# Computing Pi

## Problem

Calculate and output the first n digits of .

## Thinking

What I need:

1. A series to approximate

the convergence speed of the series is fast enough.

1. A data structure to compute the series

Int, double is not enough to show the result because of the appearance of a lot of big numbers in the computation.

1. A standard to judge when to stop in the computation

If we just computed the result to the n-th digit, we would never know whether the result is right. So we need to continue computing until we can make sure of the result.

## Algorithm

1. Series
2. How to compute
3. result =result\*10000;
4. item=result/10000 ( without decimal), we can get and store the first digit of the result;
5. remain= result%10000 ,we use the remain to express the decimal;
6. repeat the above process until we get all the N digits.

## Core of code

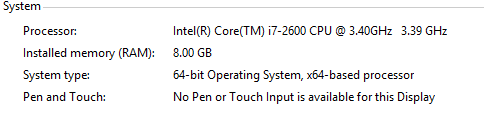
|  |
| --- |
| const int LEN = 4;  const int BASE = 10000; // BASE == 10 \*\* LEN  const int TIMES = 15;  double get\_item(double pi[], int n)  {  double item = 0;  for (int k = n - 1; k >= 0; k--) {  item += pi[k] \* BASE;  pi[k] = item - floor(item / (2 \* k + 1))\*(2 \* k + 1);  item = floor(item / (2 \* k + 1));  if (k > 0) item \*= k;  //detect overflow  if (item<0)  {  printf("overflow\n");  getchar();  return -1;  }  }  return item;  }  void main(int argc, char\* argv[])  {  int DIGITS;  cout << "input wanted digits:\n";  cin >> DIGITS;  double \*pi = new double[DIGITS / LEN \* TIMES];  int n = DIGITS / LEN \* TIMES;  double \*result = new double[DIGITS / LEN];  double remainder = 0;  for (int i = 0; i < n; i++)  pi[i] = 2;  int count = 0;  for (int i = 0; n>0; n -= TIMES, i++)  {  double item = get\_item(pi, n);  result[i] = remainder + floor(item / BASE);  if (result[i] > BASE)  {  result[i - 1] += floor(result[i] / BASE);  result[i] = result[i] - floor(result[i] / BASE)\*BASE;  }  if (i % 2 == 0)  cout << endl;  remainder = item - floor(item / BASE)\*BASE;  }  }   * 指定每次循环计算的位数 LEN = 4。 * 指定计算的基数 BASE ，这个基数是 10 的 LEN 次方。 * 指定要计算的圆周率的位数 DIGITS，必须是 LEN 倍数。 * 指定为了达到所需的精度，必须将上述公式迭代的次数是循环次数的多少倍: TIMES 。 * 定义数组 pi ，用来存放进位后的余数。以及数组 pi 的大小 n 。 * 将数组 pi 的值初始化为 2 ，因为所用级数中每次迭代都要加 2 。 * 主循环，每次循环输出 LEN 位圆周率的值。 |

## Unit test

1. Baseline

The first 1,000,000 digits of from Mathematical

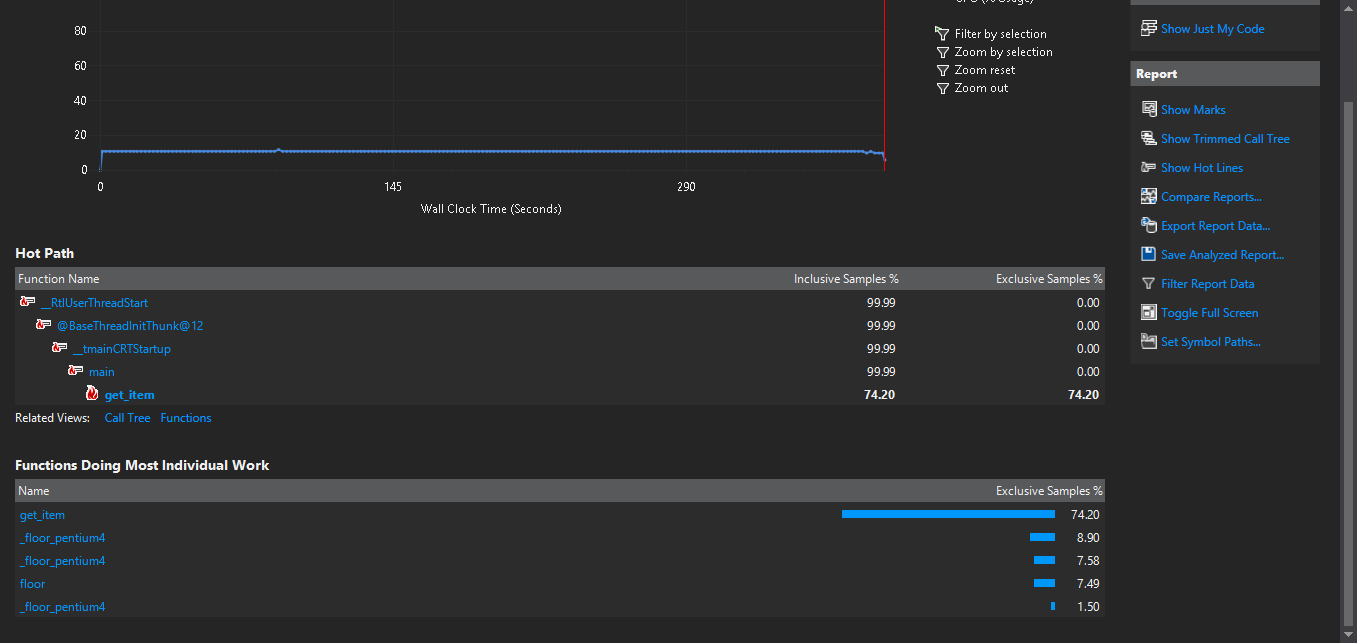
1. Computer



1. Test

|  |  |
| --- | --- |
| n | Time/sec |
| 10,000 | 4.204 |
| 30,000 | 14.498 |
| 50,000 | 27.692 |
| 100,000 | 92.404 |
| 500,000 | 2290.63 |

1. Performance analysis



The percentage of function **get\_item()** is 74.20%, most important.