Национальный Исследовательский Университет «МЭИ»

Институт радиотехники и электроники им. В.А. Котельникова Кафедра РТС

Курсовая работа

По дисциплине

«Аппаратура потребителей СРНС»

«Разработка модуля расчета координат спутника Beidou»

Студент: Ряшенцева В.И.

Группа: Эр-15-16

Преподаватель: Корогодин И.В.

Исходные данные

Спутник №19 системы Beidou.

Этап 1. Использование сторонних средств

Заданному в задании спутнику 19 соответствует спутник со следующим номером:

Таблица 1

Спутник	PRN	ID	SCN
BEIDOU 3-M1	C19	2017-069A	43001

При помощи сервиса CelesTrak можно получить изображение формы орбиты и положение спутника на ней. Изображения по требованию в задании необходимо получить на период 18:00 МСК 16 февраля 2021.

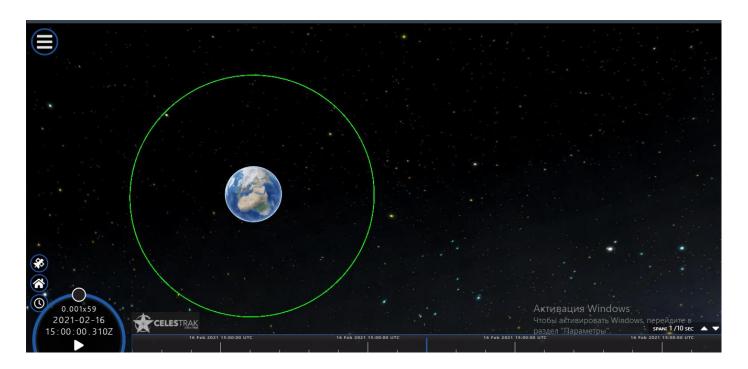


Рисунок 1 — Форма орбиты спутника

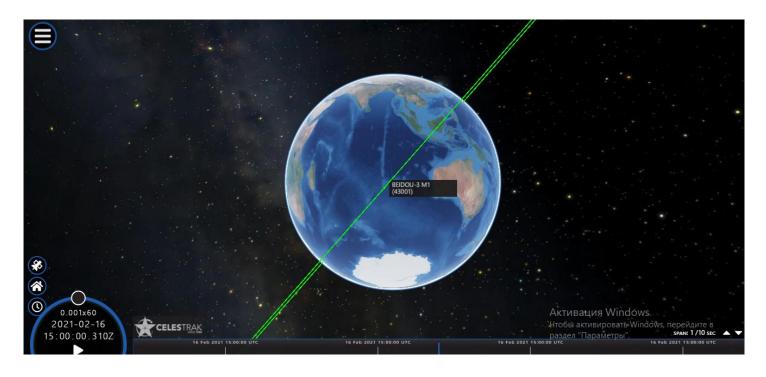


Рисунок 2 — Положение спутника на орбите

По данным Trimble GNSS Planning Online необходимо рассчитать диаграмму угла места и азимута спутника на интервал времени с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года. Расчет производится с учетом координат, соответствующих корпусу Е МЭИ.

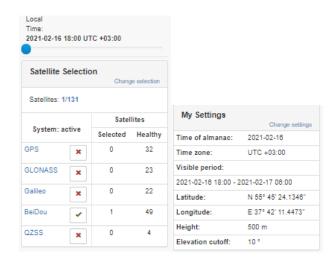


Рисунок 3 — Hастройки Trimble GNSS Planning Online

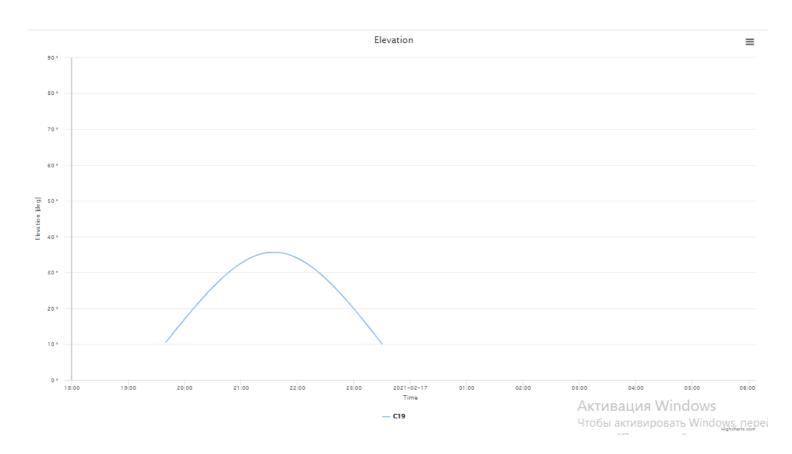


Рисунок 3 — Диаграмма угла места спутника

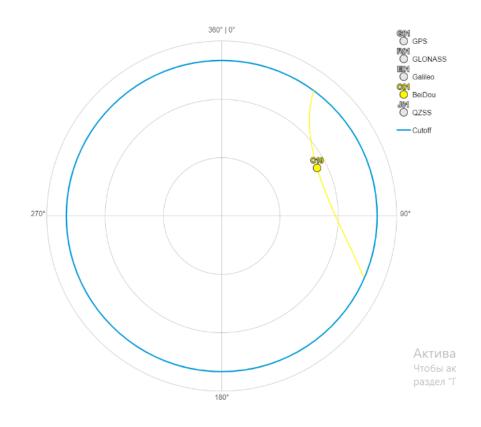


Рисунок 4 — Диаграмма угла азимута спутника

Ниже, на рисунке 5, приведены эфемериды спутника.

```
1 SatNum 19
2 toe (ms) 284400000.000
3 Crs (m) -5.939062500000000000e+01
4 Dn (rad/ms) 4.05159716540537396e-12
5 M0 (rad) 2.09961820806318311e+00
6 Cuc (rad) -2.80654057860374451e-06
7 e 8.22309288196265697e-04
8 Cus (rad) 5.74858859181404114e-06
9 sqrtA (sqrt (m)) 5.28262378883361816e+03
10 Cic (rad) 6.42612576484680176e-08
11 Omega0 (rad) -2.83355370614262747e-01
12 Cis (rad) -7.59027898311614990e-08
13 i0 (rad) 9.65653770875602335e-01
14 Crc (m) 2.438437500000000000e+02
15 omega (rad) -1.24505537862834337e+00
16 OmegaDot (rad/ms) -7.01279211094012322e-12
17 iDot (rad/ms) -2.57867884089573192e-13
18 Tgd (ms) 1.2200000000000000e+05
19 toc (ms) 2.84400000000000000e+05
19 toc (ms) 2.84400000000000000e+08
20 af2 (ms/ms^2) 0.000000000000000e+08
21 af1 (ms/ms) 1.37863054305853439e-11
22 af0 (ms) 7.30971693992614746e-01
23 URA 0
24 IODE 257
25 IODC 1
26 codeL2 0
27 L2P 0
28 WN 789
```

Рисунок 5 — Эфемериды спутника

Этап 2. Моделирование

На данном этапе работы необходимо реализовать на языке Matlab функцию расчета положения спутника № 19 СРНС Beidou на заданный момент по шкале времени UTC. В качестве эфемерид будут использоваться данные, полученные на предыдущем этапе.

Важно уточнить, что так как система Beidou унаследовала модель GPS, то расчеты будут производиться соответственно алгоритму из ИКД GPS [1].

Для начала необходимо разобраться с исходными данными. Как было сказано выше, эфемериды будут взяты с этапа 1 данной работы, но кроме этих данных, для расчета интервала прогноза необходимо учесть формат времени системы Beidou и рассчитать количество секунд от начала текущей недели.

Для этого запишем формат времени Beidou:

$$WN:SOW$$
,

где *WN* — номер недели начиная с 1 января 2006 г., *SOW* — количество секунд от начала текущей недели.

Таким образом, необходимо рассчитать разницу между временами 16/02/2021 15:00:00 и 01/01/2006 00:00:00:

$$2021 - 2006 = 15 лет$$

Из них 4 года является високосными (2008, 2012, 2016 и 2020) + 47 дней (32+15=47) + 15 часов 0 сек

– Временная разница в полных днях:

$$365 \cdot (15-4) + 366 \cdot 4 + 47 = 5526$$
 дней

– Временная разница в полных неделях:

$$\frac{5526}{7}$$
 = 789 недель

- Количество дней после полных недель:

$$5526 - 789 \cdot 7 = 3$$
 дня

– Количество секунд от начала текущей недели:

$$3 \cdot 24 \cdot 3600 + 15 \cdot 3600 = 313200$$
 cek

Количество секунд от начала текущей недели с учетом добавленных секунд[2]:

$$313200+4=313204$$
 cek

Ответ: 789:313204.

С учетом полученных данных и прогноза с 18:00 МСК 16 февраля до 06:00 МСК 17 февраля 2021 года (то есть временем прогноза, равным 43200 сек) построены трехмерные графики множества положений спутника Beidou в двух вариантах: в СК ЕСЕГ WGS84 (рисунок 6) и соответствующей ей инерциальной СК ЕСІ (рисунок 7).

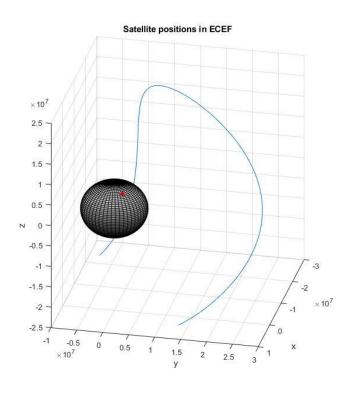


Рисунок 6 — Множество положений спутника Beidou в системе ECEF

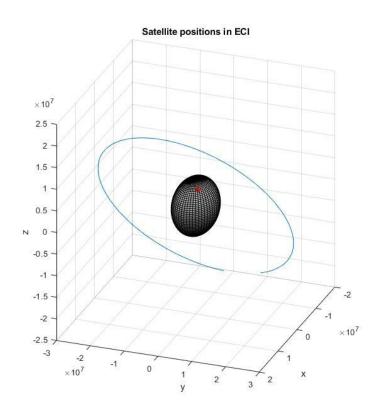


Рисунок 7 — Множество положений спутника Beidou в системе ECI

На рисунках 6 и 7 помимо траектории спутника так же изображена модель Земли с нанесенной на нее координатой антенны на крыше корпуса Е НИУ «МЭИ».

Переход из системы ECEF в систему ECI был осуществлен также согласно алгоритму из ИКД.

По полученным графикам видно, что за установленный интервал времени спутник не успевает полностью пройти всю свою траекторию. Экспериментально было установлено, что для этого необходимо задаться приблизительно в два раза большим интервалом.

Помимо траектории спутников в трехмерном виде было интересно получить эту траекторию в полярной системе координат и сравнить ее с результатом из Trimble GNSS Planning Online, изображенным на рисунке 4.

