

Vetores

Algoritmos e Programação de Computadores – ABI/LFI/TAI



**INSTITUTO
FEDERAL**
Brasília

Prof. Daniel Saad Nogueira
Nunes

IFB – Instituto Federal de Brasília,
Campus Taguatinga



Sumário

- 1 Introdução
- 2 Vetores
- 3 Exemplos



Sumário

1 Introdução



Introdução

- Imagine o seguinte problema: dados 100 valores inteiros lidos do teclado, verificar se todos são distintos.
- Como resolvê-lo?
- Poderíamos declarar **cem** variáveis inteiras e realizar uma comparação par a par.



Introdução

```
1  #include <stdio.h>
2
3  int main(void){
4      int numero_1, numero_2, ..., numero_100;
5      printf("Digite o primeiro valor: ");
6      scanf("%d",&numero_1);
7      ...
8      printf("Digite o centésimo valor: ");
9      scanf("%d",&numero_100);
10     // realiza comparações
11     ...
12     return 0;
13 }
```



Introdução

- Esta solução, apesar de funcionar, é extremamente massante e nada elegante.
- Felizmente, na linguagem C podemos declarar uma coleção de variáveis de um mesmo tipo.
- Mecanismo: **vetores!**



Sumário

2 Vetores



Vetores

- Os vetores fornecem uma maneira de declarar uma coleção de variáveis do mesmo tipo sob o mesmo identificador.
- Através do identificador, podemos acessar qualquer variável de uma maneira bem simples.



Sumário

- 2 Vetores
 - Sintaxe
 - Organização
 - Utilização
 - Cuidados
 - Inicialização
 - Funções e vetores



Vetores: Sintaxe

- Para declarar um vetor de tamanho `tam` de um determinado `tipo`, utilizamos a seguinte sintaxe: `tipo nome_vetor[tam];`
- Por exemplo, declaração de um vetor de inteiros, chamado `idades`, de tamanho 100: `int idades[100];`
- Declaração de um vetor de números ponto-flutuante, de precisão dupla, chamado `salarios`, de tamanho 50: `double salarios[50];`



Vetores: Sintaxe

- Para acessar o i -ésimo valor do vetor, utilizamos o operador `[i]`.
- O primeiro valor do vetor está na posição 0.
- O último valor do vetor, está na posição $n - 1$, em que n é o tamanho do vetor.



Vetores: Sintaxe

- Para acessar o terceiro valor do vetor `salarios`: `salarios[2];`
- Para acessar o último valor do vetor `idades`: `idades[99];`
- Para acessar o primeiro valor do vetor `idades`: `idades[0];`



Sumário

2 Vetores

- Sintaxe
- **Organização**
- Utilização
- Cuidados
- Inicialização
- Funções e vetores



Vetores: Organização

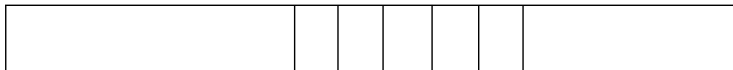
- Ao declarar um vetor de um determinado tamanho n , temos a presença de n espaços consecutivos de memória.
- Exemplo para um vetor de tamanho 5:

```
int vetor[5];
```

vetor

0 1 2 3 4

Memória





Vetores: Organização

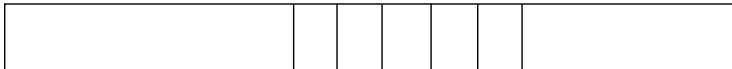
- Ao declarar um vetor de um determinado tipo T de tamanho n , temos a presença de n espaços consecutivos de memória, cada um com tamanho igual ao exigido pelo tipo T.
- Exemplo para um vetor de inteiros de tamanho 5:

```
int vetor[5];
```

```
vetor
```

1000 1004 1008 1012 1016

Memória





Sumário

2 Vetores

- Sintaxe
- Organização
- **Utilização**
- Cuidados
- Inicialização
- Funções e vetores



Vetores: Utilização

- Prosseguindo com o nosso problema inicial, vamos construir um programa que leia 100 valores inteiros e verifique se todos os valores são distintos.



