

Aula 04: Laboratório
- Funções e procedimentos
- Vetores e matrizes

Prof. Jesús P. Mena-Chalco
jesus.mena@ufabc.edu.br

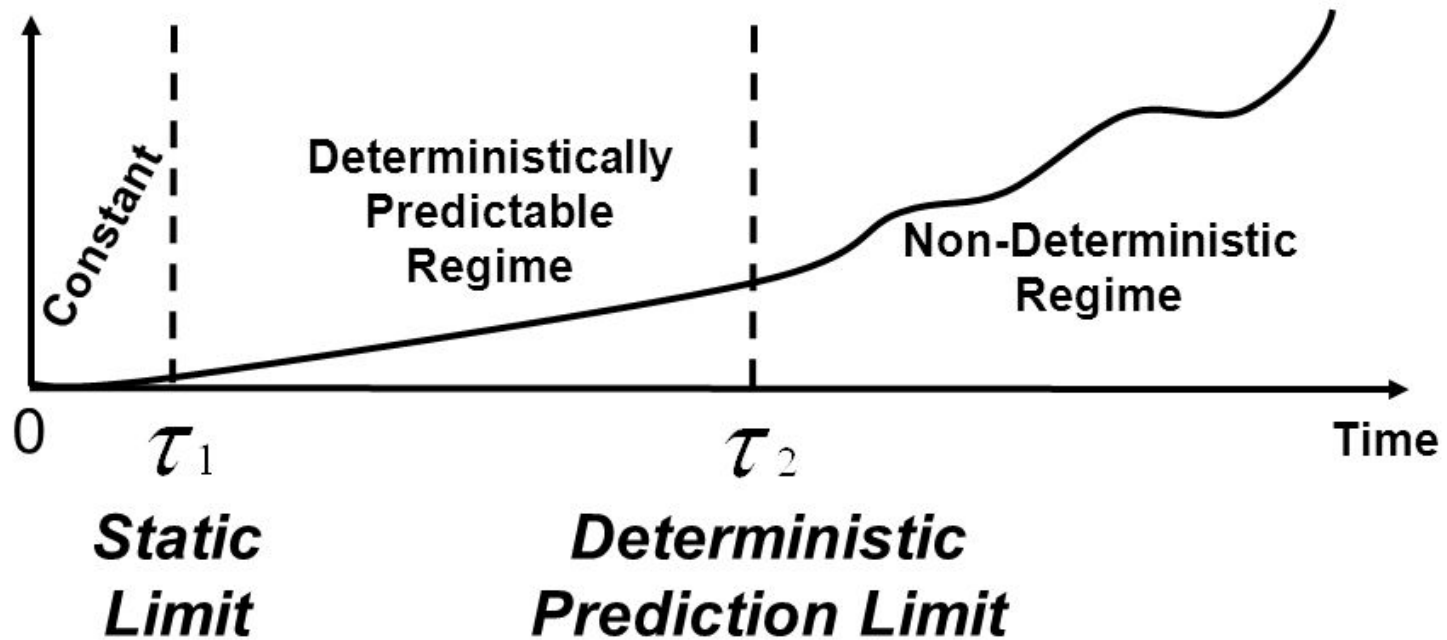
3Q-2017

Weather Time Scale Analysis

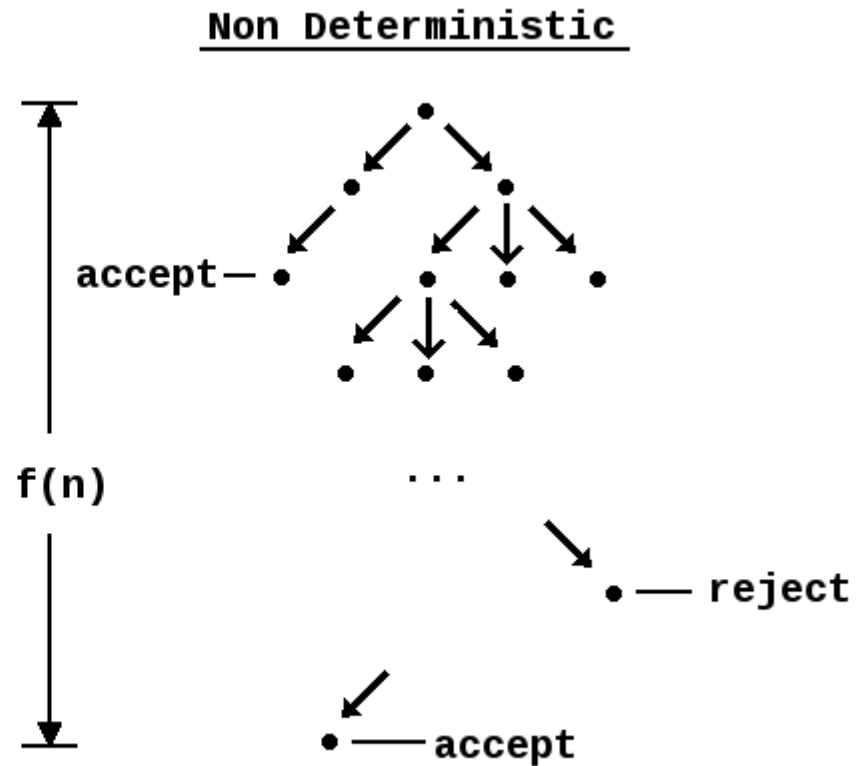
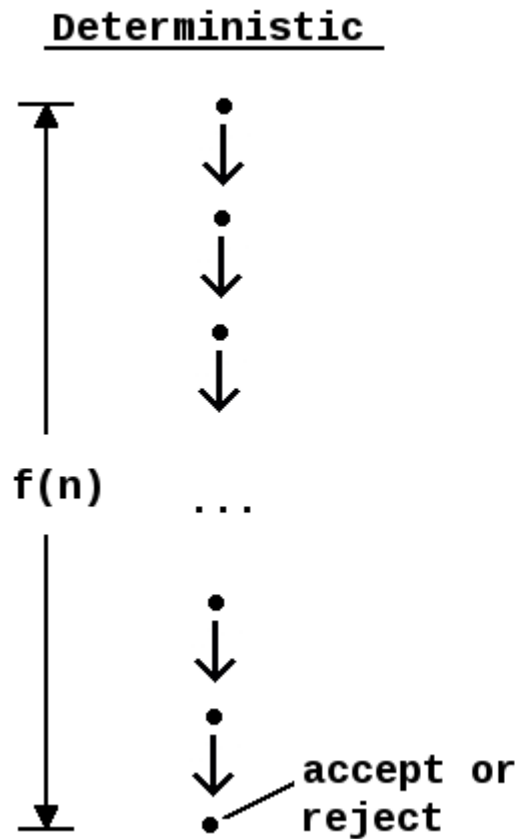
Concept



Error in
Phenomena
Knowledge



Algoritmos: Determinísticos Vs Não-determinísticos



Um programa **determinístico** sempre gera a mesma saída para o mesmo conjunto de entrada.

Um programa é **não-determinístico** quando apresenta resultados diferentes com os mesmos conjuntos de entrada.