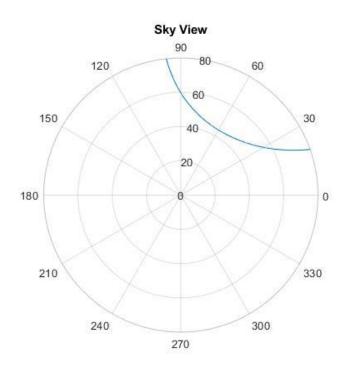


Рисунок 8 — Множество положений спутника Beidou в полярной системе координат

При сравнении результатов, изображенных на рисунках 4 и 8 важно учесть, что из-за особенностей алгоритмов Matlab, график в полярной системе координат строиться с установкой градусов против часовой стрелки и нулевым значением справа. Учитывая эти факторы видно, что результаты совпадают с небольшими расхождениями, которые могут быть связаны с тем, что при моделировании использовались одни эфемериды и то, что обрезка системы координат в Trimble GNSS Planning Online и Matlab могут отличаться.

Код программы в Matlab представлен в приложении.

Этап 3. Реализация

На данном этапе работы необходимо было разработать на языке C/C++ функцию расчета положения спутника Beidou на заданное время по шкале UTC, минимизируя время её исполнения и количество затрачиваемой оперативной памяти.

В качестве языка программирования был выбран язык C++, написание программы производилось в Visual studio 2019 (VS) (код в приложении).

Алгоритм расчета местоположения спутника включает в себя уравнение Кеплера. По условию задания необходимо было протестировать ее.

Решение уравнения Кеплера было выведено в отдельный программный модуль, для реализации теста решения данного уравнения была подключена OpenSource библиотека для Unit-тестирования Simple C++ testing library [3].

Для проверки корректной работы теста, были переприсвоены такие параметры как эксцентриситет и средняя аномалия (для получения заведомо ложных результатов). Результат работы теста показан на рисунке 9.

Рисунок 9 — Результат теста работы функции решения уравнения Кеплера

Далее необходимо было сравнить расчетного положения спутника в сравнении с Matlab с шагом 1 с. Данный этап был реализован путем сравнения массивов данными (результат работы кода, описанный в этапе 2 данного отчета) Matlab и C++. Для уменьшения количества открытых файлов в C++,

было принято решения сравнивать не координаты x, y, z, a дальность: $R = \sqrt{x^2 + y^2 + z^2} \ .$

В ходе сравнения результирующих массивов Matlab и C++ была найден максимальное расхождение в значениях, которое составляет $5,06639 \cdot 10^{-7}$ [м].

Полученная разница обусловлена тем, что тип данных в Matlab и C++ имеют разную точность (количество байт, хранимых в памяти).

Для проверки наличия утечки памяти были использованы внутренние инструменты VS (подключение хедера crtdbg.h) в режиме debug.

С помощью функции _CrtMemCheckpoint() был сделан снепшот утилизации памяти в момент запуска программы и после её выполнения. После чего оба снепшета сравниваются посредством вызова функции _CrtMemDifference().

Для того, чтобы убедиться, что вся память, выделенная приложением, была освобождена и не произошла утечка памяти с момента запуска программы, была вызвана функция _CrtDumpMemoryLeaks(). Данные по утечке памяти были выедены в окно отладки VS, как показаны на рисунке 10.

Рисунок 10 — Окно debug; leak memory

По заданию необходимо было выполнить профилирование программы С++. Для этого были задействованы внутренний модуль отладки VS — профилировщик производительности. В ходе профилирования были получены сведения по утилизации СРU, а также данные по затрачиваемому времени работы функций каждого модуля. На графике, представленном на рисунке 11, отображена утилизация в ходе выполнения программы, общее время выполнения программы, а также критический путь — список объектов с наибольшим затрачиваемым временем на работу.



Рисунок 11 — Отчет профилирования программы С++

В ходе профилирования было выявлено общее время работы программы, равное 2 [mc] и затрачиваемое время на работу каждой отдельной функции (в приложении).

Таким образом, в ходе выполнения данного этапа была написана программа на языке C++, что целесообразно при разработке встраиваемого ПО. Кроме того, в процессе разработке производилось тестирование уравнения Кеплера, что позволяет быстро обнаружить ошибку при не верных исходных данных. Также по заданию необходимо было произвести сравнение

результатов работы программы, на писанных на языках Matlab и C++, в ходе которого было установлено, что данные языки имеют различия в типах данных (в памяти хранится разное количество байт). Более того, интересно было протестировать программу на наличие утечки памяти, так как при ее наличии может произойти переполнении всей оперативной памяти устройства и т.п. В ходе данного тестирования утечка памяти не была обнаружена.

Заключение

В ходе выполнения данного курсового проекта было произведено ознакомление и изучения ИКД Beidou [1].

При помощи сторонних средств была построена орбита заданного в задании спутника Beidou, получена траектория его движения на заданный промежуток времени.

При помощи алгоритма, предоставленного в ИКД, были изучены все этапы нахождения местоположения спутника. Далее этот алгоритм был реализован на языке Matlab.

Заключительным этапом выполнения курсового проекта является факт разработки функции расчета местоположения спутника Beidou на языке C++. В ходе выполнения данного этапа было произведено сравнение результатов работы программы на языках Matlab и C++; обнаружен факт различия размера хранимых байт в памяти. Разработанный алгоритм протестирован на верность полученных результатов и отсутствие утечки памяти.

Список источников

- **1.** Interface Specification IS-GPS-200L, August 2020. [Электронный ресурс]. URL: https://www.gps.gov/technical/icwg/IS-GPS-200L.pdf (дата обращения: 03.04.2021).
- **2.** Дополнительная секунда [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Дополнительная_секунда (дата обращения: 03.04.2021).
- **3.** Библиотека для Unit-тестирования C++ [Электронный ресурс]. URL: https://github.com/mitya57/cxxunit (дата обращения: 05.05.2021).

Приложение

```
clc
close all
%% ephemeris
e = 8.22309288*10^{(-4)}; % eccentricity
mu = 3.986005*10^14; % WGS 84 value of the earth's gravitational constant
Omega e = 7.2921151467*10^{(-5)}; % WGS 84 value of the earth's rotatuon rate
toe = 284400 + 4; % time of ephemeris
A = (5.28262379*10^3)^2; % semi-major axis ^2
n0 = sqrt(mu/A^3); % computed mean motion
i0 = 0.96565377; % inclination
Om01 = 0.28335371; % longitude of the node
omega = -1.24505538; % perigee argument
M0 = 2.09961821; % mean anomaly
Dn = 4.05159717*10^{-9}; % mean motion difference from computed value
n = n0 + Dn; % corrected mean motion
OmegaDot = -7.0127921109*10^{(-9)}; % rate of right ascension
IDOT = -2.57867884*10^{(-10)}; % rate of inclination angle
Crs = -59.390625; % amplitude of the sine harmonic correction term to the
orbit radius
Cuc = -2.80654058*10^{(-6)}; % amplitude of the cosine harmonic correction term
to the argument of latitude
Cus = 5.74858859*10^{(-6)}; % amplitude of the sine harmonic correction term to
the argument of latitude
Cic = 6.42612576*10^{(-8)}; % amplitude of the cosine harmonic correction term
to the angle of inclination
Cis = -7.59027898*10^{(-8)}; % amplitude of the sine harmonic correction term
to the angle of inclination
Crc = 243.84375; % amplitude of the cosine harmonic correction term to the
orbit radius
응응
for i = 1:43200
t = 313200 + 4 + i; % TOW
tk = t - toe; % time from ephemeris reference epoch % if tk is greater than
302400 seconds, subtract 604800 sec from tk.
% if tk is less than -302400 sec, add 604800 seconds to tk
Mk = M0 + n*tk; % mean anomaly
%% Kepler's equation may be solved by iteration
E = zeros(1,4);
E(1,1) = Mk;
       for j = 2:4
              E(1,j) = E(1,j-1) + (Mk - E(1,j-1) + e*sin(E(1,j-1)))/(1-
e*cos(E(1,j-1))); % 3 iterations
              E(1,j-1) = E(1,j);
       end
E k = E(1,4);
nu k = atan2((sqrt(1 - e^2)*sin(E k)))/(1 - e*cos(E k)),(cos(E k) - e)/(1 - e^2)/(1 
e*cos(E k))); % true anomaly
Phi k = nu k + omega; % argument of latitude
delta u k = Cus*sin(2*Phi k) + Cuc*cos(2*Phi k); % argument of latitude
correction
delta r k = Crs*sin(2*Phi k) + Crc*cos(2*Phi k); % radius correction
delta i k = Cis*sin(2*Phi k) + Cic*cos(2*Phi k); % inclination correction
u k = Phi k + delta u k; % corrected argument of latitude
r k = A*(1 - e*cos(E k)) + delta r k; % corrected radius
i k = i0 + delta i k + IDOT*tk; %corrected inclination
```

```
%% positions in orbital
x_k = r_k \cos(u_k);
y_k = r_k * sin(u_k);
Omega k = Om01 + (OmegaDot - Omega e)*tk - Omega e*toe; % corrected longitude
of ascending node
%% earth-fixed
x k1 = x k*cos(Omega k) - y k*cos(i k)*sin(Omega k);
y k1 = x k*sin(Omega k) + y k*cos(i k)*cos(Omega k);
z_k1 = y_k*sin(i k);
x1 coord(1,i) = x_k1;
y1 \operatorname{coord}(1,i) = y k1;
z1\_coord(1,i) = z_k1;
%% transfer
tetta = Omega_e*tk;
x k2 = x k1*cos(tetta) - y_k1*sin(tetta);
y k2 = x k1*sin(tetta) + y k1*cos(tetta);
z k2 = z k1;
x2 \operatorname{coord}(1,i) = x k2;
y2 \operatorname{coord}(1,i) = y k2;
z2 \operatorname{coord}(1,i) = z k2;
%% porar coordinates
Earth radius = 6378136;
H = 500;
a = Earth radius;
B = deg2rad(55.45241346);
N = a/sqrt((1-e^2*(sin(B))^2));
L = deg2rad(37.42114473);
Coord x = (N+H) * cos(B) * cos(L);
Coord y = (N+H) * cos(B) * sin(L);
Coord z = ((1-e^2) *N+H) *sin(B);
p = sqrt((x k1-Coord x).^2+(y k1-Coord y).^2+(z k1-Coord z).^2);
tetta1 = asin((z k1-Coord z)/p);
phi1 = atan2((x k1-Coord x), (y k1-Coord y));
tetta1 i(1,i) = -tetta1*180/pi + 90;
phi1_i(1,i) = -phi1;
end
%% plot
% Earth with coord
[x sphere, y sphere, z sphere] = sphere(50);
x Earth=Earth radius*x sphere;
y Earth=Earth radius*y sphere;
z Earth=Earth radius*z sphere;
figure (1)
subplot (1,2,1)
surf(x Earth, y Earth, z Earth); hold on;
colormap ('gray');
plot3(Coord_x, Coord_y, Coord_z, 'r.','MarkerSize', 20 );
plot3(x1 coord(1,:), y1 coord(1,:), z1 coord(1,:));
xlabel('x');
ylabel('y');
zlabel('z');
title('Satellite positions in ECEF')
figure (2)
subplot (1,2,1)
```

```
surf(x_Earth,y_Earth,z_Earth); hold on;
colormap ('gray');
plot3(Coord_x, Coord_y, Coord_z, 'r.', 'MarkerSize', 20 );
plot3(x2_coord(1,:), y2_coord(1,:), z2_coord(1,:));
xlabel('x');
ylabel('y');
zlabel('z');
title('Satellite positions in ECI')

figure(3)
grid on;
polarplot(phi1_i(1,:), tetta1_i(1,:));
rlim([0 80]);
title('Sky View')
```