Vetores: Utilização

Estratégia

- Ler os cem valores em um vetor de inteiros.
- Realizar comparações par a par entre os elementos e através de uma variável indicadora, verificar:
 - ▶ Se existe algum par de elementos iguais, então a resposta é falsa.
- Após todas as comparações, se foi verificado que não há par de elementos iguais, então a resposta é verdadeira.



Vetores: Utilização

Declaração

```
4   int i, j;  
5   int numeros[100];  
6   int resposta = 1;
```



Vetores: Utilização

Leitura

```
7  for (i = 0; i < 100; i++) {  
8      printf("Digite o elemento %d: ", i);  
9      scanf("%d", &numeros[i]);  
10 }
```



Vetores: Utilização

Comparação

```
11  for (i = 0; i < 100 && resposta; i++) {  
12      for (j = i + 1; j < 100 && resposta; j++) {  
13          if (numeros[i] == numeros[j])  
14              resposta = 0;  
15      }  
16  }
```



Vetores: Utilização

Checagem

```
if (resposta) {  
    printf("Todos os elementos são distintos.\n");  
}  
else {  
    printf("Existem elementos repetidos.\n");  
}
```



Vetores: Utilização

Programa Completo

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i, j;
5      int numeros[100];
6      int resposta = 1;
7      for (i = 0; i < 100; i++) {
8          printf("Digite o elemento %d: ", i);
9          scanf("%d", &numeros[i]);
10     }
11     for (i = 0; i < 100 && resposta; i++) {
12         for (j = i + 1; j < 100 && resposta; j++) {
13             if (numeros[i] == numeros[j])
14                 resposta = 0;
15         }
16     }
```



Vetores: Utilização

Programa Completo

```
17  if (resposta) {
18      printf("Todos os elementos são distintos.\n");
19  }
20  else {
21      printf("Existem elementos repetidos.\n");
22  }
23  return 0;
24 }
```




Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- **Cuidados**
- Inicialização
- Funções e vetores



Vetores: Cuidados

Out of Bounds

- Um vetor na C tem um tamanho fixo.
- É um erro de lógica tentar acessar posições que não pertencem ao vetor.
- As posições válidas de um vetor de tamanho n estão no intervalo $[0, n - 1]$.
- O acesso a alguma posição inválida, como -1 ou n , por exemplo, pode ocasionar uma **falha de segmentação**.



Vetores: Cuidados

Out of Bounds

- O compilador não verifica isso para você.
- É total responsabilidade do programador controlar o acesso aos elementos de um vetor.



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=1;i<=5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=1;i<=5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=1;i<=5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=1;i<=5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Programa Corrigido

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=0;i<5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=0;i<5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Declaração

- Como os vetores tem tamanho fixo, é importante escolher o seu tamanho adequadamente.
- Não se pode declarar um tamanho menor do que o necessário.



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void) {
4      int vetor[5];
5      int n;
6      int i;
7      printf("Digite o número de valores que deseja ler: ");
8      scanf("%d", &n);
9      for (i = 0; i < n; i++) {
10         scanf("%d", &vetor[i]);
11     }
12     printf("Os elementos digitados foram: ");
13     for (i = 0; i < n; i++) {
14         printf("%d ", vetor[i]);
15     }
16     printf("\n");
17     return 0;
18 }
```




Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void) {
4      int vetor[5];
5      int n;
6      int i;
7      printf("Digite o número de valores que deseja ler: ");
8      scanf("%d", &n);
9      for (i = 0; i < n; i++) {
10         scanf("%d", &vetor[i]);
11     }
12     printf("Os elementos digitados foram: ");
13     for (i = 0; i < n; i++) {
14         printf("%d ", vetor[i]);
15     }
16     printf("\n");
17     return 0;
18 }
```



Vetores: Cuidados

Programa Corrigido

