e indo até $n - 1$. Se algumas dessas divisões não tiver resto (ou seja, tiver resto zero), o número não é primo.

Por exemplo: para saber se 5 é primo, basta verificar os restos das divisões: $5/2$, $5/3$, $5/4$. Como todas essas divisões tem resto diferente de zero, então o número 5 é primo.

Algumas observações importantes: a) o único número primo par é 2; b) por definição, o número 1 não é primo; e c) a função **módulo** é útil na identificação de números primos, pois ela retorna o resto da divisão entre dois números inteiros (descubra como usar a função módulo na linguagem C).

Detalhes de implementação:

Neste exercício você **deverá utilizar o arquivo fonte “`primos.c`”** fornecido para você. Esse arquivo já tem o código para solicitar a faixa numérica do usuário, dentro de valores válidos (estude muito bem esses códigos para entender e aprender).

Você precisa acrescentar o código para imprimir todos os números primos que estiverem dentro da faixa especificada pelo usuário (coloque seu código logo abaixo da parte que se inicia com `// TODO` :).

Seu código precisa iterar através da faixa de valores especificada pelo usuário, incluindo os números que ele informou, e testar se esses números são primos ou não. Se forem primos, devem ser impressos no terminal.

Input:

Ao ser iniciado seu programa exibirá o prompt “**Menor número:** ”: o usuário deverá informar um número inteiro maior ou igual a 1. Depois disso seu programa exibirá o prompt “**Maior número:** ”: o usuário deverá informar um número inteiro maior do que o menor número já informado.

Considere que o usuário informará somente números inteiros n válidos, na faixa $1 \leq n \leq 100$. Além disso o usuário informará corretamente o maior número, ou seja, será sempre maior do que o menor número informado.

Output:

Após receber os inputs do usuário seu programa deverá imprimir todos os números primos existentes na faixa numérica informada pelo usuário, um número por linha.

Restrições:

Considere que o usuário sempre informará o menor número (min) e o maior número (max) na faixa: $1 \leq min < max \leq 100$.

Entrega:

Utilize o *starter file* “`primos.c`” (mantenha esse padrão de nome), preencha as informações de identificação e envie o arquivo no Autolab, no exercício denominado “Números Primos”.

Limites de execução:

Limite de memória: 512 MiB; limite de tempo: 1 segundo.

Exemplo:

O print abaixo mostra a execução do programa em um terminal de comandos (linha 1), o menor número escolhido pelo usuário (linha 2), o maior número escolhido pelo usuário (linha 3), a lista de primos (linhas 4–8), e a posição final correta do *prompt* de comando após o término do programa (linha 9).

```
01 ./primos
02 Menor número: 3
03 Maior número: 13
04 3
05 5
06 7
07 11
08 13
09 $
```

Aqui está outro exemplo de execução do programa:

```
01 ./primos
02 Menor número: 50
03 Maior número: 80
04 53
05 59
06 61
07 67
08 71
09 73
10 79
11 $
```