Procedimento: exemplo 1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  void main () {
5      int numero;
6
7      numero = rand();
8      printf("%d\n", numero);
9  }
```

int rand(void)

Devolve um número inteiro entre 0 e RAND_MAX

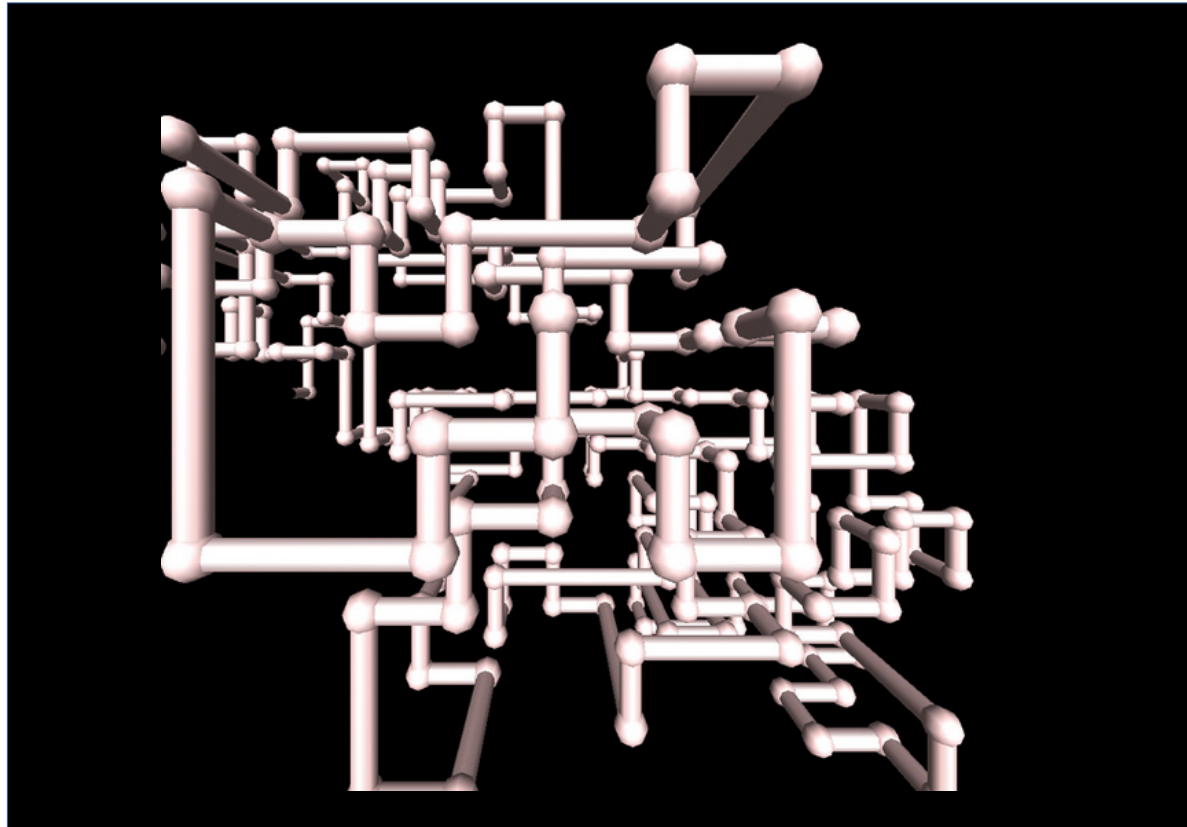
Números aleatórios



Me da um
número
aleatório



Pronto



stdlib.h

Números aleatórios

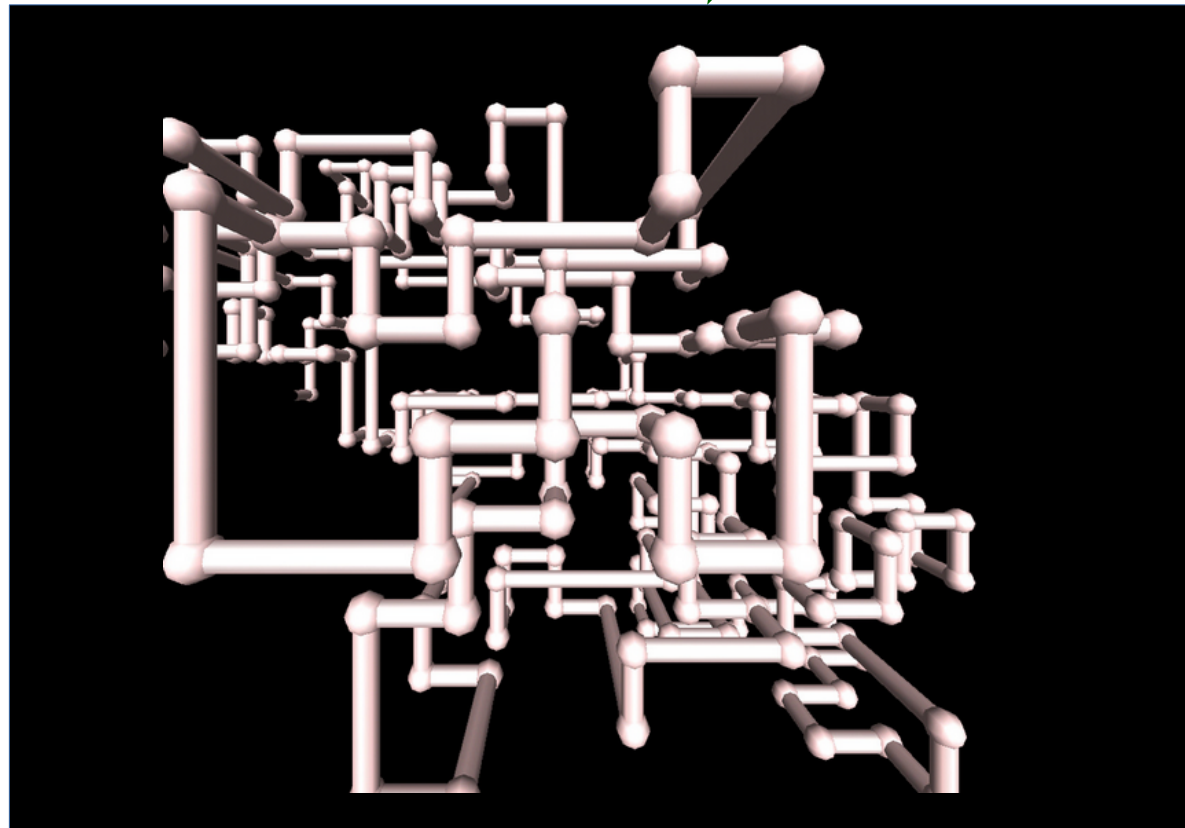
Temperatura
Hora atual do sistema
Número de rotações do HD
...



Me da um
número
aleatório



Pronto



stdlib.h

Números aleatórios

[Home](#) [Games](#) [Numbers](#) [Lists & More](#) [Drawings](#) [Web Tools](#) [Statistics](#) [Testimonials](#) [Learn More](#) [Login](#)

RANDOM.ORG

Search RANDOM.ORG

Google™ Custom Search