```
1  #include <stdio.h>
2  int main(void) {
3      int vetor[5];
4      int n;
5      int i;
6      printf("Digite o número de valores que deseja ler: ");
7      scanf("%d", &n);
8      if (n > 5) {
9          printf("Número de valores alterado para 5.\n");
10         n = 5;
11     }
12     for (i = 0; i < n; i++) {
13         scanf("%d", &vetor[i]);
14     }
15     printf("Os elementos digitados foram: ");
16     for (i = 0; i < n; i++) {
17         printf("%d ", vetor[i]);
18     }
19     printf("\n");
20     return 0;
21 }
```



Vetores: Cuidados

Declaração

- Como os vetores tem tamanho fixo, é importante escolher o seu tamanho adequadamente.
- Não se pode declarar um tamanho menor do que o necessário.



Vetores: Cuidados

- Suponha as seguintes declarações:

```
1  int n;  
2  int vetor[5];  
3  int x;
```

- Imagine que as variáveis estejam dispostas da seguinte forma na memória:

```
int vetor[5];
```

```
      vetor  
      n 0 1 2 3 4 x
```

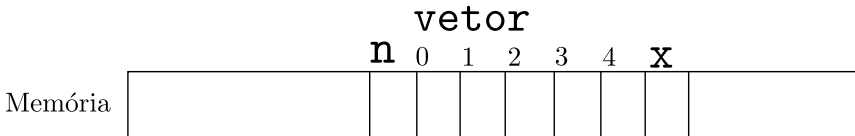
Memória





Vetores: Cuidados

```
int vetor[5];
```



- O que acontece se fizermos `vetor[5] = -2;`
- O que acontece se fizermos `vetor[-1] = 0;`
- Estaremos manipulando, indevidamente, as variáveis `n` e `x`.
- Mesmo que o seu programa não aborte, ele terá um *bug* difícil de detectar por conta do mal uso do vetor.



Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- Cuidados
- **Inicialização**
- Funções e vetores



Inicialização

- Em alguns casos pode-se desejar criar um vetor já inicializado com alguns elementos.
- Isso pode ser feito de maneira simples em C:

```
int vetor[] = {1,2,3,4,5};
```
- O tamanho do vetor é inferido automaticamente e é opcional colocá-lo neste tipo de declaração.



Inicialização

```
1  #include <stdio.h>
2
3  int main(void){
4      int i;
5      int vetor_int[] = {1,2,3,4,5};
6      float vetor_float[] = {0.5,2.3,1.1,2.4,3.9,-1.4};
7      for(i=0;i<5;i++){
8          printf("%d ",vetor_int[i]);
9      }
10     printf("\n");
11     for(i=0;i<6;i++){
12         printf("%.1f ",vetor_float[i]);
13     }
14     printf("\n");
15     return 0;
16 }
```




Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- Cuidados
- Inicialização
- Funções e vetores



Funções: Vetores

- Diferentemente de variáveis simples, os vetores, ao serem passados como parâmetros de função, ao serem modificados pela função, persistem as modificações.
- Isto é: não é criada uma cópia do vetor!



Funções: Vetores

```
1  #include <stdio.h>
2
3  void zera_vetor(int vetor[5]){
4      int i;
5      for(i=0;i<5;i++){
6          vetor[i] = 0;
7      }
8  }
9
10 void imprime_vetor(int vetor[5]){
11     int i;
12     for(i=0;i<5;i++){
13         printf("%d ",vetor[i]);
14     }
15     printf("\n");
16 }
```



Funções: Vetores

```
18 int main(void){
19     int v[5] = {1,2,3,4,5};
20     imprime_vetor(v);
21     zera_vetor(v);
22     imprime_vetor(v);
23     return 0;
24 }
```

O que será impresso?



Funções: Vetores

- Também não é possível retornar vetores através de funções.



Funções: Vetores

