

## 素数の判定

```
1 /* 素数の判定 */
2 #include<stdio.h>
3 #include<math.h>
4
5 main()
6 {
7     int p,q,x;
8     int prime=1;
9
10    printf("input the number ");
11    scanf("%d",&x);
12    q=sqrt(x);
13
14    if ( x%2 == 0 )
15        prime=0;
16    else
17        {
18            p=3;
19            while( p <= q )
20                {
21                    if (x%p == 0)
22                        {
23                            prime=0;
24                            break;
25                        }
26                    else
27                        p=p+2;
28                }
29        }
30
31    if ( prime == 1 )
32        printf("%d is a prime.\n",x);
33    else
34        {
35            printf("%d is not a prime.\n",x);
36            printf("It is divisible by %d.\n",p);
37        }
38 }
```

### 実行例

```
input the number 2819
2819 is a prime.
```

```
input the number 8413901
8413901 is not a prime.
It is divisible by 1319.
```

```
input the number 2147483647
2147483647 is a prime.
```

## ニュートン法

```
1 /* ニュートン法 */
2 #include <stdio.h>
3 #include <math.h>
4 #include "func.c"
5
6 main()
7 {
8     int i=0;
9     double x0,x1,eps=1.0e-15;
10
11     x0=3.5;
12     while ( fabs(f(x0)) >= eps )
13     {
14         printf(" %d: x=%19.15f  f(x)=%11.3e\n",i,x0,f(x0));
15         x1=x0-f(x0)/d(x0);
16         i=i+1;
17         x0=x1;
18     }
19     printf(" %d: x=%19.15f  f(x)=%11.3e\n",i,x0,f(x0));
20 }
1 /* func.c */
2 double f(double x)
3 {
4     return (x*x-2.0);
5 }
6
7 double d(double x)
8 {
9     return(2.0*x);
10 }
```

### 実行例

```
0: x= 3.5000000000000000 f(x)= 1.025e+01
1: x= 2.035714285714286 f(x)= 2.144e+00
2: x= 1.509085213032581 f(x)= 2.773e-01
3: x= 1.417195710107738 f(x)= 8.444e-03
4: x= 1.414216699979654 f(x)= 8.875e-06
5: x= 1.414213562376576 f(x)= 9.845e-12
6: x= 1.414213562373095 f(x)= 2.734e-16
```

## モンテカルロ法

```
1 /* モンテカルロ方による $\pi$ の計算 */
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <math.h>
5 #define pi 3.1415926535
6
7 main()
8 {
9     int n,n0,n1,in=0,ip=1;
10    float x,y,p;
11
12    n0=1;
```

```

13  n1=10;
14
15  while (ip<=9)
16  {
17      for (n=n0; n<=n1; n++)
18      {
19          x=(float) rand()/RAND_MAX;
20          y=(float) rand()/RAND_MAX;
21          if (x*x+y*y <= 1) in++;
22      }
23      p=4*(float) in/(float) n1;
24      printf("%2d %8f %11.3e \n",ip,p,fabs(p-pi));
25
26      n0=n1+1;
27      n1=n1*10;
28      ip++;
29  }
30 }

```

### 実行例

ip	4*in/n1	誤差
1	3.200000	5.841e-02
2	3.120000	2.159e-02
3	3.132000	9.593e-03
4	3.171200	2.961e-02
5	3.141520	7.263e-05
6	3.141664	7.137e-05
7	3.141130	4.627e-04
8	3.141698	1.052e-04
9	3.141604	1.105e-05