Search

True Random Number Service

Do you own an iOS or Android device? [Check out our app!](#)

Note: Our much improved new API is currently in public beta - learn more on [api.random.org](#)

HTTP Interface Description

RANDOM.ORG is a true random number service that generates randomness via atmospheric noise. This page explains how to interface to the service via the Hyper-Text Transfer Protocol (HTTP). There is also the [HTTP Client Archive](#), which contains clients that other people have written.



Important note!

If you access RANDOM.ORG via an automated client, please make sure you observe the [Guidelines for Automated Clients](#) or your computer may be banned.

If you are writing a general-purpose client, please make sure it is easy for your users to run it in accordance with the guidelines.

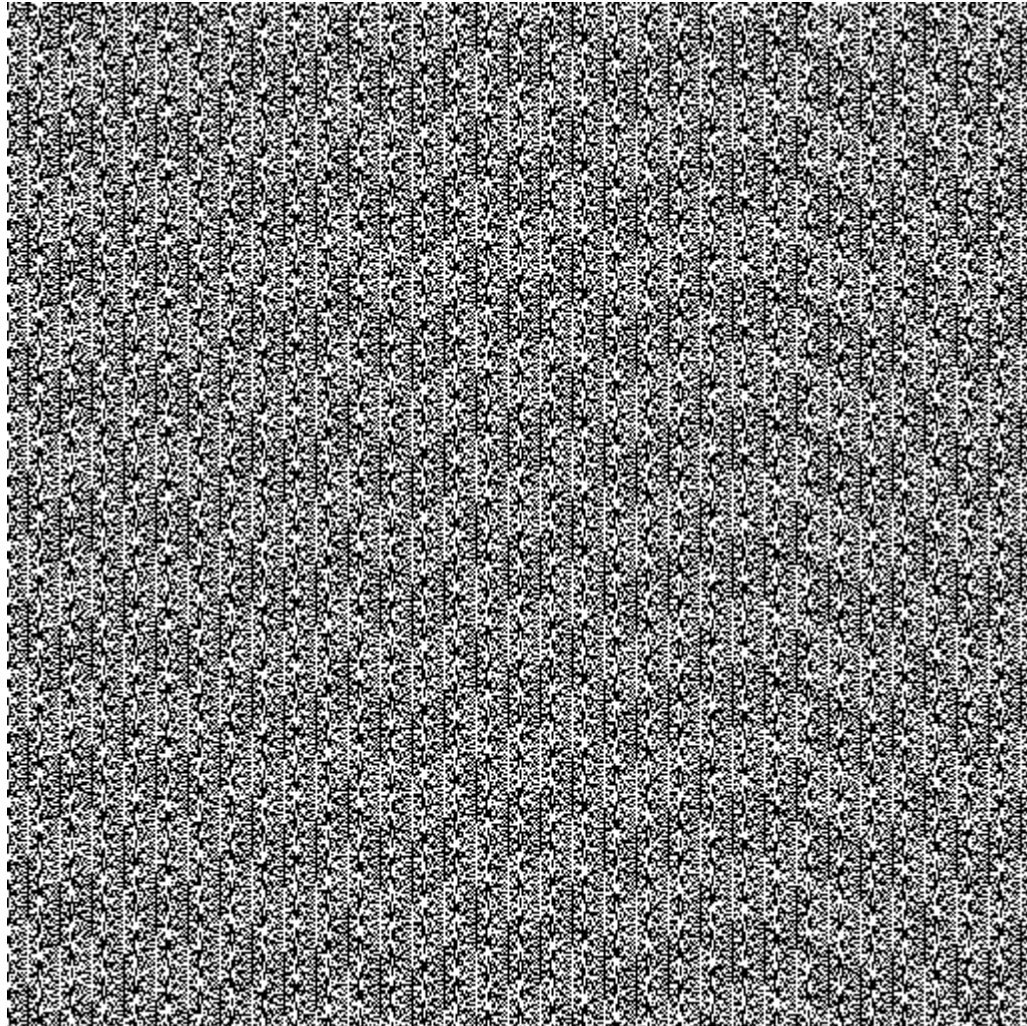
This page contains documentation for the [Integer Generator](#), the [Sequence Generator](#), the [String Generator](#) and the [Quota Checker](#), which allows you to examine your current bit allowance.

All the interfaces on this page return HTTP status code 503 (Service Unavailable) in the case of errors and code 200 (OK) when successful. Not all languages allow you to access the HTTP status codes in a straightforward manner. A reasonable workaround is to look for the string "Error:" (don't forget the colon) as the first line of the response. This will work for all the generators on this page, including the [String