beidou2.0.cpp

```
#include <iostream>
#include <cmath>
#include "windows.h"
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtdbg.h>
using namespace std;
long double kepler(long double e, long double Mk);
long double testing(long double e, long double Mk);
int main()
{
       CrtMemState sOld;
       _CrtMemState sNew;
      _CrtMemState sDiff;
      _CrtMemCheckpoint(&sOld); //take a snapchot
      long double e, mu, Omega_e, toe, A, n0, i0, Om01, omega, M0, Dn, n, OmegaDot,
IDOT, Crs, Cuc, Cus, Cic, Cis, Crc;
      long double t, tk, Mk, E_k, nu_k, Phi_k, delta_u_k, delta_r_k, delta_i_k, u_k,
r_k, i_k;
      long double x_k, y_k, Omega_k, x_k1, y_k1, z_k1;
      e = 8.22309288E-4;
      mu = 3.986005E+14;
      Omega_e = 7.2921151467E-5;
      toe = 284404;
      A = pow(5.28262379E+3, 2);
      n0 = pow(mu / (pow(A, 3)), 0.5);
      i0 = 0.96565377;
      Om01 = 0.28335371;
      omega = -1.24505538;
      M0 = 2.09961821;
      Dn = 4.05159717E-9;
      n = n0 + Dn;
      OmegaDot = -7.0127921109E-9;
      IDOT = -2.57867884E-10;
      Crs = -59.390625;
      Cuc = -2.80654058E-6;
      Cus = 5.74858859E-6;
      Cic = 6.42612576E-8;
      Cis = -7.59027898E-8;
      Crc = 243.84375;
      t = 313204; // TOW
      tk = t - toe; // time from ephemeris reference epoch
      Mk = M0 + n * tk; // mean anomaly
      // Kepler's equation may be solved by iteration
      testing(e,Mk);
      E_k = kepler(e, Mk);
      nu k = atan2((sqrt(1 - pow(e, 2)) * sin(E k)) / (1 - e * cos(E k)), (cos(E k) - e)
/ (1 - e * cos(E k))); // true anomaly
       Phi k = nu k + omega; // argument of latitude
      delta_u_k = Cus * sin(2 * Phi_k) + Cuc * cos(2 * Phi_k); // argument of latitude
correction
      delta_r_k = Crs * sin(2 * Phi_k) + Crc * cos(2 * Phi_k); // radius correction
```

```
delta_i_k = Cis * sin(2 * Phi_k) + Cic * cos(2 * Phi_k); // inclination correction
       u_k = Phi_k + delta_u_k; // corrected argument of latitude
       r_k = A * (1 - e * cos(E_k)) + delta_r_k; // corrected radius
       i_k = i0 + delta_i_k + IDOT * tk; // corrected inclination
       // positions in orbital
       x_k = r_k * cos(u_k);
       y_k = r_k * sin(u_k);
       Omega k = Om01 + (OmegaDot - Omega e) * tk - Omega e * toe; // corrected longitude
of ascending node
       // earth-fixed
       x_k1 = x_k * cos(Omega_k) - y_k * cos(i_k) * sin(Omega_k);

y_k1 = x_k * sin(Omega_k) + y_k * cos(i_k) * cos(Omega_k);
       z_k1 = y_k * sin(i_k);
       //cout << E_k << endl;
      cout << "x=" << x_k1 << endl;
cout << "y=" << y_k1 << endl;
cout << "z=" << z_k1 << endl;
       _CrtMemCheckpoint(&sNew); //take a snapchot
       _CrtMemDifference(&sDiff, &sOld, &sNew);
       OutputDebugString(L"-----'CrtMemDumpStatistics -----');
_CrtMemDumpStatistics(&sDiff);
       OutputDebugString(L"-----");
       _CrtMemDumpAllObjectsSince(&sOld);
       OutputDebugString(L"-----");
       if ( CrtDumpMemoryLeaks() == false)
       {
              OutputDebugString(L"There is no Memory Leaks");
       }
       return 0;
}
```

kepler.cpp

```
#include <math.h>

long double kepler(long double e, long double Mk)
{
      //long double E_k;
      long double E[] = { 0, 0, 0, 0 };
      E[0] = Mk;
      for (int j = 1; j < 4; j++)
      {
            E[j] = E[j - 1] + (Mk - E[j - 1] + e * sin(E[j - 1])) / (1 - e * cos(E[j - 1])); // 3 iterations
            E[j - 1] = E[j];
      }

    return E[3];
}</pre>
```

test_kepler.cpp

```
#include "testing.h"
//#include "Unit.h"
long double kepler(long double e, long double Mk);
struct test_kepler_function : TestCase {
      void run(long double e, long double Mk) override {
             /*void test(double e, double Mk)
             {*/
             double TrueRes = kepler(8.22309288E-4, 6.00016);
       ASSERT ALMOST EQUAL(kepler(e, Mk), TrueRes);
      }
};
REGISTER_TEST(test_kepler_function, "Kepler testing");
long double testing(long double e, long double Mk) {
    bool failfast = false;
    bool nocatch = false;
    int result = 0;
    for (const TestCaseInfo& info : storage) {
        std::cout << " * " << info.name << std::endl;</pre>
        TestCase* test_case = info.test_case;
        test_case->failfast = failfast;
        bool success = true;
        if (nocatch) {
           test_case->run(e,Mk);
        else {
           try {
               test_case->run(e,Mk);
            catch (const std::exception& e) {
               << std::endl;
        if (test_case->assertions_successful < test_case->assertions_total) {
           success = false;
        if (success) {
            std::cout << " Result: " << 0_SUCCESS("SUCCESS")</pre>
               << " (" << test_case->assertions_total << " assertions passed)"</pre>
                << std::endl;</pre>
        }
        else {
            std::cout << " Result: " << 0_ERROR("FAIL")</pre>
                << " (" << test_case->assertions_successful << " of "</pre>
                << test_case->assertions_total << " assertions passed)"</pre>
                << std::endl;</pre>
            result = 1;
        delete test case;
    return result;
}
```

testing.h

```
/* Simple C++ testing library.
 * Copyright: 2014-2017, Dmitry Shachnev.
 * Permission is hereby granted, free of charge, to any person obtaining
 ^{st} a copy of this software and associated documentation files (the
 * "Software"), to deal in the Software without restriction, including
 * without limitation the rights to use, copy, modify, merge, publish,
 * distribute, sublicense, and/or sell copies of the Software, and to
 * permit persons to whom the Software is furnished to do so, subject to
 * the following conditions:
 * The above copyright notice and this permission notice shall be included
 * in all copies or substantial portions of the Software.
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
 * CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
 * TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
 * SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
#ifndef CXXUNIT TESTING HPP
#define CXXUNIT TESTING HPP
#include <cfenv>
#include <cmath>
#include <cstdlib>
#include <cstring>
#include <algorithm>
#include <exception>
#include <iostream>
#include <string>
#include <vector>
#include "printing.h"
#ifdef __linux__
# include <csignal>
# include <execinfo.h>
#endif
#ifdef _MSC_VER
# define __PRETTY_FUNCTION__ __FUNCTION__
#define REGISTER_TEST(cls, name) \
  static TestCaseProcessor _##cls({new cls(), name})
#define PRINT ERROR() \
  std::cerr << " " << E_ERROR("ERROR") << " in `" << __PRETTY_FUNCTION__ << "' at " <<
__FILE__ \
            << " line " << __LINE__ << ":" << std::endl;
#define ASSERT TRUE(expression) \
  do { \
    auto _value = (expression); \
    if (!do assert( value)) { \
     PRINT_ERROR(); \
                         Expression `" #expression "' is not true." << std::endl; \</pre>
      std::cerr <<
      std::cerr << "
                         Value is " << repr(_value) << std::endl; \</pre>
      handle failfast(); \
```

```
} \
  } while (0)
#define ASSERT FALSE(expression) \
 do { \
    auto _value = (expression); \
    if (!do_assert(!_value)) { \
      PRINT_ERROR(); \
                         Expression `" #expression "' is not false." << std::endl; \</pre>
      std::cerr <<
      std::cerr << "
                         Value is " << repr( value) << std::endl; \</pre>
     handle failfast(); \
    } \
  } while (0)
#define ASSERT_RELATION(e1, rel, e2) \
 do { \
   auto _v1 = (e1); \
auto _v2 = (e2); \
    if (!do_assert(_v1 rel _v2)) { \
      PRINT_ERROR(); \
                         Expression `" #e1 " " #rel " " #e2 "' is not true." <</pre>
      std::cerr <<
std::endl; \
      std::cerr << " " << #e1 " = " << repr(_v1) << ", " #e2 " = " << repr(_v2) <<
std::endl; \
      handle_failfast(); \
    } \
  } while (0)
#define ASSERT_EQUAL(e1, e2) \
 do { \
    auto _v1 = (e1); \
    auto _v2 = (e2); \
    if (!do_assert(_v1 == _v2)) { \
      PRINT_ERROR(); \
      std::cerr << "
                       Expressions `" #e1 "' and `" #e2 "' are not equal." <<
std::endl; \
      std::cerr << " " << #e1 " = " << repr(_v1) << ", " #e2 " = " << repr(_v2) <<
std::endl; \
     handle failfast(); \
    } \
  } while (0)
#define ASSERT_ALMOST_EQUAL(e1, e2) \
  do { \
    long double precision=0.00001; \
    auto _v1 = (e1); \
    auto _v2 = (e2); \
    if (!do_assert(std::fabs(_v1 - _v2) < precision)) { \</pre>
      PRINT_ERROR(); \
                         Expressions `" #e1 "' and `" #e2 "' are not equal." <<</pre>
      std::cerr <<
std::endl; \
      std::cerr << " " << #e1 " = " << repr(_v1) << ", " #e2 " = " << repr(_v2) <<
std::endl; \
     handle failfast(); \
    } \
  } while (0)
#define ASSERT_STRINGS_EQUAL(e1, e2) \
 do { \
    std::string _v1 = (e1); \
    std::string _v2 = (e2); \
    if (!do_assert(_v1 == _v2)) { \
     PRINT_ERROR(); \
                         Strings `" #e1 "' (1) and `" #e2 "' (2) are not equal." <<
     std::cerr << '
std::endl; \
                         (1): '" << _v1 << "'," << std::endl; \</pre>
      std::cerr << "
                          (2): '" << _v2 << "'" << std::endl; \
      std::cerr << "
     handle failfast(); \
    } \
```

```
} while (0)
#define ASSERT FLOATS EQUAL(e1, e2) \
 do { \
    auto _v1 = (e1); \
    auto _v2 = (e2); \
    if (!do_assert(compare_double(_v1, _v2))) { \
      PRINT_ERROR(); \
                         Floating point numbers `" #e1 "' and `" #e2 "' are not equal."
      std::cerr <<
<< std::endl; \</pre>
                      " << #e1 " = " << repr( v1) << ", " #e2 " = " << repr( v2) <<
      std::cerr << "
std::endl; \
      handle failfast(); \
  } while (0)
#define ASSERT_THROWS(exception_class, expression) \
    bool caught = false; \
    try { \
      (void)(expression); \
    } catch (const exception_class &_exc) { \
     caught = true; \
     break; \
    } \
    if (!do_assert(caught)) { \
     PRINT_ERROR(); \
std::cerr << "</pre>
                         Exception " << #exception_class << " not thrown." << std::endl;</pre>
\
     handle_failfast(); \
  } while(0)
template<typename T>
T repr(const T value) { return value; }
uint16_t repr(const uint8_t value) { return value; }
bool compare floats(float v1, float v2) {
    return (std::fabs(v1 - v2) * 100000.f <= std::min(fabs(v1), fabs(v2)));
}*/
bool compare_double(double v1, double v2) {
    return (std::fabs(v1 - v2) * 1000000000000. <= std::min(fabs(v1), fabs(v2)));
struct TestCase {
    unsigned assertions_total;
    unsigned assertions_successful;
    bool failfast;
    TestCase() :
        assertions_total(0),
        assertions_successful(0),
        failfast(false)
    {}
    bool do_assert(bool condition) {
        ++assertions_total;
        if (condition) {
            ++assertions_successful;
        return condition;
    }
    void handle failfast() {
```

```
if (failfast) {
            std::cerr << "Exiting immediately because failfast = true." << std::endl;</pre>
            exit(1);
        }
    }
    virtual void run(long double e, long double Mk) = 0;
    virtual ~TestCase() {}
};
struct TestCaseInfo {
    TestCase* test case;
    std::string name;
};
static std::vector<TestCaseInfo> storage;
struct TestCaseProcessor {
    TestCaseProcessor(TestCaseInfo info) {
        storage.push_back(info);
    }
};
#ifdef __linux_
void signal_handler(int signum) {
    std::cerr << E_ERROR("Signal occurred") << ": " << strsignal(signum) << std::endl;</pre>
    /* Print the backtrace */
    void* buffer[10];
    size_t size = backtrace(buffer, 10);
    backtrace_symbols_fd(buffer, size, STDERR_FILENO);
    /* Now call the default handler */
    signal(signum, SIG_DFL);
    exit(128 + signum);
}
#endif
void print help(const char* command name) {
    std::cout << "Usage: " << command_name << " [-f] [-n]" << std::endl;</pre>
    std::cout << std::endl;</pre>
    std::cout << " -f, --fail-fast: Exit after first failure" << std::endl;</pre>
    std::cout << " -n, --no-catch: Do not catch C++ exceptions (useful for debugging)"</pre>
<< std::endl;</pre>
/*int main(int argc, char** argv) {
    bool failfast = false;
    bool nocatch = false;
    for (int i = 1; i < argc; ++i) {
        char* arg = argv[i];
        if (!strcmp(arg, "--fail-fast") || !strcmp(arg, "-f")) {
            failfast = true;
        else if (!strcmp(arg, "--no-catch") || !strcmp(arg, "-n")) {
            nocatch = true;
        }
        else {
            print_help(argv[0]);
            return 0;
        }
    }
#ifdef USE GNU
    feenableexcept(FE DIVBYZERO | FE INVALID | FE OVERFLOW | FE UNDERFLOW);
```

```
#endif
#ifdef __linux_
    signal(SIGSEGV, signal_handler);
#endif
    int result = 0;
    for (const TestCaseInfo& info : storage) {
        std::cout << " * " << info.name << std::endl;</pre>
        TestCase* test case = info.test case;
        test case->failfast = failfast;
        bool success = true;
        if (nocatch) {
            test_case->run();
        }
        else {
            try {
                test_case->run();
             catch (const std::exception& e) {
                 success = false;
std::cerr << " " << E_ERROR("Exception occurred") << ": " << e.what()</pre>
<< std::endl;
        if (test_case->assertions_successful < test_case->assertions_total) {
            success = false;
        if (success) {
             std::cout << " Result: " << 0_SUCCESS("SUCCESS")</pre>
                 << " (" << test_case->assertions_total << " assertions passed)"</pre>
                 << std::endl;
        }
        else {
             std::cout << " Result: " << O_ERROR("FAIL")</pre>
                 << " (" << test_case->assertions_successful << " of "</pre>
                 << test_case->assertions_total << " assertions passed)"</pre>
                 << std::endl;
             result = 1;
        delete test_case;
    return result;
}
*/
#endif /* CXXUNIT_TESTING_HPP */
```