```
1  #include <stdio.h>
2
3  int[] le_vetor() {
4      int i;
5      int v[100];
6      for (i = 0; i < 100; i++) {
7          scanf("%d", &v[i]);
8      }
9      return v;
10 }
11
12 void imprime_vetor(int v[100]){
13     int i;
14     for(i=0;i<100;i++){
15         printf("%d",v[i])
16     }
17 }
18
19 int main(void) {
20     int vetor[100] = le_vetor();
21     return 0;
22 }
```



Funções: Vetores

Retorno de Vetores

- Na verdade conseguimos retornar vetores, mas de outra forma: ponteiros + alocação dinâmica de memória.
- Isso ficará para outra aula.



Funções: Vetores

- Como os vetores são alterados, podemos modificar o exemplo anterior para o seguinte.



Funções: Vetores

```
1  #include <stdio.h>
2  void le_vetor(int v[100]) {
3      int i;
4      for (i = 0; i < 100; i++) {
5          scanf("%d", &v[i]);
6      }
7  }
8
9  void imprime_vetor(int v[100]) {
10     int i;
11     for (i = 0; i < 100; i++) {
12         printf("%d", v[i]);
13     }
14     printf("\n");
15 }
16
```



Funções: Vetores

```
17 int main(void) {  
18     int vetor[100];  
19     le_vetor(vetor);  
20     imprime_vetor(vetor);  
21     return 0;  
22 }
```



Funções: Vetores

- Durante a definição de uma função que receba o vetor, podemos omitir o tamanho do vetor entre os colchetes.
- Para que a função saiba qual o tamanho do vetor, podemos passar uma variável inteira extra.
- Ex:

```
1 void exemplo_funcao(int vetor[], int n){  
2     ...  
3 }
```



Funções: Vetores

- Mas qual a vantagem desta forma de declaração? Afinal, precisamos de dois parâmetros agora. . .
- A vantagem é que, com esta forma, é possível criar funções que funcionem com qualquer tamanho de vetor, pois o parâmetro do tamanho do vetor está desacoplado do mesmo.
- Vamos modificar as funções `le_vetor` e `imprime_vetor` para este fim.



Funções: Vetores

```
1 void le_vetor(int v[], int n) {
2     int i;
3     for (i = 0; i < n; i++) {
4         scanf("%d", &v[i]);
5     }
6 }
7
8 void imprime_vetor(int v[], int n) {
9     int i;
10    for (i = 0; i < n; i++) {
11        printf("%d ", v[i]);
12    }
13    printf("\n");
14 }
```



Funções: Vetores

- Com isso, podemos escrever programas que leiam e imprimam vetores de tamanhos diferentes reutilizando as funções.



Funções: Vetores

```
1  #include <stdio.h>
2
3  void le_vetor(int v[], int n) {
4      int i;
5      for (i = 0; i < n; i++) {
6          scanf("%d", &v[i]);
7      }
8  }
9
10 void imprime_vetor(int v[], int n) {
11     int i;
12     for (i = 0; i < n; i++) {
13         printf("%d ", v[i]);
14     }
15     printf("\n");
16 }
```



Funções: Vetores

```
18 int main(void) {
19     int vetor[5], vetor2[10];
20     le_vetor(vetor,5);
21     le_vetor(vetor2,10);
22     imprime_vetor(vetor,5);
23     imprime_vetor(vetor2,10);
24     return 0;
25 }
```




Sumário

3 Exemplos



Sumário

3 Exemplos

- Produto escalar
- Par cuja multiplicação é \mathbb{C}
- Crivo de Eratóstenes



Exemplos

Produto Escalar

- Dado dois vetores de números ponto-flutuante, precisão dupla, computar o seu produto escalar.
- Sejam V_1 e V_2 , dois vetores de tamanho $n \leq 10$, o produto escalar é definido como:

$$\sum_{i=0}^{n-1} V_1[i] \cdot V_2[i]$$

- A soma dos produtos de cada elemento do primeiro vetor com o elemento na posição equivalente do segundo vetor.



Exemplos

Estratégia

- A estratégia aqui é utilizar uma variável acumuladora para registrar o produto escalar a cada iteração do laço.



Exemplos

Leitura

