МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Качество и метрология программного обеспечения»

Тема: «Измерение характеристик динамической сложности программ с помощью профилировщика SAMPLER»

Студент гр. 7304 Моторин Е.В.

Преподаватель Ефремов М.А.

Санкт-Петербург

2021

Задание

1. Ознакомиться с документацией на монитор SAMPLER и выполнить под его управлением тестовые программы test_cyc.c и test_sub.c с анализом параметров повторения циклов, структуры описания циклов, способов профилирования процедур и проверкой их влияния на точность и чувствительность профилирования.

2. Скомпилировать и выполнить под управлением SAMPLER'a программу на C, разработанную в 1-ой лабораторной работе.

Выполнить разбиение программы на функциональные участки и снять профили для двух режимов:

1 - измерение только полного времени выполнения программы;

2 - измерение времен выполнения функциональных участков (ФУ).

Убедиться, что сумма времен выполнения ФУ соответствует полному времени выполнения программы.

3. Выявить "узкие места", связанные с ухудшением производительности программы, ввести в программу усовершенствования и получить новые профили. Объяснить смысл введенных модификаций программ.

Ход работы

Программы транслировались с использованием компилятора Borland C++ v. 3.1. Профилирование выполнялось с помощью sampler_old, который запускался на 64-разрядной виртуальной машине под управлением ОС Windows 10.

Тестовые программы

Код программы test_cyc.c с нумерацией строк представлен в приложении А.

Результаты профилирования программы test_cyc.c:

Таблица с результатами измерений (используется 13 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 8	1 : 10	4336.31	1	4336.31
1 : 10	1 : 12	8668.43	1	8668.43
1 : 12	1 : 14	21672.34	1	21672.34
1 : 14	1 : 16	43348.03	1	43348.03
1 : 16	1 : 19	4334.64	1	4334.64
1 : 19	1 : 22	8670.11	1	8670.11
1 : 22	1 : 25	21676.53	1	21676.53
1 : 25	1 : 28	43348.03	1	43348.03
1 : 28	1 : 34	4336.31	1	4336.31
1 : 34	1 : 40	8669.27	1	8669.27
1 : 40	1 : 46	21673.18	1	21673.18
1 : 46	1 : 52	43348.03	1	43348.03

Исходя из результата, на время выполнения влияет только количество итераций цикла (линейная зависимость). Циклы с одинаковым количеством итераций демонстрируют почти одно и то же время выполнения.

Код программы test_sub.c с нумерацией строк представлен в приложении Б.

Результаты профилирования программы test_sub.c:

Таблица с результатами измерений (используется 5 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 24	1 : 26	433698.18	1	433698.18
1 : 26	1 : 28	867392.18	1	867392.18
1 : 28	1 : 30	2168467.46	1	2168467.46
1 : 30	1 : 32	4336936.59	1	4336936.59

Результаты демонстрируют линейную зависимость времени выполнения функции от количества итераций цикла внутри функции. Время выполнения на два порядка больше, чем в предыдущей программе, не смотря на меньшее количество итераций, – это связано с тем, что в основном цикле каждую итерацию выполняется вложенный цикл.

Программа из первой лабораторной работы

Код программы из первой лабораторной работы с нумерацией строк представлен в приложениях В (для измерения полного времени) и Г (для измерения времен выполнения ФУ) при различных входных параметрах.

1) X = 5, ordr = 1:

Результаты профилирования с измерением полного времени:

Таблица с результатами измерений (используется 2 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 82	1 : 84	2722.98	1	2722.98

Результаты профилирования с измерением времен ФУ:

Таблица с результатами измерений (используется 20 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 17	1 : 19	70.40	1	70.40
1 : 19	1 : 27	252.27	1	252.27
1 : 27	1 : 29	0.00	1	0.00
1 : 29	1 : 31	15.92	10	1.59
1 : 31	1 : 37	1.68	1	1.68
1 : 31	1 : 33	10.90	9	1.21
1 : 33	1 : 35	113.98	9	12.66
1 : 35	1 : 37	5.87	9	0.65
1 : 37	1 : 41	508.72	10	50.87