time.cpp

```
#include <iostream>
#include <cmath>
#include <fstream>
#include <stdlib.h>
using namespace std;
long double kepler(long double e, long double Mk);
//double testing(double e, double Mk);
int main()
{
      long double e, mu, Omega_e, toe, A, n0, i0, Om01, omega, M0, Dn, n, OmegaDot,
IDOT, Crs, Cuc, Cus, Cic, Cis, Crc;
      long double tk, Mk, E_k, nu_k, Phi_k, delta_u_k, delta_r_k, delta_i_k, u_k, r_k,
      long double x_k, y_k, Omega_k, x_k1, y_k1, z_k1;
      int t;
      e = 8.22309288E-4;
      mu = 3.986005E+14;
      Omega e = 7.2921151467E-5;
      toe = 284404;
      A = pow(5.28262379E+3, 2);
      n0 = pow(mu / (pow(A, 3)), 0.5);
      i0 = 0.96565377;
      Om01 = 0.28335371;
      omega = -1.24505538;
      M0 = 2.09961821;
      Dn = 4.05159717E-9;
      n = n0 + Dn;
      OmegaDot = -7.0127921109E-9;
      IDOT = -2.57867884E-10;
      Crs = -59.390625;
      Cuc = -2.80654058E-6;
      Cus = 5.74858859E-6;
      Cic = 6.42612576E-8;
      Cis = -7.59027898E-8;
      Crc = 243.84375;
      long double* x1_coord = new long double[43200];
      long double* y1_coord = new long double[43200];
      long double* z1_coord = new long double[43200];
      long double* R = new long double[43200];
       long double* Rmc = new long double[43200];
      long double* res = new long double[43200];
      for (int i = 0; i < 43200; i++)
      {
             t = 313204 + i; // TOW
             tk = t - toe; // time from ephemeris reference epoch
             Mk = M0 + n * tk; // mean anomaly
             // Kepler's equation may be solved by iteration
             //testing(e,Mk);
             E_k = kepler(e, Mk);
```

```
nu_k = atan2((sqrt(1 - pow(e, 2)) * sin(E_k)) / (1 - e * cos(E_k)),
(\cos(E_k) - e) / (1 - e * \cos(E_k))); // true anomaly
              Phi_k = nu_k + omega; // argument of latitude
              delta u k = Cus * sin(2 * Phi k) + Cuc * cos(2 * Phi k); // argument of
latitude correction
              delta r k = Crs * sin(2 * Phi k) + Crc * cos(2 * Phi k); // radius
correction
              delta_i_k = Cis * sin(2 * Phi_k) + Cic * cos(2 * Phi_k); // inclination
correction
              u_k = Phi_k + delta_u_k; // corrected argument of latitude r_k = A * (1 - e * cos(E_k)) + delta_r_k; // corrected radius
              i_k = i0 + delta_i_k + IDOT * tk; // corrected inclination
              // positions in orbital
              x_k = r_k * cos(u_k);
              y_k = r_k * sin(u_k);
              Omega_k = Om01 + (OmegaDot - Omega_e) * tk - Omega_e * toe; // corrected
longitude of ascending node
              // earth-fixed
              x_k1 = x_k * cos(Omega_k) - y_k * cos(i_k) * sin(Omega_k);
              y_k1 = x_k * sin(Omega_k) + y_k * cos(i_k) * cos(Omega_k);
              z_k1 = y_k * sin(i_k);
              x1\_coord[i] = x_k1;
              y1\_coord[i] = y_k1;
              z1\_coord[i] = z_k1;
              R[i] = sqrt(pow(x1_coord[i], 2) + pow(y1_coord[i], 2) + pow(z1_coord[i],
2));
              //cout << R[i] << endl;
       }
#pragma warning(disable : 4996)
       FILE* fin;
       fin = fopen("Data1.txt", "r");
       if (fin == NULL)
       {
              puts("Файл input.txt не найден!");
       }
       else
       {
              int k;
              double N;
              k = 0;
              while (fscanf(fin, "%d", &N) != EOF)
                     Rmc[k] = N;
                     k++;
                     cout << N << endl;</pre>
              }
       }*/
       ifstream f;
       f.open("Data2.txt");
       //double N;
       maxres = 0;
```

```
for (int i = 0; i < 43199; i++)</pre>
{
        f >> Rmc[i];
        //cout << Rmc[0]<<endl;
res[i] = fabs(R[i+1] - Rmc[i]);
        //cout << res[i] << endl;</pre>
        if (maxres < res[i])</pre>
                maxres = res[i];
}
cout << maxres << endl;</pre>
//cout << R[0]<<endl;
//cout << Rmc[0]<<endl;
delete[] R;
delete[] Rmc;
delete[] res;
delete[] x1_coord;
delete[] y1_coord;
delete[] z1_coord;
return 0;
```

}

Отчет по профилированю

Имя функции	Чи сл о вы зов ов	% затра ченн ого инкл юзив ного врем ени	% затра ченн ого экскл юзив ного врем ени	Сред нее затра ченн ое инкл юзив ное врем я	Сред нее затра ченн ое экскл юзив ное врем я	Имя моду ля
						beido u2.0.
mainCRTStartup	1	100	0,39	2,02	0,01	exe
						beido
						u2.0.
scrt_common_main	1	99,6	3,06	2,01	0,06	exe
						beido u2.0.
scrt_common_main_seh	1	96,53	1,47	1,95	0,03	exe
		00,00	_, .,	_,	0,00	beido
						u2.0.
main	1	86,32	0,11	1,74	0	exe
						beido
						u2.0.
invoke_main	1	86,34	0,02	1,74	0	exe
??6?\$basic_ostream@DU?\$char_traits@D@std@						MSVC P140
@@std@@QAEAAV01@O@Z	5	71,42	71,42	0,29	0,29	D.dll
C C 344 C C C 11 11 11 11 12 3 C 1		, ,,,,_	, _,	0,20	0,20	ucrtb
						ased.
stdio_common_vfprintf	2	3,76	3,76	0,04	0,04	dll
						beido
Nelson and in this line of the late was all	_	2.01	0.65	0.04	0.01	u2.0.
`dynamic initializer for 'storage''	1	2,01	0,65	0,04	0,01	exe beido
						u2.0.
pre_c_initialization	1	1,82	0,05	0,04	0	exe
	-	,	-,	-,		beido
						u2.0.
_vfprintf_l	2	3,8	0,03	0,04	0	exe
						beido
printf		2.02	0.01	0.04		u2.0.
printf	2	3,82	0,01	0,04	0	exe
						KERN EL32.
SetUnhandledExceptionFilter	1	1,63	1,63	0,03	0,03	DLL