```
8     scanf("%d", &n);
9     printf("Leitura do primeiro vetor.\n");
10    for (i = 0; i < n; i++) {
11        printf("v1[%d] = ", i);
12        scanf("%lf", &v1[i]);
13    }
14    printf("Leitura do segundo vetor.\n");
15    for (i = 0; i < n; i++) {
16        printf("v2[%d] = ", i);
17        scanf("%lf", &v2[i]);
18    }
```



Exemplos

Cálculo

```
19  for (i = 0, soma = 0.0; i < n; i++) {  
20      soma += v1[i] * v2[i];  
21  }  
22  printf("O produto escalar é %.2f.\n",soma);
```



Exemplos

Programa Completo

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i, n;
5      double soma;
6      double v1[10], v2[10];
7      printf("Digite o tamanho dos vetores: ");
8      scanf("%d", &n);
9      printf("Leitura do primeiro vetor.\n");
10     for (i = 0; i < n; i++) {
11         printf("v1[%d] = ", i);
12         scanf("%lf", &v1[i]);
13     }
```



Exemplos

Programa Completo

```
14 printf("Leitura do segundo vetor.\n");
15 for (i = 0; i < n; i++) {
16     printf("v2[%d] = ", i);
17     scanf("%lf", &v2[i]);
18 }
19 for (i = 0, soma = 0.0; i < n; i++) {
20     soma += v1[i] * v2[i];
21 }
22 printf("O produto escalar é %.2f.\n", soma);
23 return 0;
24 }
```




Sumário

3 Exemplos

- Produto escalar
- Par cuja multiplicação é C
- Crivo de Eratóstenes



Exemplos

Exemplo

- Dado um vetor de inteiros de tamanho 10 e um inteiro C , encontrar dois elementos distintos do vetor cuja multiplicação é C e imprimi-los.
- Caso não haja esse par de elementos, uma mensagem deverá ser informada ao usuário.



Exemplos

Estratégia

- A estratégia aqui é checar todos os pares $(i, j), i \neq j$, de elementos do vetor e verificar se existe algum cuja multiplicação é C .
- Usamos dois laços aninhados e uma variável indicadora para verificar se o par já foi encontrado.
- Caso não haja tal par, uma mensagem é informada.



Exemplos

Leitura

```
7   for (i = 0; i < 10; i++) {
8       printf("v[%d] = ", i);
9       scanf("%d", &v[i]);
10  }
11  printf("Digite o valor de c: ");
12  scanf("%d", &c);
```



Exemplos

Cálculo

```
13  int encontrado = 0;
14  for (i = 0; i < 10 && !encontrado; i++) {
15      for (j = i + 1; j < 10 && !encontrado; j++) {
16          if (v[i] * v[j] == c) {
17              encontrado = 1;
18              printf("%d * %d = %d\n", v[i], v[j], c);
19          }
20      }
21  }
22  if (!encontrado) {
23      printf("Não existe um par de elementos que multiplicam %d", c);
24  }
```



Exemplos

Programa Completo

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i, j;
5      int v[10];
6      int c;
7      for (i = 0; i < 10; i++) {
8          printf("v[%d] = ", i);
9          scanf("%d", &v[i]);
10     }
11     printf("Digite o valor de c: ");
12     scanf("%d", &c);
```



Exemplos

Programa Completo