1 : 41	1 : 29	222.10	9	24.68
1 : 41	1 : 43	9.22	1	9.22
1 : 43	1 : 48	106.44	1	106.44
1 : 48	1 : 50	0.84	1	0.84
1 : 50	1 : 56	781.94	12	65.16
1 : 56	1 : 50	212.04	11	19.28
1 : 56	1 : 58	10.06	1	10.06
1 : 58	1 : 61	156.72	1	156.72
1 : 61	1 : 68	88.00	1	88.00
1 : 68	1 : 70	88.84	1	88.84
1 : 70	1 : 72	2.51	1	2.51
1 : 72	1 : 92	1.68	1	1.68
1 : 92	1 : 94	0.84	1	0.84
1 : 94	1 : 119	0.84	1	0.84

Суммарное время всех ФУ (2661,81 мкс) не сильно отличается (примерно 2.3%) от полного времени выполнения функции (2722,98 мкс).

2) X = 12, ordr = 1

Результаты профилирования с измерением полного времени:

Таблица с результатами измерений (используется 2 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 82	1 : 84	553.14	1	553.14

Результаты профилирования с измерением времен ФУ:

Таблица с результатами измерений (используется 3 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 17	1 : 98	70.40	1	70.40
1 : 98	1 : 119	491.39	1	391.39

Суммарное время всех ФУ (561.79 мкс) немного отличается (примерно 1.5%) от полного времени выполнения функции (553.14 мкс).

3) X = 5, ordr = 3

Результаты профилирования с измерением полного времени:

Таблица с результатами измерений (используется 2 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 82	1 : 84	3209.07	1	3209.07

Результаты профилирования с измерением времен ФУ:

Таблица с результатами измерений (используется 20 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 17	1 : 19	69.56	1	69.56
1 : 19	1 : 27	251.43	1	251.43
1 : 27	1 : 29	0.00	1	0.00
1 : 29	1 : 31	6.70	10	0.67
1 : 31	1 : 37	0.00	1	0.00
1 : 31	1 : 33	3.35	9	0.37
1 : 33	1 : 35	106.44	9	11.83
1 : 35	1 : 37	0.00	9	0.00
1 : 37	1 : 41	501.18	10	50.12
1 : 41	1 : 29	215.39	9	23.93
1 : 41	1 : 43	9.22	1	9.22
1 : 43	1 : 48	106.44	1	106.44
1 : 48	1 : 50	0.00	1	0.00
1 : 50	1 : 56	771.89	12	64.32
1 : 56	1 : 50	201.98	11	18.36
1 : 56	1 : 58	9.22	1	9.22
1 : 58	1 : 61	155.89	1	155.89
1 : 61	1 : 68	88.00	1	88.00

1 : 68	1 : 76	87.16	1	87.16
1 : 76	1 : 80	71.24	1	71.24
1 : 80	1 : 83	288.31	1	288.31
1 : 83	1 : 87	115.66	2	57.83
1 : 87	1 : 83	20.95	1	20.95
1 : 87	1 : 89	20.95	1	20.95
1 : 89	1 : 92	1.68	1	1.68
1 : 92	1 : 94	0.00	1	0.00
1 : 94	1 : 119	0.00	1	0.00

Суммарное время всех ФУ (3102,64 мкс) не сильно отличается (примерно 3.3%) от полного времени выполнения функции (3209,07 мкс).

Измененная программа из первой лабораторной работы

В программу были добавлены улучшения: упрощены записи некоторых выражений, убраны лишние переменные. Измененный код программы из первой лабораторной работы с нумерацией строк представлен в приложениях Д (для измерения полного времени) и Е (для измерения времени выполнения ФУ).

Результаты профилирования с измерением полного времени:

Таблица с результатами измерений (используется 2 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 79	1 : 81	2621.57	1	2621.57

Общее время выполнения функции уменьшилось на 101.41 мкс или на 3.7%.

