

San José State University
Department of Computer Engineering

CMPE 180A

Data Structures and Algorithms in C++

Fall 2020
Instructor: Ron Mak

Assignment #1

Assigned: Tuesday, August 25
Due: Tuesday, September 1 at 5:30 PM
CodeCheck: <http://codecheck.it/files/20082407544wvyblyxihynvg7qdnc6pvfs0>
Note: Use Firefox or Chrome browsers, not Safari.
Canvas: Assignment #1: Pi Recipes
Points: 100

Pi recipes

This assignment will give you practice using arithmetic expressions, some C++ built-in math functions, and control statements to implement formulas that compute estimates of the value of pi (π). These formulas are by John Machin (1686-1751), Srinivasa Ramanujan (1887-1920), and the brothers David and Gregory Chudnovsky (born 1947 and 1952).



John Machin



Srinivasa Ramanujan



David and Gregory Chudnovsky

Your program will use each formula to compute an estimate of π to 16 decimal places. Therefore, your program should use the `double` datatype in its calculations.

At the above CodeCheck URL, complete the program **PiRecipes.cpp** in CodeCheck's edit box, and then press the "Submit" button. CodeCheck will compile and run your program and compare your output to a master report. You can type into CodeCheck's edit box directly, or you can first edit and test your program in an IDE such as Eclipse, and then cut and paste it into CodeCheck.

Choose descriptive variable names. Include meaningful comments in your code, but don't over-comment. Include your name in a comment at the top of your program.

See the tutorials at <http://www.cs.sjsu.edu/~mak/tutorials/index.html> for how to install and configure Eclipse for C++ development, and, if necessary, VirtualBox and Ubuntu.

Academic integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss (<http://theory.stanford.edu/~aiken/moss/>). **Copying another student's program or sharing your program is a violation of academic integrity.** Moss is not fooled by renaming variables, reformatting source code, or re-ordering functions.

Violators of academic integrity will suffer severe sanctions, including academic probation. Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.

The Ramanujan formulas

Ramanujan devised many formulas to compute estimates of π . The formulas for this assignment:

Ramanujan #0:

$$\pi = \frac{12}{\sqrt{190}} \ln[(2\sqrt{2} + \sqrt{10})(3 + \sqrt{10})]$$

where \ln is the natural logarithm function. Use the C++ built-in square root and natural logarithm functions.

Ramanujan #1:

$$\frac{1}{\pi} = \frac{\sqrt{8}}{99^2} \sum_{n=0}^{\infty} \frac{(4n)!}{(4^n n!)^4} \frac{1103 + 26390n}{99^{4n}}$$

where $n!$ (" n factorial") is $1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-2)(n-1)n$. For example, $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$. By definition, $0! = 1$.

The summation

$$\sum_{n=0}^{\infty} \frac{(4n)!}{(4^n n!)^4} \frac{1103 + 26390n}{99^{4n}}$$

implies a loop where n starts at 0 and increments by 1 for each iteration, theoretically up though infinity. For each iteration, generate a value by plugging the value of n into the formula, and add each generated value to a running sum that you've initialized to 0. The sum should converge towards $1/\pi$ (after you multiplied it by the constant that precedes the summation), which you must invert to get an estimate of π itself. Stop the iterations when two consecutive π estimates differ by less than 10^{-17} , which means you've attained 16 decimal digits. To prevent a runaway loop in case the sum fails to converge, you should also put a hard limit on the number of iterations, such as 20.

Ramanujan #2:

$$\frac{4}{\pi} = \frac{1}{882} \sum_{n=0}^{\infty} \frac{(-1)^n (4n)!}{(4^n n!)^4} \frac{1123 + 21460n}{882^{2n}}$$

What does $(-1)^n$ accomplish?

Chudnovsky:

$$\frac{1}{\pi} = 12 \sum_{n=0}^{\infty} \frac{(-1)^n (6n)!}{(3n)! (n!)^3} \frac{13591409 + 545140134n}{(640320^3)^{(n+\frac{1}{2})}}$$

Machin:

$$\frac{\pi}{4} = 4 \arctan \frac{1}{5} - \arctan \frac{1}{239}$$

where arctan is the arctangent function. Do not use the C++ built-in arctangent function. Instead, compute the arctangent using this Taylor series:

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Stop computing the series when the terms change the sum by less than 10^{-17} .

For the Ramanujan #1 and #2 and the Chudnovsky formulas, see <https://keisan.casio.com/exec/system/1355104874>. Your calculated values may differ from the webpage's values because the webpage calculates to a different number of decimal digits.

Note: The first 20 decimal digits of π are **3.14159 26535 89793 23846**
 In a later assignment, you will be able to compute a million or more decimal digits of π .

Expected output

Your program should generate output similar to this master output:

```
Ramanujan pi recipe #0:

  Estimate:  3.1415926535897936

Ramanujan pi recipe #1:

  Iteration  Estimate
    1      3.1415927300133055
    2      3.1415926535897936
    3      3.1415926535897931
    4      3.1415926535897931

Ramanujan pi recipe #2:

  Iteration  Estimate
    1      3.1415850400712375
    2      3.1415926535976215
    3      3.1415926535897931
    4      3.1415926535897931

Chudnovsky pi recipe:

  Iteration  Estimate
    1      3.1415926535897345
    2      3.1415926535897936
    3      3.1415926535897936

Machin pi recipe:

  Estimate:  3.1415926535897940
```

Useful online references

- <http://www.cplusplus.com/>
- <http://www.cplusplus.com/reference/cmath/>

Using CodeCheck

At the URL shown above, you will see an outline of the program solution. There are several places where you must complete the code, marked

```
/***** Complete this function. *****/
```

When you think you have a solution, press the “Submit” button at the bottom of the page. CodeCheck will run your program and compare its output to the master output. You can try as many times as you need to get satisfactory output. The number of tries will not affect your score.

CodeCheck does text comparisons of your program’s output with the master output. Because of roundoff errors and differences in the way you coded the formulas, do not be concerned if CodeCheck marks the last few digits of your estimates as “wrong” or you have a few more iterations. We will not deduct points unless your output is very different from the master output.

Submission into Canvas

When you’re satisfied with your program in CodeCheck, click the “Download Report” button at the very bottom of the page to download a signed zip file of your solution. Submit this signed zip file from CodeCheck into Canvas:

Assignment #1: Pi Recipes.

Note: You must submit the signed zip file that you download from CodeCheck, or your submission will not be graded. Do not rename the zip file.

Rubric

Your program will be graded according to these criteria:

Criteria	Maximum points
Good output (as determined by CodeCheck) <ul style="list-style-type: none">• Good output values.• Good output format.	30 <ul style="list-style-type: none">• 15• 15
Good program design <ul style="list-style-type: none">• Good use of arithmetic expressions.• Good use of control statements and loops.	40 <ul style="list-style-type: none">• 20• 20
Good program style <ul style="list-style-type: none">• Descriptive variable names.• Meaningful comments.	30 <ul style="list-style-type: none">• 15• 15