Generator](#) (which could by chance produce the string "Error" in a successful response, but which cannot produce the colon character).

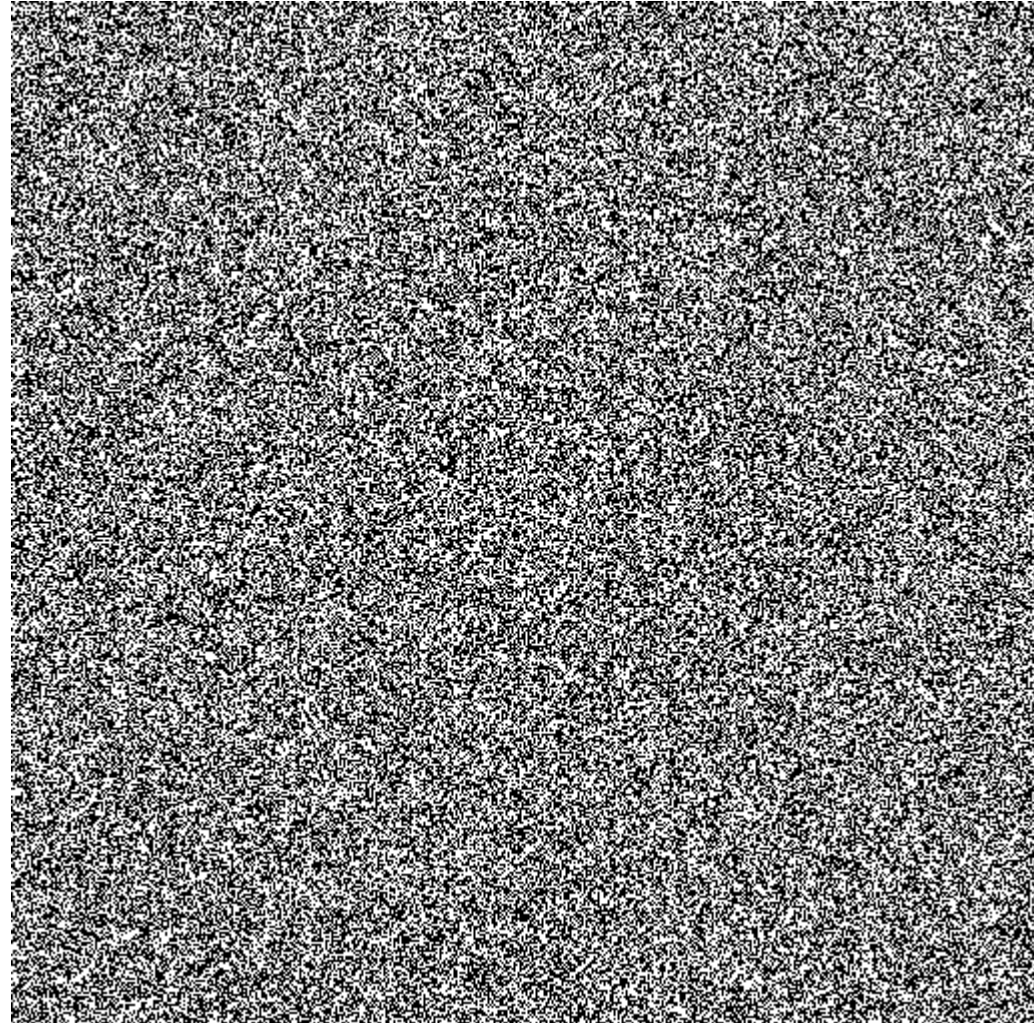
Please note that the old CGI scripts (randbyte, randnum, etc.) are no longer supported and you should use the ones described below instead. In particular, the old scripts do not return the 503 status code in case of errors (they return the 200 response code in all cases), so please use the new ones instead.

Números aleatórios

Pseudo-random



True-random



Procedimento: exemplo 1

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void main () {
6      int numero;
7
8      srand(time(NULL));
9
10     numero = rand();
11     printf("%d\n", numero);
12 }
```