Результаты профилирования с измерением времен ФУ:

Таблица с результатами измерений (используется 20 из 416 записей)

Исх.Поз.	Прием.Поз.	Общее время(мкс)	Кол-во прох.	Среднее время(мкс)
1 : 20	1 : 22	71.24	1	71.24
1 : 22	1 : 30	251.43	1	251.43
1 : 30	1 : 32	0.84	1	0.84
1 : 32	1 : 34	9.22	10	0.92
1 : 34	1 : 40	0.84	1	0.84
1 : 34	1 : 36	5.03	9	0.56
1 : 36	1 : 38	107.28	9	11.92
1 : 38	1 : 40	0.00	9	0.00
1 : 40	1 : 44	502.02	10	50.20
1 : 44	1 : 32	215.39	9	23.93
1 : 44	1 : 46	8.38	1	8.38
1 : 46	1 : 51	105.60	1	105.60
1 : 51	1 : 53	0.00	1	0.00
1 : 53	1 : 59	769.37	12	64.11
1 : 59	1 : 53	198.63	11	18.06
1 : 59	1 : 61	9.22	1	9.22
1 : 61	1 : 64	155.89	1	155.89
1 : 64	1 : 71	87.16	1	87.16
1 : 71	1 : 73	87.16	1	87.16
1 : 73	1 : 75	0.84	1	0.84
1 : 75	1 : 95	0.00	1	0.00
1 : 95	1 : 97	0.00	1	0.00

Общее время на каждом из участков стало меньше (ФУ: было – 2655.94, стало – 2585.84, время выполнения уменьшилось на 2.5%, измерение полного времени: было - 2722.98, стало – 2621.57, время выполнения уменьшилось на 3.7%).

Выводы

В ходе лабораторной работы изучен монитор SAMPLER. Выполнено профилирование тестовых программ test_cyc.c и test_sub.c, которое показало линейную зависимость между временем выполнения программы и количеством итераций цикла.

Проанализировано полное время выполнения программы, разработанной в 1-ой лабораторной работе, и время выполнения её ФУ, за счет чего удалось частично усовершенствовать производительность.

ПРИЛОЖЕНИЕ А

TEST_CYC.C

```
1   #include <stdlib.h>
2   #include "Sampler.h"
3   #define Size 10000
4   int i, tmp, dim[Size];
5
6   void main()
7   {
8      SAMPLE;
9      for(i=0;i<Size/10;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
10     SAMPLE;
11     for(i=0;i<Size/5;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
12     SAMPLE;
13     for(i=0;i<Size/2;i++){ tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
14     SAMPLE;
15     for(i=0;i<Size;i++)  { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
16     SAMPLE;
17     for(i=0;i<Size/10;i++)
18     { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
19     SAMPLE;
20     for(i=0;i<Size/5;i++)
21     { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
22     SAMPLE;
23     for(i=0;i<Size/2;i++)
24     { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
25     SAMPLE;
26     for(i=0;i<Size;i++)
27     { tmp=dim[0]; dim[0]=dim[i]; dim[i]=tmp; };
28     SAMPLE;
29     for(i=0;i<Size/10;i++)
30     { tmp=dim[0];
31     dim[0]=dim[i];
32     dim[i]=tmp;
33     };
34     SAMPLE;
35     for(i=0;i<Size/5;i++)
36     { tmp=dim[0];
37     dim[0]=dim[i];
38     dim[i]=tmp;
39     };
40     SAMPLE;
41     for(i=0;i<Size/2;i++)
42     { tmp=dim[0];
43     dim[0]=dim[i];
44     dim[i]=tmp;
45     };
46     SAMPLE;
47     for(i=0;i<Size;i++)
48     { tmp=dim[0];
49     dim[0]=dim[i];
50     dim[i]=tmp;
51     };
52     SAMPLE;
53  }
```

ПРИЛОЖЕНИЕ Б

TEST_SUB.C

```
1 #include "Sample.h"
2 const unsigned Size = 1000;
3
4
5   void TestLoop(int nTimes)
6   {
7   static int TestDim[Size];
8   int tmp;
9   int iLoop;
10
11     while (nTimes > 0)
12     {
13     nTimes --;
14
15      iLoop = Size;
16      while (iLoop > 0)
17      {
18      iLoop -- ;
19      tmp = TestDim[0];
20      TestDim[0] = TestDim[nTimes];
21      TestDim[nTimes] = tmp;
22      }
23      }
24      } /* TestLoop */
25
26
27   void main()
28   {
29   SAMPLE;
30   TestLoop(Size / 10); // 100 * 1000
31   SAMPLE;
32   TestLoop(Size / 5);  // 200 * 1000
33   SAMPLE;
34   TestLoop(Size / 2);  // 500 * 1000
35   SAMPLE;
36   TestLoop(Size / 1);  // 1000* 1000
37   SAMPLE;
38   }
```

