

Desafio (Crivo de Eratóstenes)

Vimos em um exercício anterior uma estratégia para descobrir se um número é primo com base em um loop onde procuramos por algum divisor para o número. Uma estratégia completamente diferente é o Crivo de Eratóstenes:

http://en.wikipedia.org/wiki/Sieve_of_Eratosthenes

Esta estratégia funciona como se os números fossem passados por uma série de “peneiras”. Ela consiste em montar uma tabela contendo vários números, de 2 a `N_MAX`, onde `N_MAX` é o maior número que pode ser testado pela tabela. Em C, esta tabela pode ser criada alocando-se um vetor de `N_MAX` posições.

Em resumo, o processo consiste no seguinte. No início, todas as posições da tabela recebem o valor “verdadeiro” (por exemplo, 1). Tomamos então o índice 2, e atribuímos a todas as posições com índices **múltiplos de 2** no vetor o valor “falso” (representador por 0, por exemplo). Vamos então percorrendo o vetor, e cada vez que encontramos uma posição com valor “verdadeiro”, repetimos o processo de transformar as posições com índices múltiplos daquela posição o valor “falso”. Veja a animação no link da Wikipedia.

Ao final, teremos uma tabela que diz facilmente se qualquer número menor que `N_MAX` é primo. Construir a tabela leva tempo e consome memória, mas o teste para verificar se um número `x` é primo é simplesmente: `tabela[x]`. Isso quer dizer que, por um lado, não vale a pena usar o Crivo de Eratóstenes apenas para saber se alguns poucos números são primos. Por outro lado, se tivermos que fazer a verificação para muitos números, essa estratégia pode se mostrar mais eficiente.

Tarefa

Você está recebendo um programa que implementa a lógica descrita acima. Sua tarefa consiste em:

1. testar o programa dado para alguns valores de `num`;
2. alterar `N_MAX` para 5000000. O programa roda? Depois que entregar esta tarefa, procure entender o porquê!
3. alterar o programa de tal forma que:
 - (a) a criação da tabela fique em uma função, a qual recebe como parâmetro de entrada o tamanho da tabela (o qual deve ser escolhido pelo usuário) e retorna a referência para um vetor alocado dinamicamente que armazena em cada posição se o valor associado é primo ou não. Protótipo:

```
int* crivoDeEratosthenes (int tamanho);
```

Após a alteração, espera-se que a função main() seja:

```
1  int main()
2  {
3      int valorMax, num,
4          *crivo;
5
6      printf("Deseja verificar at qual valor m ximo? ");
7      scanf("%d", &valorMax);
8
9      crivo = crivoDeEratostenes(valorMax);
10
11     printf("\n\nPesquisa por primos entre 1 e %d...\n", valorMax);
12     printf("Digite -1 para sair.\n");
13
14     scanf("%d", &num);
15     while(num != -1)
16     {
17         if(crivo[num])
18             printf("Eh primo!\n");
19         else
20             printf("Nao eh primo!\n");
21
22         scanf("%d", &num);
23     }
24
25     free(crivo);
26     return 0;
27
28 }
```