```
13  int encontrado = 0;
14  for (i = 0; i < 10 && !encontrado; i++) {
15      for (j = i + 1; j < 10 && !encontrado; j++) {
16          if (v[i] * v[j] == c) {
17              encontrado = 1;
18              printf("%d * %d = %d\n", v[i], v[j], c);
19          }
20      }
21  }
22  if (!encontrado) {
23      printf("Não existe um par de elementos que multiplicam %d", c);
24  }
25  return 0;
26  }
```



Sumário

3 Exemplos

- Produto escalar
- Par cuja multiplicação é \mathbb{C}
- Crivo de Eratóstenes



Exemplo: Crivo de Eratóstenes

- O crivo de Eratóstenes é um método que permite computar todos os números primos de um intervalo $[1, n]$.
- Ele funciona da seguinte forma:
 - ▶ Computa-se a raiz quadrada, arredondada para baixo, de n , chame o resultado de y . Este será o último número a ser checado.
 - ▶ Crie uma lista de todos os números naturais de 2 até n .
 - ▶ 2 é primo, então eliminamos todos os múltiplos de 2 desta lista.
 - ▶ O próximo número não eliminado, 3, tem que ser primo, repetimos o procedimento.
 - ▶ E assim fazemos para os demais números, até que o próximo número não eliminado seja maior que y . Os números que não foram eliminados são primos.



Exemplo: Crivo de Eratóstenes

| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|----------------------|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Prime numbers |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 2 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | |



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers
2 3



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2 3



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers
2 3 5



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2 3 5



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2 3 5 7



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2 3 5 7



Exemplo: Crivo de Eratóstenes

| | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 |
| 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 |
| 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 |
| 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 |
| 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 |
| 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 |

Prime numbers

2 3 5 7

11



Exemplo: Crivo de Eratóstenes



Prime numbers

2 3 5 7

11 13



Exemplo: Crivo de Eratóstenes



Prime numbers

2 3 5 7
11 13 17



Exemplo: Crivo de Eratóstenes



Prime numbers

2 3 5 7
11 13 17 19



Exemplo: Crivo de Eratóstenes