int rand(void)

Devolve um número inteiro entre 0 e RAND_MAX

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void main () {
6      int numero;
7
8      srand(time(NULL));
9
10     numero = rand()%17;
11     printf("%d\n", numero);
12 }
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void main () {
6      int numero;
7
8      srand(time(NULL));
9
10     numero = rand()%17;
11     printf("%d\n", RAND_MAX)
12 }
```

Procedimento: exemplo 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void linha(int n) {
5     int i;
6     int posicao = rand()%n;
7
8     for (i=0; i<n; i++)
9         if (posicao==i)
10            printf("-");
11        else
12            printf("*");
13    printf("\n");
14 }
15
16 void main() {
17     linha(10);
18     linha(10);
19 }
```

int rand(void)

Devolve um número inteiro entre 0 e RAND_MAX

```
*****_***
*****_***
```

Procedimento: exemplo

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void linha(int n) {
6     int i;
7     int posicao = rand()%n;
8
9     for (i=0; i<n; i++)
10        if (posicao==i)
11            printf("-");
12        else
13            printf("*");
14    printf("\n");
15 }
16
17 void main() {
18     srand(time(NULL));
19     linha(10);
20     linha(10);
21 }
```

```
****_*****
*****_****
```

Desafio!

Modifique o programa para que sejam apresentadas consecutivamente os pares de linhas desde que o elemento selecionado em cada linha seja diferente.

Isto é, o programa deve parar quando as duas linhas sejam iguais.

Quantas pares de linhas foram apresentadas?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 void linha(int n) {
6     int i;
7     int posicao = rand()%n;
8
9     for (i=0; i<n; i++)
10        if (posicao==i)
11            printf("-");
12        else
13            printf("*");
14    printf("\n");
15 }
16
17 void main() {
18     srand(time(NULL));
19     linha(10);
20     linha(10);
21 }
```

Desafio da aula anterior

Modifique o programa para que sejam apresentadas consecutivamente os pares de linhas desde que o elemento selecionado em cada linha seja diferente.