		I	I			beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>1,29</td><td>0,55</td><td>0,03</td><td>0,01</td><td>exe</td></testcaseinfo<>	1	1,29	0,55	0,03	0,01	exe
		, -	-,	-,	-,-	beido
						u2.0.
`dynamic initializer for '_test_kepler_function''	1	1,53	0,02	0,03	0	exe
		·	·			beido
						u2.0.
pre_cpp_initialization	1	1,66	0,02	0,03	0	exe
						beido
						u2.0.
scrt_set_unhandled_exception_filter	1	1,64	0	0,03	0	exe
						MSVC
?sputn@?\$basic_streambuf@DU?\$char_traits@D						P140
@std@@@std@@QAE_JPBD_J@Z	4	4,38	4,38	0,02	0,02	D.dll
						MSVC
?put@?\$basic_ostream@DU?\$char_traits@D@st						P140
d@@@std@@QAEAAV12@D@Z	5	3,96	3,96	0,02	0,02	D.dll
						beido
P. C. C.		0.07	0.07	0.00	0.00	u2.0.
scrt_narrow_argv_policy::configure_argv	1	0,97	0,97	0,02	0,02	exe
						beido
stduanarators a cetduahar traits cebar	1	4 00	0.00	0.02	0	u2.0.
std::operator<< <std::char_traits<char>></std::char_traits<char>	4	4,88	0,09	0,02	0	exe beido
						u2.0.
std::endl <char< td=""><td>5</td><td>4,5</td><td>0,05</td><td>0,02</td><td>0</td><td>exe</td></char<>	5	4,5	0,05	0,02	0	exe
??6?\$basic_ostream@DU?\$char_traits@D@std@		.,5	0,03	0,02		MSVC
@@std@@QAEAAV01@P6AAAV01@AAV01@@Z						P140
@Z	5	4,54	0,04	0,02	0	D.dll
				,		beido
						u2.0.
TestCaseProcessor::TestCaseProcessor	1	0,77	0,01	0,02	0	exe
						beido
						u2.0.
pow <double< td=""><td>1</td><td>0,48</td><td>0,48</td><td>0,01</td><td>0,01</td><td>exe</td></double<>	1	0,48	0,48	0,01	0,01	exe
						beido
						u2.0.
scrt_initialize_crt	1	0,6	0,45	0,01	0,01	exe
						beido
		0.40	0.40	0.04	0.04	u2.0.
kepler	1	0,42	0,42	0,01	0,01	exe
						beido u2.0.
scrt_initialize_default_local_stdio_options	1	0,56	0.24	0.01	0.01	
scrt_initialize_default_local_stulo_options	T	0,30	0,34	0,01	0,01	exe beido
						u2.0.
_initialize_default_precision	1	0,33	0,33	0,01	0,01	exe
		0,33	0,33	5,01	0,01	beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,57</td><td>0,03</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,57	0,03	0,01	0	exe
233372000 - 20000000000000000000000000000000		2,3,	2,00	J,J±	J	٥٨٠٥

				I	1	beido
std::_Container_base12::_Alloc_proxy <std::allocat< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocat<>						u2.0.
or <std::_container_proxy>></std::_container_proxy>	2	0,78	0,02	0,01	0	exe
or stan_container_prox/*	_	0,70	0,02	0,01		beido
						u2.0.
std::basic_string <char< td=""><td>1</td><td>0,27</td><td>0,02</td><td>0,01</td><td>0</td><td>exe</td></char<>	1	0,27	0,02	0,01	0	exe
	_	<u> </u>	0,01	0,01		beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,29</td><td>0,02</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,29	0,02	0,01	0	exe
		-, -	-,-	-,-		beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,39</td><td>0,02</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,39	0,02	0,01	0	exe
			-			beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,45</td><td>0,02</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,45	0,02	0,01	0	exe
			-			beido
						u2.0.
std::basic_string <char< td=""><td>1</td><td>0,3</td><td>0,02</td><td>0,01</td><td>0</td><td>exe</td></char<>	1	0,3	0,02	0,01	0	exe
-						beido
						u2.0.
`dynamic atexit destructor for 'storage''	1	0,46	0,01	0,01	0	exe
						beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,58</td><td>0,01</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,58	0,01	0,01	0	exe
						beido
std::_Default_allocator_traits <std::allocator<testc< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<testc<>						u2.0.
aseInfo>>::construct <testcaseinfo< td=""><td>1</td><td>0,26</td><td>0,01</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,26	0,01	0,01	0	exe
						beido
						u2.0.
TestCaseInfo::TestCaseInfo	1	0,31	0	0,01	0	exe
						beido
			_			u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,59</td><td>0</td><td>0,01</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,59	0	0,01	0	exe
						beido
and the second state of the second		0.56	0	0.01		u2.0.
post_pgo_initialization	1	0,56	0	0,01	0	exe
	24					beido
Chack Far Dahuggar Luct Mu Cada	31 9	0.22	0.22	0		u2.0.
CheckForDebuggerJustMyCode	9	0,33	0,33	0	0	exe
?flush@?\$basic_ostream@DU?\$char_traits@D@s						MSVC P140
td@@@std@@QAEAAV12@XZ	5	0,28	0,28	0	0	D.dll
tue e estue e QAEAAVIZ e AZ	Э	0,20	0,20	U	U	beido
						u2.0.
std::allocator <std::_container_proxy>::allocate</std::_container_proxy>	4	0,74	0,28	0	0	exe
staanocator stacontainer_proxy>anocate	4	0,74	0,20	U	0	beido
						u2.0.
std:: Allocate<8	5	0,47	0,28	0	0	exe
Stati_/ modute so		<i>5,</i> −77	0,20			beido
						u2.0.
onexit	3	0,24	0,24	0	0	exe
		J,27	<i>∪,</i> ∠¬	U	J	CAC

		Ī	İ			beido
						u2.0.
atan2l	1	0,22	0,22	0	0	exe
		- /	-,			beido
						u2.0.
operator delete	5	0,2	0,2	0	0	exe
		-	-			MSVC
?widen@?\$basic_ios@DU?\$char_traits@D@std@						P140
@@std@@QBEDD@Z	5	0,2	0,2	0	0	D.dll
						beido
						u2.0.
operator new	6	0,19	0,19	0	0	exe
						beido
						u2.0.
scrt_file_policy::set_commode	1	0,15	0,15	0	0	exe
						beido
						u2.0.
powl	3	0,12	0,12	0	0	exe
						KERN
						EL32.
IsProcessorFeaturePresent	1	0,11	0,11	0	0	DLL
						beido
atdulacia astronora sabar	4	0.10	0.11	0		u2.0.
std::basic_ostream <char< td=""><td>4</td><td>0,18</td><td>0,11</td><td>0</td><td>0</td><td>exe</td></char<>	4	0,18	0,11	0	0	exe
						beido u2.0.
cosl	12	0,09	0,09	0	0	exe
CO31	12	0,03	0,03	0		MSVC
						P140
??0_Lockit@std@@QAE@H@Z	6	0,07	0,07	0	0	D.dll
		7,51	7,51	-		beido
						u2.0.
scrt_file_policy::set_fmode	1	0,07	0,07	0	0	exe
		·	·			beido
						u2.0.
std::_Construct_in_place <std::_container_proxy< td=""><td>4</td><td>0,15</td><td>0,07</td><td>0</td><td>0</td><td>exe</td></std::_container_proxy<>	4	0,15	0,07	0	0	exe
						beido
						u2.0.
std::_Narrow_char_traits <char< td=""><td>5</td><td>0,06</td><td>0,06</td><td>0</td><td>0</td><td>exe</td></char<>	5	0,06	0,06	0	0	exe
						beido
						u2.0.
std::_Container_base12::_Orphan_all	6	0,16	0,06	0	0	exe
						beido
	4.0	0.45	0.05	_		u2.0.
cos	12	0,15	0,05	0	0	exe
						beido
std:westersTestCaseInfo	1	0,05	0.05	0		u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,05</td><td>0,05</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,05	0,05	0	0	exe
						beido
sinl	8	0,05	0,05	0	0	u2.0.
SIIII	ō	0,05	0,05	U	U	exe

1		ĺ	ĺ	I	ĺ	beido
						u2.0.
std::basic_string <char< td=""><td>3</td><td>0,5</td><td>0,05</td><td>0</td><td>0</td><td>exe</td></char<>	3	0,5	0,05	0	0	exe
standaro_string tendi		0,0	0,00			beido
						u2.0.
std::basic_ostream <char< td=""><td>4</td><td>0,1</td><td>0,05</td><td>0</td><td>0</td><td>exe</td></char<>	4	0,1	0,05	0	0	exe
	•		0,00			beido
						u2.0.
isa_available_init	1	0,16	0,04	0	0	exe
		-, -	-,-		_	beido
	42					u2.0.
_RTC_CheckEsp	8	0,04	0,04	0	0	exe
			,			beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_Large_string_engaged	6	0,04	0,04	0	0	exe
			,			beido
						u2.0.
_RTC_CheckStackVars	19	0,04	0,04	0	0	exe
		·	·			beido
						u2.0.
sin	8	0,09	0,04	0	0	exe
std::_Container_proxy_ptr12 <std::allocator<std::_< td=""><td></td><td></td><td></td><td></td><td></td><td></td></std::allocator<std::_<>						
Container_proxy>						beido
>::_Container_proxy_ptr12 <std::allocator<std::_co< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<std::_co<>						u2.0.
ntainer_proxy> >	2	0,2	0,04	0	0	exe
						beido
						u2.0.
std::basic_string <char< td=""><td>3</td><td>0,11</td><td>0,04</td><td>0</td><td>0</td><td>exe</td></char<>	3	0,11	0,04	0	0	exe
						beido
std::_Default_allocator_traits <std::allocator<std::_< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<std::_<>						u2.0.
Container_proxy> >::deallocate	4	0,24	0,04	0	0	exe
						beido
						u2.0.
std::basic_string <char< td=""><td>7</td><td>0,07</td><td>0,03</td><td>0</td><td>0</td><td>exe</td></char<>	7	0,07	0,03	0	0	exe
						beido
	_					u2.0.
std::_Compressed_pair <std::allocator<char></std::allocator<char>	7	0,04	0,03	0	0	exe
						beido
and a second control of		0.00	0.00			u2.0.
std::vector <testcaseinfo< td=""><td>2</td><td>0,08</td><td>0,03</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	2	0,08	0,03	0	0	exe
						beido
atdu. Daellaasta (0	_	0.24	0.03	0		u2.0.
std::_Deallocate<8	5	0,24	0,03	0	0	exe
ctdy Compressed pairwetdyalleasters Tast Casalaf						beido
std::_Compressed_pair <std::allocator<testcaseinf< td=""><td>6</td><td>0.03</td><td>0.02</td><td>0</td><td></td><td>u2.0.</td></std::allocator<testcaseinf<>	6	0.03	0.02	0		u2.0.
0>	р	0,03	0,03	U	0	exe
						beido u2.0.
stdy Default allocate traits: Allocate	5	0,19	0,03	0	0	
std::_Default_allocate_traits::_Allocate	5	0,19	0,03	U	U	exe beido
						u2.0.
std::_Narrow_char_traits <char< td=""><td>6</td><td>0,03</td><td>0,03</td><td>0</td><td>0</td><td></td></char<>	6	0,03	0,03	0	0	
StuINGITOW_CHAI_CHAICS>CHAI	O	0,05	0,03	U	U	exe