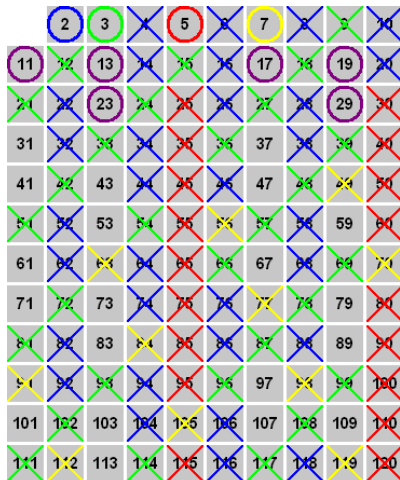


Prime numbers

2 3 5 7
11 13 17 19
23



Exemplo: Crivo de Eratóstenes



Prime numbers

2 3 5 7
11 13 17 19
23 29



Exemplo: Crivo de Eratóstenes

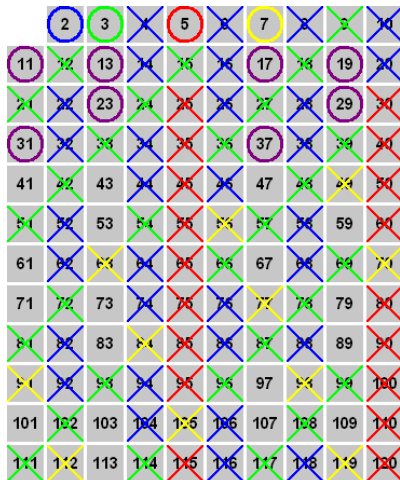


Prime numbers

2 3 5 7
11 13 17 19
23 29 31



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |



Exemplo: Crivo de Eratóstenes

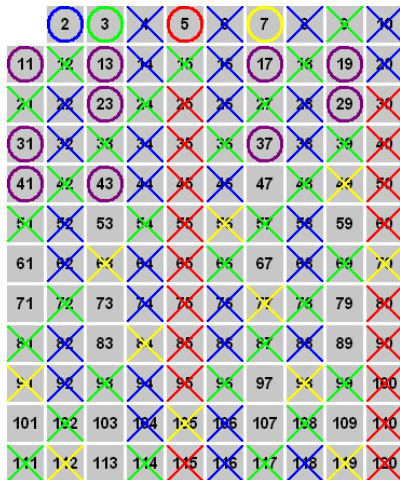


Prime numbers

2 3 5 7
 11 13 17 19
 23 29 31 37
 41



Exemplo: Crivo de Eratóstenes



Prime numbers

2 3 5 7
 11 13 17 19
 23 29 31 37
 41 43



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | | | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | | |



Exemplo: Crivo de Eratóstenes

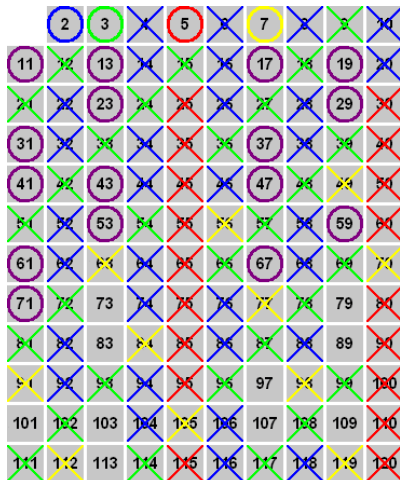


Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | | | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |



Exemplo: Crivo de Eratóstenes

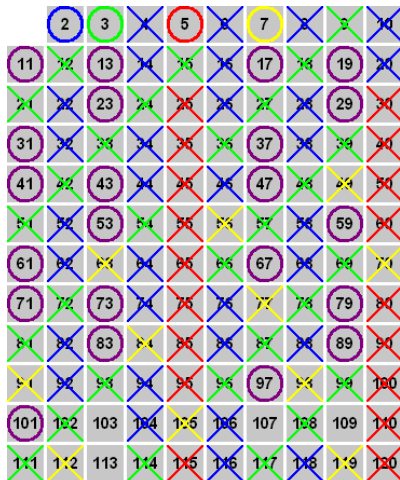


Prime numbers

| | | | |
|----|----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | | | |



Exemplo: Crivo de Eratóstenes

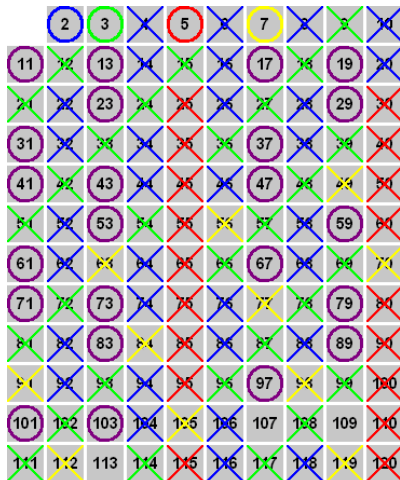


Prime numbers

| | | | |
|----|-----|----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | 101 | | |



Exemplo: Crivo de Eratóstenes

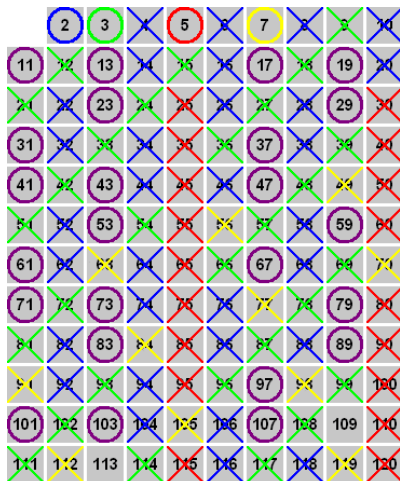


Prime numbers

| | | | |
|----|-----|-----|----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | 101 | 103 | |



Exemplo: Crivo de Eratóstenes

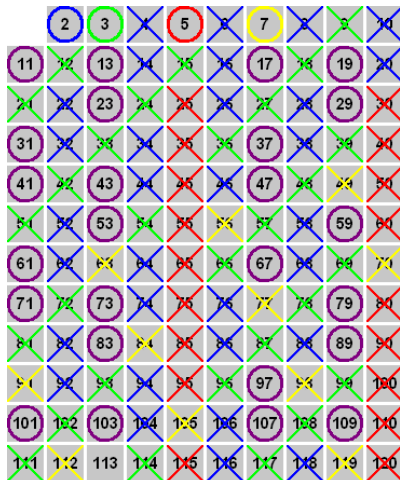


Prime numbers

| | | | |
|----|-----|-----|-----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | 101 | 103 | 107 |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|-----|-----|-----|-----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | 101 | 103 | 107 |
| 109 | | | |



Exemplo: Crivo de Eratóstenes



Prime numbers

| | | | |
|-----|-----|-----|-----|
| 2 | 3 | 5 | 7 |
| 11 | 13 | 17 | 19 |
| 23 | 29 | 31 | 37 |
| 41 | 43 | 47 | 53 |
| 59 | 61 | 67 | 71 |
| 73 | 79 | 83 | 89 |
| 97 | 101 | 103 | 107 |
| 109 | 113 | | |



Exemplo: Crivo de Eratóstenes

Problema

- Dado um inteiro n , imprimir todos os números primos do intervalo $[1, n]$.



Exemplo: Crivo de Eratóstenes

- Antes de tudo, vamos criar uma função que inicialize a nossa lista.
- Nossa lista será um vetor de inteiros `crivo`, e cada entrada i do nosso vetor será 1, indicando que o número ainda não foi eliminado.
- Quando o número for eliminado, basta fazer `crivo[i]=0`.
- Claramente, as entradas 0 e 1 tem que ser falsas, pois não são primos.