ПРИЛОЖЕНИЕ В

Полное время LAB1.CPP

```
#include <math.h>
#include <stdio.h>
#include "Sampler.h"

float bessy (float x, float n) {
      const  float small = 1.0E-8;
      const  float euler = 0.57721566;
      const  float pi    = 3.1415926;
      const  float pi2   = 0.63661977;

      int     j;
      float  x2,sum,sum2,t,t2,
             ts,term,xx,y0,y1,
             ya,yb,yc,ans,a,b,
             sina,cosa;

      if (x<12) {
             xx = 0.5 * x;
             x2 = xx * xx;
             t= log(xx) + euler;
             sum = 0.0;
             term = t;
             y0 = t;
             j = 0;
             do{
                    j=j+1;
                    if (j != 1) {sum = sum + 1 / (j - 1);}
                    ts = t - sum;
                    term = -x2 * term / (j * j)*(1 - 1/(j * ts));
                    y0 = y0 + term;
             }while ( fabs(term) >= small);
             term = xx * (t - 0.5);
             sum = 0.0;
             y1 = term;
             j = 1;
             do{
                    j = j + 1;
                    sum = sum + 1.0 / (j - 1);
                    ts = t - sum;
                    term = (-x2 * term)/(j * (j - 1))*((ts - 0.5/j)/(ts +
   0.5/(j - 1)));
                    y1 = y1 + term;
             }while (fabs(term) >= small);
             y0 = pi2 * y0;
             y1 = pi2 * (y1 - 1/x);
             if (n==0.0) {ans = y0;}
             else {
                    if (n==1.0) {ans = y1;}
                    else
                    {
                           ts = 2.0 / x;
                           ya = y0;
                           yb = y1;
```

```
                        for (j=2;  j<ceil(n+0.01);j+1)
                        {
                              yc = ts * (j - 1) * yb - ya;
                              ya = yb;
                              yb = yc;
                        }
                        ans = yc;
                  }
            }
            return ans;
      }
      else    return       sqrt(2 /(pi * x)) * sin(x - pi/4 - n * pi/2);
}

void main (void)
{
      float   x;
      float   ordr;
      int     done;
      float  ans;
      done = 0;
      ordr = 1;
      do{
            if (ordr<0.0) {done = 1;}
            else
            {
                  do{
                        x = 5;
                  }while (x < 0.0);
                  SAMPLE;
                  ans = bessy(x,ordr);
                  SAMPLE;
                  ordr = -1;
            }
      }while (done == 0)
}
```