				1	1	beido
std::allocator <std::_container_proxy>::allocator<s< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></s<></std::_container_proxy>						u2.0.
td::_Container_proxy> <char></char>	6	0,03	0,03	0	0	exe
tan_container_proxyx tenur		0,00	0,00			beido
std::_Delete_plain_internal <std::allocator<std::_c< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<std::_c<>						u2.0.
ontainer_proxy>>	4	0,29	0,03	0	0	exe
	-		-,			beido
						u2.0.
std::vector <testcaseinfo< td=""><td>6</td><td>0,07</td><td>0,03</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	6	0,07	0,03	0	0	exe
			,			beido
						u2.0.
std::basic_string <char< td=""><td>1</td><td>0,04</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,04	0,02	0	0	exe
						beido
						u2.0.
std::basic_string <char< td=""><td>1</td><td>0,23</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,23	0,02	0	0	exe
?rdbuf@?\$basic_ios@DU?\$char_traits@D@std@						MSVC
@@std@@QBEPAV?\$basic_streambuf@DU?\$cha						P140
r_traits@D@std@@@2@XZ	12	0,02	0,02	0	0	D.dll
						beido
std::_Destroy_range <std::allocator<testcaseinfo></std::allocator<testcaseinfo>						u2.0.
>	2	0,26	0,02	0	0	exe
						beido
						u2.0.
std::_Char_traits <char< td=""><td>1</td><td>0,02</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,02	0,02	0	0	exe
						MSVC
224		0.00	0.00	0		P140
??1_Lockit@std@@QAE@XZ	6	0,02	0,02	0	0	D.dll
						beido
atdy Cantainay hass12y Cantainay hass12	4	0.03	0.03	0		u2.0.
std::_Container_base12::_Container_base12	4	0,02	0,02	0	0	exe
						MSVC P140
[MSVCP140D.dll]	4	0,02	0,02	0	0	D.dll
[WSVEI 140D.dill]	7	0,02	0,02	U	-	beido
						u2.0.
pow	3	0,14	0,02	0	0	exe
The state of the s		-7-	-7			beido
						u2.0.
std::_Unfancy <testcaseinfo></testcaseinfo>	2	0,02	0,02	0	0	exe
			· · · · · · · · · · · · · · · · · · ·			KERN
						EL32.
GetModuleHandleW	1	0,02	0,02	0	0	DLL
						beido
std::_Deallocate_plain <std::allocator<std::_contai< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<std::_contai<>						u2.0.
ner_proxy>>	4	0,26	0,02	0	0	exe
						beido
						u2.0.
operator new	5	0,02	0,02	0	0	exe
						beido
						u2.0.
std::basic_ostream <char< td=""><td>4</td><td>0,04</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	4	0,04	0,02	0	0	exe

		ĺ	ĺ		1	beido
						u2.0.
security_init_cookie	1	0,02	0,02	0	0	exe
	_	0,02	0,02			beido
						u2.0.
std::_Unfancy <std::_container_proxy></std::_container_proxy>	4	0,02	0,02	0	0	exe
		,				beido
						u2.0.
std::_Get_size_of_n<8>	4	0,02	0,02	0	0	exe
						beido
						u2.0.
std::basic_ostream <char< td=""><td>4</td><td>0,05</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	4	0,05	0,02	0	0	exe
						beido
						u2.0.
std::basic_ostream <char< td=""><td>4</td><td>0,02</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	4	0,02	0,02	0	0	exe
						beido
				_	_	u2.0.
std::basic_string <char< td=""><td>1</td><td>0,04</td><td>0,02</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,04	0,02	0	0	exe
						MSVC
[MCVCD4.40D_JII]		0.02	0.02	0		P140
[MSVCP140D.dll]	4	0,02	0,02	0	0	D.dll
						beido u2.0.
std::_Container_proxy::_Container_proxy	4	0,02	0,01	0	0	exe
stacontainer_proxycontainer_proxy	4	0,02	0,01	U	- 0	MSVC
?_Osfx@?\$basic_ostream@DU?\$char_traits@D@						P140
std@@astd@@QAEXXZ	4	0,01	0,01	0	0	D.dll
	-	-,	-,	-		MSVC
?setstate@?\$basic_ios@DU?\$char_traits@D@std						P140
@@@std@@QAEXH_N@Z	4	0,01	0,01	0	0	D.dll
						beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_String_val <std::_simple_types<char> ></std::_simple_types<char>	3	0,05	0,01	0	0	exe
std::_Uninitialized_backout_al <std::allocator<test< td=""><td></td><td></td><td></td><td></td><td></td><td></td></std::allocator<test<>						
CaseInfo>						beido
>::~_Uninitialized_backout_al <std::allocator<testc< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<testc<>						u2.0.
aseInfo> >	1	0,02	0,01	0	0	exe
						beido
		0.00	0.04	0		u2.0.
std::_Voidify_iter <std::_container_proxy *=""></std::_container_proxy>	4	0,02	0,01	0	0	exe
						beido
stduaddressofestdu Container provis	4	0,02	0,01	0	0	u2.0.
std::addressof <std::_container_proxy></std::_container_proxy>	4	0,02	0,01	U	U	exe beido
						u2.0.
std::forward <std::_container_base12 *=""></std::_container_base12>	4	0,01	0,01	0	0	exe
Stamorward Stam_Container_base12	-	0,01	0,01	U		beido
						u2.0.
sqrtl	1	0,01	0,01	0	0	exe
	_	5,52	2,01			ntdll.
RtlInitializeSListHead	1	0,01	0,01	0	0	dll
		-,	-,			

		I				beido
						u2.0.
std::_Compressed_pair <std::allocator<char></std::allocator<char>	2	0,05	0,01	0	0	exe
- Compressed_pair Standard Standard	_	0,00	0,01			MSVC
						P140
?width@ios_base@std@@QBE_JXZ	4	0,01	0,01	0	0	D.dll
		-,-	-,-			beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_Bxty::_Bxty	3	0,01	0,01	0	0	exe
_ , _ ,		,				beido
						u2.0.
std::_Compressed_pair <std::allocator<char></std::allocator<char>	3	0,04	0,01	0	0	exe
			·			beido
						u2.0.
std::_Uninitialized_move <testcaseinfo *<="" td=""><td>1</td><td>0,07</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></testcaseinfo>	1	0,07	0,01	0	0	exe
						beido
						u2.0.
std::basic_string <char< td=""><td>2</td><td>0,03</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></char<>	2	0,03	0,01	0	0	exe
						beido
						u2.0.
scrt_main_policy::set_app_type	1	0,01	0,01	0	0	exe
						beido
std::_Compressed_pair <std::allocator<testcaseinf< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<testcaseinf<>						u2.0.
0>	1	0,04	0,01	0	0	exe
						MSVC
2 : 11 0: 1 0 : 10 0045 1 107		0.04	0.04			P140
?width@ios_base@std@@QAE_J_J@Z	4	0,01	0,01	0	0	D.dll
						beido
TestCaseInfo::~TestCaseInfo	3	0,51	0.01	0	0	u2.0.
restcaseiiio restcaseiiiio	3	0,51	0,01	0	0	exe beido
						u2.0.
std::forward <testcaseinfo &="" const=""></testcaseinfo>	3	0,02	0,01	0	0	exe
Staorward vresteasening const de	3	0,02	0,01	0		beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_Bxty::~_Bxty	3	0,01	0,01	0	0	exe
		-,-	-,-			beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::~_String_val <std::_simple_types<char>></std::_simple_types<char>	3	0,02	0,01	0	0	exe
			·			beido
						u2.0.
local_stdio_printf_options	3	0,23	0,01	0	0	exe
						beido
						u2.0.
pow <long double<="" td=""><td>2</td><td>0,11</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></long>	2	0,11	0,01	0	0	exe
						beido
						u2.0.
std::vector <testcaseinfo< td=""><td>2</td><td>0,02</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	2	0,02	0,01	0	0	exe
						beido
std::_Default_allocator_traits <std::allocator<testc< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<testc<>						u2.0.
aseInfo> >::destroy <testcaseinfo></testcaseinfo>	1	0,23	0,01	0	0	exe