```
void inicializa(int crivo[], int n) {  
    int i;  
    crivo[0] = 0;  
    crivo[1] = 0;  
    for (i = 2; i <= n; i++) {  
        crivo[i] = 1;  
    }  
}
```



Exemplo: Crivo de Eratóstenes

- Agora, vamos criar uma função que elimine todos os múltiplos de um determinado número (maiores que este número) até o valor de n .

```
void elimina(int lista[],int n,int numero){  
    for(int i=numero*2;i<=n;i+=numero){  
        lista[i] = 0;  
    }  
}
```



Exemplo: Crivo de Eratóstenes

- Com posse dessas duas funções, podemos implementar o crivo.

```
void executa_crivo(int n){
    int crivo[n+1];
    inicializa_crivo(crivo,n);
    int i;
    for(i=2;i<=sqrt(n);i++){
        if(crivo[i]){
            elimina(crivo,n,i);
        }
    }
}
```




Exemplo: Crivo de Eratóstenes

- Por fim, podemos simplesmente varrer a nossa tabela e imprimir todos os números primos.

```
void imprime(int crivo[], int n) {
    int i;
    printf("Números primos de 1 a %d\n", n);
    for (i = 2; i <= n; i++) {
        if (crivo[i]) {
            printf("%d ", i);
        }
    }
    printf("\n");
}
```



Exemplo: Crivo de Eratóstenes

```
1  #include <math.h>
2  #include <stdio.h>
3
4  void inicializa(int crivo[], int n);
5  void elimina(int lista[], int n, int numero);
6  void imprime(int crivo[], int n);
7  void executa_crivo(int n);
8
9  int main(void) {
10     int n;
11     printf("O programa imprimirá todos os primos até um número n. Digite o "
12           "valor de n: ");
13     scanf("%d", &n);
14     executa_crivo(n);
15     return 0;
16 }
```



Exemplo: Crivo de Eratóstenes

```
18 void inicializa(int crivo[], int n) {
19     int i;
20     crivo[0] = 0;
21     crivo[1] = 0;
22     for (i = 2; i <= n; i++)
23         crivo[i] = 1;
24 }
25
26 void elimina(int lista[], int n, int numero) {
27     for (int i = numero * 2; i <= n; i += numero)
28         lista[i] = 0;
29 }
30
31 void imprime(int crivo[], int n) {
32     int i;
33     printf("Números primos de 1 a %d\n", n);
34     for (i = 2; i <= n; i++) {
35         if (crivo[i]) {
36             printf("%d ", i);
37         }
38     }
39     printf("\n");
40 }
```