Isto é, o programa deve parar quando as duas linhas sejam iguais.

Quantas pares de linhas foram apresentadas?

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int linha(int n) {
6     int i;
7     int posicao = rand()%n;
8
9     for (i=0; i<n; i++)
10         if (posicao==i)
11             printf("-");
12         else
13             printf("*");
14     printf("\n");
15     return posicao;
16 }
17
18 int linhas() {
19     int l1, l2, cont=0;
20
21     do {
22         l1 = linha(10);
23         l2 = linha(10);
24         cont++;
25     } while (l1!=l2);
26
27     return cont;
28 }
29
30 void main() {
31     srand(time(NULL));
32     printf("%d\n", linhas());
33 }
```

PI: John Wallis

Crie um programa para calcular o valor de PI seguindo a Identidade de John Wallis (1655). Considere como parâmetro o número de termos na produtória.

$$\prod_{n=1}^{\infty} \left(\frac{2n}{2n-1} \cdot \frac{2n}{2n+1} \right) = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \cdots = \frac{\pi}{2}$$

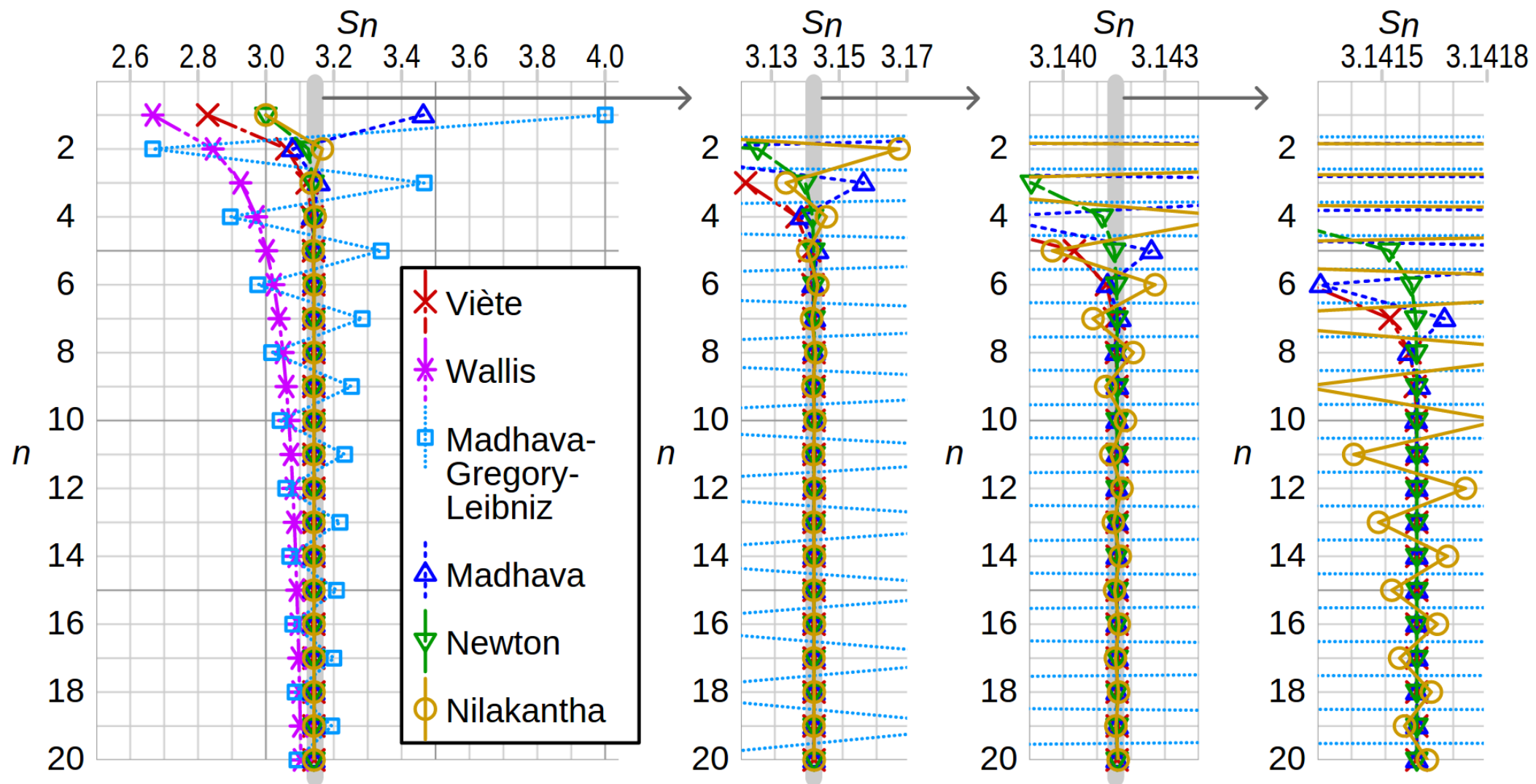
```
1 #include <stdio.h>
2
3 double pi(int t) {
4
5
6
7
8
9
10
11 }
12
13 void main() {
14     printf("%lf\n", pi(1));
15     printf("%lf\n", pi(10));
16     printf("%lf\n", pi(100));
17     printf("%lf\n", pi(1000));
18 }
```

PI: John Wallis

$$\prod_{n=1}^{\infty} \left(\frac{2n}{2n-1} \cdot \frac{2n}{2n+1} \right) = \frac{2}{1} \cdot \frac{2}{3} \cdot \frac{4}{3} \cdot \frac{4}{5} \cdot \frac{6}{5} \cdot \frac{6}{7} \cdot \frac{8}{7} \cdot \frac{8}{9} \cdots = \frac{\pi}{2}$$