ПРИЛОЖЕНИЕ Г
Время ФУ LAB1.CPP

```
1. #include <math.h>
2. #include <stdio.h>
3. #include "Sampler.h"
4.
5. float bessy (float x, float n) {
6.       const  float small = 1.0E-8;
7.       const  float euler = 0.57721566;
8.       const  float pi    = 3.1415926;
9.       const  float pi2   = 0.63661977;
10.
11.      int     j;
12.      float  x2,sum,sum2,t,t2,
13.             ts,term,xx,y0,y1,
14.             ya,yb,yc,ans,a,b,
15.             sina,cosa;
16.
17.      SAMPLE;
18.      if (x<12) {
19.             SAMPLE;
20.             xx = 0.5 * x;
21.             x2 = xx * xx;
22.             t= log(xx) + euler;
23.             sum = 0.0;
24.             term = t;
25.             y0 = t;
26.             j = 0;
27.             SAMPLE;
28.             do{
29.                    SAMPLE;
30.                    j=j+1;
31.                    SAMPLE;
32.                    if (j != 1) {
33.                           SAMPLE;
34.                           sum = sum + 1 / (j - 1);
35.                           SAMPLE;
36.                    }
37.                    SAMPLE;
38.                    ts = t - sum;
39.                    term = -x2 * term / (j * j)*(1 - 1/(j * ts));
40.                    y0 = y0 + term;
41.                    SAMPLE;
42.             }while ( fabs(term) >= small);
43.             SAMPLE;
44.             term = xx * (t - 0.5);
45.             sum = 0.0;
46.             y1 = term;
47.             j = 1;
48.             SAMPLE;
49.             do{
50.                    SAMPLE;
51.                    j = j + 1;
52.                    sum = sum + 1.0 / (j - 1);
```

```
                  ts = t - sum;
                  term = (-x2 * term)/(j * (j - 1))*((ts - 0.5/j)/(ts +
   0.5/(j - 1)));
                  y1 = y1 + term;
            SAMPLE;
            }while (fabs(term) >= small);
            SAMPLE;
            y0 = pi2 * y0;
            y1 = pi2 * (y1 - 1/x);
            SAMPLE;
            if (n==0.0) {
                  SAMPLE;
                  ans = y0;
                  SAMPLE;
            }
            else {
                  SAMPLE;
                  if (n==1.0) {
                        SAMPLE;
                        ans = y1;
                        SAMPLE;
                  }
                  else
                  {
                        SAMPLE;
                        ts = 2.0 / x;
                        ya = y0;
                        yb = y1;
                        SAMPLE;
                        for (j=2;  j<ceil(n+0.01);j+1)
                        {
                              SAMPLE;
                              yc = ts * (j - 1) * yb - ya;
                              ya = yb;
                              yb = yc;
                              SAMPLE;
                        }
                        SAMPLE;
                        ans = yc;
                  }
                  SAMPLE;
            }
            SAMPLE;
            return ans;
      }
      else   {
               SAMPLE;
               return      sqrt(2 /(pi * x)) * sin(x - pi/4 - n * pi/2);
              }
     }

     void main (void)
     {
            float   x;
            float   ordr;
            int     done;
            float  ans;
```

```
109.            done = 0;
110.            ordr = 1;
111.            do{
112.                  if (ordr<0.0) {done = 1;}
113.                  else
114.                  {
115.                        do{
116.                              x = 5;
117.                        }while (x < 0.0);
118.                        ans = bessy(x,ordr);
119.                        SAMPLE;
120.                        ordr = -1;
121.                  }
122.            }while (done == 0)
123.      }
```

ПРИЛОЖЕНИЕ Д

Полное время измененной LAB1.C

```
1. #include <math.h>
2. #include <stdio.h>
3. #include "Sampler.h"
4.
5. Float ordr = 1, x, ans1;
6. Int done = 0;
7.
8. float bessy () {
9.       const  float small  = 1.0E-8;
10.      const  float euler  = 0.57721566;
11.      const  float pi     = 3.1415926;
12.      const  float pi2    = 0.63661977;
13.
14.      int     j;
15.      float  x2,sum,sum2,t,t2,
16.             ts,term,xx,y0,y1,
17.             ya,yb,yc,ans,a,b,
18.             sina,cosa;
19.
20.      if (x<12) {
21.             xx = 0.5 * x;
22.             x2 = xx * xx;
23.             t= log(xx) + euler;
24.             sum = 0.0;
25.             term = t;
26.             y0 = t;
27.             j = 0;
28.             do{
29.                    J+=1;
30.                    if (j != 1) {sum = sum + 1 / (j - 1);}
31.                    ts = t - sum;
32.                    term = -x2 * term / (j * j)*(1 - 1/(j * ts));
33.                    y0 += + term;
34.             }while ( fabs(term) >= small);
35.             term = xx * (t - 0.5);
36.             sum = 0.0;
37.             y1 = term;
38.             j = 1;
39.             do{
40.                    j += 1;
41.                    sum += 1.0 / (j - 1);
42.                    ts = t - sum;
43.                    term = (-x2 * term)/(j * (j - 1))*((ts - 0.5/j)/(ts +
   0.5/(j - 1)));
44.                    y1 += term;
45.             }while (fabs(term) >= small);
46.             y0 = pi2 * y0;
47.             y1 = pi2 * (y1 - 1/x);
```

```
            if (ordr ==0.0) {ans = y0;}
            else {
                  if (ordr ==1.0) {ans = y1;}
                  else
                  {
                        ts = 2.0 / x;
                        ya = y0;
                        yb = y1;
                        for (j=2;  j<ceil(ordr +0.01);j+1)
                        {
                              yc = ts * (j - 1) * yb - ya;
                              ya = yb;
                              yb = yc;
                        }
                        ans = yc;
                  }
            }
            return ans;
      }
      else    return       sqrt(2 /(pi * x)) * sin(x - pi/4 - ordr * pi/2);
}

void main (void)
{
      do{
            if (ordr<0.0) {done = 1;}
            else
            {
                  do{
                        x = 5;
                  }while (x < 0.0);
                  SAMPLE;
                  ans1 = bessy();
                  SAMPLE;
                  ordr = -1;
            }
      }while (done == 0)
}
```