std::allocator <std::_container_proxy><testcaseinfo> 2 0,01 0,01 0 0 0 exe std::_Container_proxy><testcaseinfo> 2 0,01 0,01 0 0 0 exe std::basic_string 1 0,05 0,01 0 0 0 exe std::_Container_proxy_ptr12<std::allocator<std::container_proxy> beid u2.0 ontainer_proxy> 2 0,01 0,01 0 0 0 exe >::~_Container_proxy> > 2 0,01 0,01 0 0 0 exe >:flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 D.dll ?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 exe MSV ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@@2@XZ 4 0,01 0,01 0 0 0 D.dll std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0,01 0 0 exe</std::allocator<std::container_proxy></testcaseinfo></testcaseinfo></std::_container_proxy>	<u> </u>	2	0.01	0.01			beido u2.0.
td::_Container_proxy> <testcaseinfo> 2 0,01 0,01 0 exe std::basic_string beid u2.0 std::Container_proxy_ptr12<std::allocator<std::_container_proxy> beid u2.0 >::~Container_proxy> beid u2.0 ontainer_proxy> 2 0,01 0,01 0 exe P14(MSV P14(P14(<td><u> </u></td><td>2</td><td>0.01</td><td>0.01</td><td></td><td></td><td></td></std::allocator<std::_container_proxy></testcaseinfo>	<u> </u>	2	0.01	0.01			
std::basic_string 1 0,05 0,01 0 0 exe std::_Container_proxy> beid u2.0 >::~_Container_proxy> beid u2.0 >::~_Container_proxy> 2 0,01 0,01 0 0 exe MSV P140			0,01	0,01	0	0	exe
std::basic_string 1 0,05 0,01 0 0 exe std::_Container_proxy_ptr12< <std::allocator<std::_container_proxy> >::~_Container_proxy_ptr12<std::allocator<std::_c ontainer_proxy=""> 2 0,01 0,01 0 0 exe MSV P14(Container_proxy) P14(Container_proxy)</std::allocator<std::_c></std::allocator<std::_container_proxy>				,			beido
std::_Container_proxy_ptr12 <std::allocator<std::_< td=""> beid >:::~Container_proxy> 2 0,01 0,01 0 0 0 exe >:::~Container_proxy> > 2 0,01 0,01 0 0 0 exe Ontainer_proxy> > 2 0,01 0,01 0 0 0 exe ***Flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 D.dll ***std::basic_string<char< td=""> 1 0,08 0,01 0 0 0 exe ***P14C ***good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ***?tie@?\$basic_ios@DU?\$char_traits@D@std@@ MSV **@std@@QBEPAV?\$basic_ostream@DU?\$char_tr 4 0,01 0,01 0 0 0 D.dll ***std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe</char<></std::allocator<std::_<>							u2.0.
std::_Container_proxy_ptr12 <std::allocator<std::_< td=""> beid >::~_Container_proxy> 2 0,01 0,01 0 0 0 exe >::~_Container_proxy> > 2 0,01 0,01 0 0 0 exe Ontainer_proxy> > 2 0,01 0,01 0 0 0 exe /**P14(**) P14(**) ?**Ilags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 exe ***Std::basic_string<char< td=""> 1 0,08 0,01 0 0 0 exe ***P14(**) MSV ***P14(**) P14(**) ***?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ***?tie@?\$basic_ios@DU?\$char_traits@D@std@@ MSV **@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@@2@XZ 4 0,01 0,01 0 0 0 D.dll ***beid** beid** ***ut:_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 0 exe</char<></std::allocator<std::_<>	td::basic string <char< td=""><td>1</td><td>0,05</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,05	0,01	0	0	exe
Container_proxy> beid u2.0 ontainer_proxy_ptr12 <std::allocator<std::_c ontainer_proxy="">> 2 0,01 0,01 0 0 0 exe ?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 D.dll ?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 D.dll std::basic_string<char< td=""> 1 0,08 0,01 0 0 0 exe ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@@2@XZ 4 0,01 0,01 0 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe</char<></std::allocator<std::_c>				-			
>::~_Container_proxy_ptr12 <std::allocator<std::_c ontainer_proxy="">></std::allocator<std::_c>							beido
?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 0 D.dll std::basic_string <char< td=""> 1 0,08 0,01 0 0 0 exe ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@altaits@D@std@altaits@D@std@std@altaits@D@std@std@std@std@std@std@std@std@std@std</char<>	::~_Container_proxy_ptr12 <std::allocator<std::_c< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<std::_c<>						u2.0.
?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 D.dll std::basic_string <char< td=""> 1 0,08 0,01 0 0 exe ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@gstd@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@g2@XZ 4 0,01 0,01 0 0 D.dll std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe</char<>	ntainer_proxy> >	2	0,01	0,01	0	0	exe
?flags@ios_base@std@@QBEHXZ 4 0,01 0,01 0 0 D.dll std::basic_string 1 0,08 0,01 0 0 exe							MSVC
std::basic_string <char< td=""> 1 0,08 0,01 0 0 exe ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@@std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@@2@XZ MSV @std@@QBEPAV?\$basic_ostream@DU?\$char_traits@D@std@@ D.dll beid beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe</char<>							P140
std::basic_string 1 0,08 0,01 0 0 exe MSV ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@ MSV @std@@QBEPAV?\$basic_ostream@DU?\$char_tr 4 0,01 0,01 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe	flags@ios_base@std@@QBEHXZ	4	0,01	0,01	0	0	D.dll
std::basic_string 1 0,08 0,01 0 0 exe MSV P140 ?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@ @std@@QBEPAV?\$basic_ostream@DU?\$char_tr aits@D@std@@@2@XZ 4 0,01 0,01 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe							beido
MSV P140 P							u2.0.
?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@ MSV @std@@QBEPAV?\$basic_ostream@DU?\$char_tr 4 0,01 0,01 0 0 0 D.dll aits@D@std@@@2@XZ 4 0,01 0,01 0 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe	td::basic_string <char< td=""><td>1</td><td>0,08</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,08	0,01	0	0	exe
?good@ios_base@std@@QBE_NXZ 4 0,01 0,01 0 0 D.dll ?tie@?\$basic_ios@DU?\$char_traits@D@std@@ @std@@QBEPAV?\$basic_ostream@DU?\$char_tr # P140 aits@D@std@@@2@XZ 4 0,01 0,01 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0							MSVC
?tie@?\$basic_ios@DU?\$char_traits@D@std@@ MSV @std@@QBEPAV?\$basic_ostream@DU?\$char_tr 4 0,01 0,01 0 0 D.dll aits@D@std@@@2@XZ 4 0,01 0,01 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe							P140
@std@@QBEPAV?\$basic_ostream@DU?\$char_tr 4 0,01 0,01 0 0 0 D.dll aits@D@std@@@2@XZ 4 0,01 0,01 0 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 0 exe	good@ios_base@std@@QBE_NXZ	4	0,01	0,01	0	0	D.dll
aits@D@std@@@2@XZ 4 0,01 0,01 0 0 D.dll beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe							MSVC
beid u2.0 std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe	9std@@QBEPAV?\$basic_ostream@DU?\$char_tr						P140
std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe	its@D@std@@@2@XZ	4	0,01	0,01	0	0	D.dll
std::_Basic_container_proxy_ptr12::_Release 2 0,01 0,01 0 0 exe							beido
							u2.0.
	td::_Basic_container_proxy_ptr12::_Release	2	0,01	0,01	0	0	exe
							beido
							u2.0.
std::_Compressed_pair <std::allocator<char> 1 0,03 0,01 0 0 exe</std::allocator<char>	td::_Compressed_pair <std::allocator<char></std::allocator<char>	1	0,03	0,01	0	0	
							beido
			0.04	2.24			u2.0.
aseInfo> >::max_size	seinfo> >::max_size	2	0,01	0,01	0	0	
							beido
	cort is managed ann	1	0.02	0.01	0	0	u2.0.
	_scrt_is_manageu_app	1	0,03	0,01	U	U	beido
							u2.0.
std::forward <std::allocator<char>></std::allocator<char>	td:·forward <std:·allocator<char>></std:·allocator<char>	2	0.01	0.01	0	0	
, , ,	taorward\staanocator\char>		0,01	0,01	0	0	beido
							u2.0.
std::numeric_limits <int>::max</int>	td::numeric limits <int>::max</int>	2	0.01	0.01	0	0	
	camanana_mak	_	0,01	0,01	-		beido
	td:: Compressed pair <std::allocator<testcaseinf< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<testcaseinf<>						u2.0.
o>	_ · · _·	2	0.01	0.01	0	0	
			-7	-7	-		beido
							u2.0.
std::basic_string <char< td=""><td>td::basic string<char< td=""><td>1</td><td>0,13</td><td>0.01</td><td>0</td><td>0</td><td></td></char<></td></char<>	td::basic string <char< td=""><td>1</td><td>0,13</td><td>0.01</td><td>0</td><td>0</td><td></td></char<>	1	0,13	0.01	0	0	
			, -	,-	-	-	beido
							u2.0.
std::addressof <std::_container_base12> 2 0,01 0,01 0 0 exe</std::_container_base12>	thunddroscofestal Containor base125	2	0,01	0,01	0	0	
	tuauuressor <stucontainer dase12=""></stucontainer>	1 1	,	•		-	
	tuaudressor <stu::_container_dase12></stu::_container_dase12>						beido
operator delete 5 0,21 0,01 0 exe	tuauuressor <sta::_container_dase12></sta::_container_dase12>						beido u2.0.

						beido
						u2.0.
std::_Get_unwrapped <testcaseinfo &="" *="" const=""></testcaseinfo>	2	0,01	0,01	0	0	exe
stan_det_unwrupped vresteusenno const d>		0,01	0,01	0	0	beido
						u2.0.
std::min <unsigned int=""></unsigned>	2	0,01	0,01	0	0	exe
Stallinin tansigned into		0,01	0,01	-		beido
						u2.0.
std::_Char_traits <char< td=""><td>1</td><td>0,01</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,01	0,01	0	0	exe
	_	0,01	0,01			beido
std::_Uninitialized_backout_al <std::allocator<test< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<test<>						u2.0.
CaseInfo> >:: autoclassinit2	1	0,01	0,01	0	0	exe
			,			beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,09</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,09	0,01	0	0	exe
						beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,04</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,04	0,01	0	0	exe
						beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_Myptr	1	0,01	0,01	0	0	exe
						beido
						u2.0.
std::vector <testcaseinfo< td=""><td>1</td><td>0,03</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	1	0,03	0,01	0	0	exe
						beido
						u2.0.
std::basic_string <char< td=""><td>1</td><td>0,01</td><td>0,01</td><td>0</td><td>0</td><td>exe</td></char<>	1	0,01	0,01	0	0	exe
						beido
Total Control of the Total Control of	4	0.24	0.01	0	0	u2.0.
TestCaseInfo::TestCaseInfo	1	0,24	0,01	0	0	exe
						beido
atan2	1	0,23	0,01	0	0	u2.0. exe
atanz		0,23	0,01	U	U	beido
						u2.0.
scrt_initialize_onexit_tables	1	0,01	0,01	0	0	exe
sate_midalize_onexit_tables		0,01	0,01	Ŭ		beido
std::_String_val <std::_simple_types<char></std::_simple_types<char>						u2.0.
>::_Myptr	1	0,02	0,01	0	0	exe
= 71		,	,			beido
scrt_narrow_environment_policy::initialize_envi						u2.0.
ronment	1	0,01	0,01	0	0	exe
		·	·			beido
						u2.0.
std::_Compressed_pair <std::allocator<char></std::allocator<char>	1	0,01	0,01	0	0	exe
						beido
						u2.0.
sqrt	1	0,02	0	0	0	exe
						beido
						u2.0.
std::allocator <testcaseinfo>::allocate</testcaseinfo>	1	0,04	0	0	0	exe