```
1  #include <stdio.h>
2
3  double pi(int t) {
4      int i;
5      double produto = 1;
6
7      for (i=1; i<=t; i++)
8          produto *= (4.0*i*i)/((2*i-1)*(2*i+1));
9
10     return produto*2;
11 }
12
13 void main() {
14     printf("%lf\n", pi(1));
15     printf("%lf\n", pi(10));
16     printf("%lf\n", pi(100));
17     printf("%lf\n", pi(1000));
18 }
```

Aproximação de pi



Linha na Matriz

Por Neilor Tonin, URI  Brasil**Timelimit: 1**

Neste problema você deve ler um número, indicando uma linha da matriz na qual uma operação deve ser realizada, um caractere maiúsculo, indicando a operação que será realizada, e todos os elementos de uma matriz $M[12][12]$. Em seguida, calcule e mostre a soma ou a média dos elementos que estão na área verde da matriz, conforme for o caso. A imagem abaixo ilustra o caso da entrada do valor 2 para a linha da matriz, demonstrando os elementos que deverão ser considerados na operação.

	0	1	2	3	4	5	6	7	8	9	10	11
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												

Entrada

A primeira linha de entrada contém um número L ($0 \leq L \leq 11$) indicando a linha que será considerada para operação. A segunda linha de entrada contém um único caractere Maiúsculo T ('S' ou 'M'), indicando a operação (Soma ou Média) que deverá ser realizada com os elementos da matriz. Seguem os 144 valores de ponto flutuante que compõem a matriz, sendo que a mesma é preenchida linha por linha, da linha 0 até a linha 11, sempre da esquerda para a direita.

Saída

Imprima o resultado solicitado (a soma ou média), com 1 casa após o ponto decimal.

Exemplo de Entrada	Exemplo de Saída
2	12.6
S	
0.0	
-3.5	
2.5	
4.1	
...	

```

1181.c x
1 #include <stdio.h>
2
3 int main() {
4
5     int linha, i, j;
6     double m[12][12], soma = 0.0;
7     char operacao;
8     scanf("%d %c", &linha, &operacao);
9     for (i = 0; i < 12; i++){
10         for (j = 0; j < 12; j++){
11             scanf("%lf", &m[i][j]);
12             if (linha == i)
13                 soma += m[i][j];
14         }
15     }
16
17     if (operacao == 'S')
18         printf("%.1f\n", soma);
19     else if (operacao == 'M')
20         printf("%.1f\n", soma/12.0);
21
22     return 0;
23 }

```

Exemplo de solução para o Problema 1181 (Linha na Matriz).