ПРИЛОЖЕНИЕ Е

Время ФУ измененной LAB1.C

```
#include <math.h>
#include <stdio.h>
#include "Sampler.h"

float   ordr = 1, x, ans1;
int     done = 0;

float bessy () {
        const  float small      = 1.0E-8;
        const  float euler      = 0.57721566;
        const  float pi    = 3.1415926;
        const  float pi2  = 0.63661977;

        int     j;
        float     x2,sum,sum2,t,t2,
                  ts,term,xx,y0,y1,
                  ya,yb,yc,ans,a,b,
                  sina,cosa;

        SAMPLE;
        if (x<12) {
                SAMPLE;
                xx = 0.5 * x;
                x2 = xx * xx;
                t= log(xx) + euler;
                sum = 0.0;
                term = t;
                y0 = t;
                j = 0;
                SAMPLE;
                do{
                        SAMPLE;
                        j+=1;
                        SAMPLE;
                        if (j != 1) {
                                SAMPLE;
                                sum = sum + 1 / (j - 1);
                                SAMPLE;
                        }
                        SAMPLE;
                        ts = t - sum;
                        term = -x2 * term / (j * j)*(1 - 1/(j * ts));
                        y0 += term;
                        SAMPLE;
```

```
45.             }while ( fabs(term) >= small);
46.             SAMPLE;
47.             term = xx * (t - 0.5);
48.             sum = 0.0;
49.             y1 = term;
50.             j = 1;
51.             SAMPLE;
52.             do{
53.                     SAMPLE;
54.                     j = j + 1;
55.                     sum = sum + 1.0 / (j - 1);
56.                     ts = t - sum;
57.                     term = (-x2 * term)/(j * (j - 1))*((ts - 0.5/j)/(ts + 0.5/(j - 1)));
58.                     y1 = y1 + term;
59.                     SAMPLE;
60.             }while (fabs(term) >= small);
61.             SAMPLE;
62.             y0 *= pi2;
63.             y1 = pi2 * (y1 - 1/x);
64.             SAMPLE;
65.             if (ordr==0.0) {
66.                     SAMPLE;
67.                     ans = y0;
68.                     SAMPLE;
69.             }
70.             else {
71.                     SAMPLE;
72.                     if (ordr==1.0) {
73.                             SAMPLE;
74.                             ans = y1;
75.                             SAMPLE;
76.                     }
77.                     else
78.                     {
79.                             SAMPLE;
80.                             ts = 2.0 / x;
81.                             ya = y0;
82.                             yb = y1;
83.                             SAMPLE;
84.                             for (j=2;  j<ceil(ordr+0.01);j+1)
85.                             {
86.                                     SAMPLE;
87.                                     yc = ts * (j - 1) * yb - ya;
88.                                     ya = yb;
89.                                     yb = yc;
90.                                     SAMPLE;
91.                             }
92.                             SAMPLE;
93.                             ans = yc;
```

```
94.                             }
95.                             SAMPLE;
96.                     }
97.                     SAMPLE;
98.                     return ans;
99.             }
100.                    else       return   sqrt(2 /(pi * x)) * sin(x - pi/4 - ordr * pi/2);
101.            }
102.
103.            void main (void)
104.            {
105.                    do{
106.                            if (ordr<0.0) {done = 1;}
107.                            else
108.                            {
109.                                    do{
110.                                            x = 5;
111.                                    }while (x < 0.0);
112.                                    ans1 = bessy();
113.                                    ordr = -1;
114.                            }
115.                    }while (done == 0);
116.            }
```