std::_get_size_of_n<32>							beido
Std::_Uninitialized_backout_al <std::allocator<test< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></std::allocator<test<>							
Std::_Uninitialized_backout_al <std::allocator<test< td=""><td>std::_Get_size_of_n<32></td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>exe</td></std::allocator<test<>	std::_Get_size_of_n<32>	1	0	0	0	0	exe
CaseInfo>>::_Release							beido
Std::exchange <std::_container_proxy*< td=""><td>std::_Uninitialized_backout_al<std::allocator<test< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<test<></td></std::_container_proxy*<>	std::_Uninitialized_backout_al <std::allocator<test< td=""><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></std::allocator<test<>						u2.0.
std::exchange <std::_container_proxy*< td=""> 1 0 0 0 0 exe exe beido u2.0. beido u2.0. beido u2.0. beido u2.0. exe beido u2.0. u2.0. beido u2.0. u2.0. beido u2.0. u2.0. beido u2.0. u2.0. u2.0. u2.0. beido u2.0. u2.0.</std::_container_proxy*<>		1	0	0	0	0	exe
Std::exchange <std::_container_proxy*< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>beido</td></std::_container_proxy*<>							beido
Std::vector <testcaseinfo< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>u2.0.</td></testcaseinfo<>							u2.0.
Std::vector <testcaseinfo< td=""><td>std::exchange<std::_container_proxy *<="" td=""><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>exe</td></std::_container_proxy></td></testcaseinfo<>	std::exchange <std::_container_proxy *<="" td=""><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>exe</td></std::_container_proxy>	1	0	0	0	0	exe
Std::vector <testcaseinfo< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>beido</td></testcaseinfo<>							beido
test_kepler_function::test_kepler_function							u2.0.
test_kepler_function::test_kepler_function	std::vector <testcaseinfo< td=""><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	1	0	0	0	0	exe
test_kepler_function::test_kepler_function							beido
beido							u2.0.
Std::allocator <testcaseinfo>::deallocate</testcaseinfo>	test_kepler_function::test_kepler_function	1	0,01	0	0	0	exe
std::allocator <testcaseinfo>::deallocate 1 0,04 0 0 exe std::vector<testcaseinfo< td=""> 1 0,09 0 0 0 exe std::Uninitialized_backout_al<std::allocator<testcaseinfo> 1 0 0 0 0 exe seInfo> 1 0 0 0 0 0 exe acrt_iob_func 2 0 0 0 0 0 0 dll acrt_iob_func 1 0 0 0 0 0 0 0 0 0 0 exe acrt_iob_func 1 0 exe 0 0<td></td><td></td><td></td><td></td><td></td><td></td><td>beido</td></std::allocator<testcaseinfo></testcaseinfo<></testcaseinfo>							beido
beido							u2.0.
Std::vector <testcaseinfo< td=""><td>std::allocator<testcaseinfo>::deallocate</testcaseinfo></td><td>1</td><td>0,04</td><td>0</td><td>0</td><td>0</td><td>exe</td></testcaseinfo<>	std::allocator <testcaseinfo>::deallocate</testcaseinfo>	1	0,04	0	0	0	exe
std::vector <testcaseinfo< td=""> 1 0,09 0 0 exe std::_Uninitialized_backout_al<std::allocator<testcaseinfo> <td< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td>beido</td></td<></std::allocator<testcaseinfo></testcaseinfo<>							beido
std::_Uninitialized_backout_al <std::allocator<test caseinfo=""> beido u2.0. selnfo> >::_Uninitialized_backout_al<std::allocator<testca selnfo=""> > 1 0 0 0 0 0 exe selnfo> > ucrtb ased. acrt_iob_func 2 0 0 0 0 0 0 dll std::allocator<testcaseinfo>::allocator<testcasein fo=""> 1 0 0 0 0 0 0 exe beido u2.0. beido u2.0. std::allocator<char>::allocator<char> 1 0 0 0 0 0 exe beido u2.0. std::move<std::allocator<char> &> 1 0 0 0 0 0 0 exe beido u2.0. std::addressof<std::_string_val<std::_simple_type s<="" td=""> 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 0 0 exe schar>> const> 1 0 0 0 0 0 0 exe schar>> const> 1 0 0 0 0 0 0 0 exe pow<long double<="" td=""> 1 0,04 0 0 0 0 0 exe schar </long></std::_string_val<std::_simple_type></std::allocator<char></char></char></testcasein></testcaseinfo></std::allocator<testca></std::allocator<test>							u2.0.
CaseInfo> beido >::_Uninitialized_backout_al <std::allocator<testca< td=""> 1 0 0 0 exe acrt_iob_func 2 0 0 0 0 dll acrt_iob_func 2 0 0 0 0 dll acrt_iob_func 1 0 0 0 0 dll acrt_iob_func 2 0 0 0 0 dll acrt_iob_func 1 0 0 0 0 dll acrt_iob_func 1 0 0 0 0 dll acrt_iob_func 1 0 0 0 0 exe acrt_iob_func 1 0 0 0 0 exe acrt_iob_func 1 0 0 0 0 exe act_iob_func 1 0 0 0 0 exe act_iob_func 1 0 0 0<</std::allocator<testca<>		1	0,09	0	0	0	exe
Selnfo>>							
acrt_iob_func							u2.0.
acrt_iob_func	seInfo>>	1	0	0	0	0	
acrt_iob_func							
beido u2.0.							
std::allocator <testcaseinfo>::allocator<testcasein fo=""> 1 0 0 0 0 exe beido u2.0. std::allocator<char>::allocator<char> 1 0 0 0 0 exe std::move<std::allocator<char>&> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type s<="" td=""> 1 0 0 0 0 exe </std::_string_val<std::_simple_type></std::allocator<char></char></char></testcasein></testcaseinfo>	acrt_iob_func	2	0	0	0	0	
fo> 1 0 0 0 exe beido u2.0. std::allocator <char>::allocator<char> 1 0 0 0 0 exe beido u2.0. std::move<std::allocator<char> &> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type s<="" td=""> 1 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 exe scchar>> const > 1 0 0 0 0 exe pow<length< td=""> 1 0,04 0 0 0 exe pow<length< td=""> 1 0,04 0 0 0 exe</length<></length<></std::_string_val<std::_simple_type></std::allocator<char></char></char>							
std::allocator <char>::allocator<char> 1 0 0 0 0 exe beido u2.0. bei</char></char>		_	•	•	0		
std::allocator <char>::allocator<char> 1 0 0 0 0 exe std::move<std::allocator<char>&> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe scchar>>> 1 0 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> </std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<></std::allocator<char></char></char>	10>	1	0	0	Ü	0	
std::allocator <char>::allocator<char> 1 0 0 0 exe std::move<std::allocator<char>&> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 exe beido u2.0. exe beido u2.0. scrt_acquire_startup_lock 1 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 0 0 0 exe schar>> const 1 0 0 0 0 exe pow<long double<="" td=""> 1 0,04 0 0 0 exe</long></std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<></std::allocator<char></char></char>							
std::move <std::allocator<char> &> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 exe beido u2.0. u2.0. exe beido u2.0. s 1 0 0 0 0 exe beido u2.0. exe beido u2.0. exe pow<long double<="" td=""> 1 0,04 0 0 0 exe</long></std::_string_val<std::_simple_type<></std::allocator<char>	ctduallocatorschars unllocatorschars	1	0	0	0	0	
std::move <std::allocator<char> &> 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe scchar>>> 1 0 0 0 0 0 exe scrt_acquire_startup_lock 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 0 0 0 0 exe s 1 0 0 0 0 exe pow<long double<="" td=""> 1 0,04 0 0 0 exe beido u2.0. exe beido u2.0. exe</long></std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<></std::allocator<char>	Studilocator <cridi>dilocator<cridi></cridi></cridi>	1	U	U	U	U	
std::move <std::allocator<char> &> 1 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 exe beido u2.0. u2.0. exe beido u2.0. s 1 0 0 0 0 exe beido std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 exe beido u2.0. exe pow<long double<="" td=""> 1 0,04 0 0 0 exe</long></std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<></std::allocator<char>							
std::addressof <std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 0 exe beido u2.0. u2.0. exe beido u2.0. exe beido u2.0. exe std::addressof<std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 exe beido u2.0. pow<long double<="" td=""> 1 0,04 0 0 0 exe beido u2.0. exe beido u2.0. exe beido u2.0. exe</long></std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<>	std::move <std::allocator<char> &></std::allocator<char>	1	0	0	0	0	
std::addressof <std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s 1 0 0 0 0 0 exe beido u2.0. scrt_acquire_startup_lock 1 0 0 0 0 exe std::addressof<std::_string_val<std::_simple_type< td=""> </std::_string_val<std::_simple_type<></std::_string_val<std::_simple_type<>	stumove\stuanocator\chai> &>	1	U	0	U	0	
s <char>>> 1 0 0 0 0 exe beido u2.0. u2.0. u2.0. u2.0. exe std::addressof<std::_string_val<std::_simple_type s<="" td=""> u2.0. u2.0. u2.0. u2.0. u2.0. exe pow<long double<="" td=""> 1 0,04 0 0 0 exe beido u2.0. beido u2.0. u2.0.</long></std::_string_val<std::_simple_type></char>	std.:addressof <std.: simple="" string="" td="" type<="" val<std.:=""><td></td><td></td><td></td><td></td><td></td><td></td></std.:>						
beido u2.0.		1	0	0	n	0	
scrt_acquire_startup_lock	3 Kelidi 2 7 2		-				
scrt_acquire_startup_lock							
std::addressof <std::_string_val<std::_simple_type s<br=""></std::_string_val<std::_simple_type> s <char> > const > 1 0 0 0 0 0 exe beido u2.0. pow<long 0="" 0,04="" 1="" double="" exe<br=""></long>beido u2.0.</char>	scrt acquire startup lock	1	0	0	0	0	
std::addressof <std::_string_val<std::_simple_type< td=""> 1 0 0 0 0 exe s<char> > const > 1 0 0 0 0 exe beido u2.0. pow<long double<="" td=""> 1 0,04 0 0 0 exe beido u2.0.</long></char></std::_string_val<std::_simple_type<>		_					
s <char> > const > 1 0 0 0 0 exe beido u2.0. pow<long double<="" td=""> 1 0,04 0 0 0 exe beido u2.0. beido u2.0.</long></char>	std::addressof <std:: simple="" string="" td="" type<="" val<std::=""><td></td><td></td><td></td><td></td><td></td><td></td></std::>						
beido u2.0. pow <long 0="" 0,04="" 1="" beido="" double="" exe="" td="" u2.0.<=""><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td></long>		1	0	0	0	0	
pow <long 0="" 0,04="" 1="" description<="" double="" td="" u2.0.="" =""><td></td><td></td><td></td><td></td><td></td><td></td><td></td></long>							
pow <long double<="" th=""> 1 0,04 0 0 0 exe beido u2.0.</long>							
beido u2.0.	pow <long double<="" td=""><td>1</td><td>0,04</td><td>0</td><td>0</td><td>0</td><td></td></long>	1	0,04	0	0	0	
u2.0.	-		-				
scrt initialize type info							u2.0.
	scrt_initialize_type_info	1	0,02	0	0	0	exe

std::_Vector_val <std::_simple_types<testcaseinfo< th=""><th></th><th></th><th></th><th></th><th></th><th></th></std::_simple_types<testcaseinfo<>						
>						beido
>::_Vector_val <std::_simple_types<testcaseinfo></std::_simple_types<testcaseinfo>			_	_	_	u2.0.
>	1	0,01	0	0	0	exe
						beido
						u2.0.
std::_Convert_size <unsigned int=""></unsigned>	1	0	0	0	0	exe
atdu. Dafa dt alla satan tuaita est dualla satan est an						beido
std::_Default_allocator_traits <std::allocator<char></std::allocator<char>	1	0	0	0	0	u2.0.
>::select_on_container_copy_construction	1	0	0	0	U	exe beido
						u2.0.
scrt_release_startup_lock	1	0	0	0	0	exe
scrt_release_startup_lock	т	0	0	0	0	beido
						u2.0.
security_check_cookie	19	0	0	0	0	exe
						beido
						u2.0.
TestCase::TestCase	1	0	0	0	0	exe
						beido
						u2.0.
TestCaseInfo::`scalar deleting destructor'	1	0,22	0	0	0	exe
						beido
						u2.0.
atexit	3	0,24	0	0	0	exe
						beido
						u2.0.
std::`dynamic atexit destructor for '_Fac_tidy_reg"	1	0	0	0	0	exe
						beido
			_	_	_	u2.0.
std::`dynamic atexit destructor for '_Fac_tidy_reg"	1	0,02	0	0	0	exe
						beido
atdu. Faa tidu waa tux Faa tidu waa t	1	0	0	0		u2.0.
std::_Fac_tidy_reg_t::~_Fac_tidy_reg_t	1	0	0	0	0	exe
						beido u2.0.
scrt_is_ucrt_dll_in_use	3	0	0	0	0	exe
scrt_is_ucrt_uii_iii_use	3	0	0	0	0	beido
						u2.0.
scrt_is_user_matherr_present	1	0	0	0	0	exe
	_					beido
						u2.0.
_RTC_Initialize	1	0	0	0	0	exe
						beido
						u2.0.
_RTC_Terminate	1	0	0	0	0	exe
						beido
						u2.0.
scrt_initialize_mta	1	0	0	0	0	exe