

```

1 #include <stdio.h>
2
3 #define N 200      /* 求める桁数 5*N 桁 */
4 #define DEG 100000 /* 桁数の基準 */
5
6 void init(int a[], int n);
7 void adds(int a[], int b[], int c[]); /* c = a+b */
8 void subs(int a[], int b[], int c[]); /* c = a-b */
9 void divsp(int a[], int b[], int n, int p); /* b = a/n */
10 int top(int a[], int p); /* 0 でない最左位置, p が大きいほど右へ */
11 void print_result(int a[]); /* 表示用関数 */
12 void arctan(int a[], int n, int d); /* n*arctan(1/d) の計算 */
13
14 int main(void)
15 {
16
17     int a[N+1], b[N+1], pi[N+1];
18
19     arctan(a, 16, 5);
20     arctan(b, 4, 239);
21
22     subs(a,b,pi);
23     print_result(pi);
24
25     return 0;
26 }
27
28 void adds(int a[], int b[], int c[])
29 {
30     int up, i, sum;
31     /* 加算 */
32     up = 0;
33     for ( i=N; i>=0; i--)
34     {
35         sum = a[i]+b[i]+up;
36         if ( sum > DEG-1 )
37         {
38             c[i] = sum - DEG;
39             up = 1;
40         }
41         else
42         {
43             c[i]=sum; up = 0;
44         }
45     }
46 }
47
48 void subs(int a[], int b[], int c[])
49 {
50     int borrow, i, sub;
51     /* 減算 */
52     borrow = 0;
53     for ( i=N; i>=0; i--)
54     {
55         sub = a[i]-b[i]-borrow;
56         if ( sub >= 0)
57         {
58             c[i] = sub; borrow = 0;
59         }
60         else
61         {
62             c[i] = DEG + sub;
63             borrow = 1;
64         }
65     }
66 }
67
68 void divsp(int a[], int b[], int bunbo, int p)
69 {
70     /* 除算 */
71     int amari, i, bunshi; /* p は配列の番号 */
72     amari = 0;
73     for ( i=p ; i<=N; i++)
74     {
75         bunshi = amari * DEG + a[i];
76         b[i] = bunshi/bunbo;
77         amari = bunshi % bunbo;
78     }
79 }
80
81 void print_result(int a[])
82 {
83     int i;
84
85     printf("%Su.",a[0]);
86     for ( i = 1; i <= N; i++)
87     {
88         printf("%05u ",a[i]);
89     }
90     printf("\n");
91 }
92
93 void arctan(int a[], int n, int d)
94 {
95     int e[N+1], f[N+1], p, i;
96
97     init(a,0); init(e,n);
98
99     /* (**) の第 1 項の計算 */
100    divsp(e, e, d, 0);
101    adds(a, e, a);
102    p = top(e,0);
103
104    /* (**) の第 2 項の計算 */
105    i = 3;
106    while( p <= N)
107    {
108        divsp(e, e, d, p);
109        divsp(e, e, d, p);
110        divsp(e, f, i, p);
111        if (i%4 ==1 )
112        {
113            adds(a, f, a);
114        }
115        else
116        {
117            subs(a, f, a);
118        }
119        p = top(e, p);
120        i += 2;
121    }
122 }
123
124 /* 0 でない最左位置 p を探す */
125 int top(int a[], int p)
126 {
127     while( p<= N && a[p] == 0) p++;
128     return p;
129 }
130
131 /* 値を初期化 a[0]=n で、それ以外は 0 */
132 void init(int a[], int n)
133 {
134     int i;
135
136     a[0] = n;
137     for( i=1; i<=N; i++)
138     {
139         a[i] = 0;
140     }
141 }

```