3.14159265358979323846264338327950288841

The Bangor Makerspace Coding Challenge

The Challenge: How quickly can you calculate 1,000 digits of π?

Prizes: Delicious pie, of course!

Requirements

Obviously, we expect you to program a computer to do it, and any language is permitted. Unusual choices are encouraged! There are two classes of entry: Conforming and Freestyle. Conforming entries must do the calculation using a specified method, given below, that is easy to understand using only basic arithmetic. Anything goes for Freestyle entries, with some common-sense restrictions. The answer must be correct – if it's not correct then you didn't calculate π.

Entries will be judged on speed: how long does it take to calculate 1,000 digits? The program should print out 1000 digits of π, after the decimal point, and (ideally) nothing else.

The judges reserve the right to disqualify any entry that they feel is not the work of the entrant. It must be *your* work. Don't copy things off the internet.

Regardless of the class of entry, your program – what *you* coded – must do the calculation. Don't simply rely on a canned calculator; for example, if you're using the computer algebra system, Maxima, then `bfloat(4*atan(1))` calculates π. That's not acceptable. On the other hand, many languages are able to do high-precision arithmetic, either with built-in features of the language or by loading an external library. For example, Java's `BigDecimal` class or Python's `decimal` module could be used for addition, subtraction, multiplication and division. That is acceptable.

All calculations must be done at run-time. Certain languages, like C, have a macro pre-processor that could be used to effectively do the calculation during the compilation stage. That is not acceptable, whether your program is Conforming or Freestyle. Conforming entries must do the calculation in a single thread.

All entries must include source code, and a discussion of how you performed the calculation, drawing attention to anything clever you've done, or how you've made use of special features of the implementation language. The explanation for a Freestyle entry must be complete and self-contained – don't merely point to a website or an academic paper – and be easily understood by an average college math major. The discussion document(s) must be provided in plain text, Markdown or LaTeX.

All materials must be placed in the public domain.

How to Do the Conforming Calculation

There are many ways to calculate π. Conforming entries must use

$$\begin{aligned}\pi \quad = \quad & \frac{3}{1} + \frac{3\cdot 2\cdot 1}{16\cdot 3\cdot 1} + \frac{3\cdot 2^2\cdot 1\cdot 3}{16^2\cdot 5\cdot 1\cdot 2} + \frac{3\cdot 2^3\cdot 1\cdot 3\cdot 5}{16^3\cdot 7\cdot 1\cdot 2\cdot 3} + \frac{3\cdot 2^4\cdot 1\cdot 3\cdot 5\cdot 7}{16^4\cdot 9\cdot 1\cdot 2\cdot 3\cdot 4} \\ & + \frac{3\cdot 2^5\cdot 1\cdot 3\cdot 5\cdot 7\cdot 9}{16^5\cdot 11\cdot 1\cdot 2\cdot 3\cdot 4\cdot 5} + \frac{3\cdot 2^6\cdot 1\cdot 3\cdot 5\cdot 7\cdot 9\cdot 11}{16^6\cdot 13\cdot 1\cdot 2\cdot 3\cdot 4\cdot 5\cdot 6} + \cdots\end{aligned}$$

This is merely a series of fractions; add them up and you get π. The more of these fractions in the total, the more accurate the estimate.

These fractions follow a pattern. In the numerator, there is always a 3, then multiply by a power of 2 (2, $2^2 = 4$, $2^3 = 8$, ...), then multiply by increasingly many of the odd numbers (1, $1 \cdot 3$, $1 \cdot 3 \cdot 5$, $1 \cdot 3 \cdot 5 \cdot 7$, ...). The denominator is similar; it has powers of 16 (16, $16^2 = 256$, $16^3 = 4096$, ...), then a single odd number (3, 5, 7, 9, ...), then a product of increasingly many numbers (1, $1 \cdot 2$, $1 \cdot 2 \cdot 3$, $1 \cdot 2 \cdot 3 \cdot 4$, ...).

Adding up fractions sounds easy, but these numerators and denominators get big, and you must be clever about how they're handled. For example, the final term given above is

$$\frac{3 \cdot 2^6 \cdot 1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot 11}{16^6 \cdot 13 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6} = \frac{1,995,840}{157,034,741,760} \approx 0.00001271.$$

Soon, the numbers involved will be too big for most languages to easily handle.

You are free to collect terms, look for common factors, or do anything else that the average high school student could understand. You may not use calculus or anything that falls under the broad category of "advanced number theory." The judges reserve the right to move a Conforming entry to the Freestyle category if they feel that you've violated the spirit of this requirement. Focus on programming, not advanced mathematics.

Some Background For the Curious

First, where did that formula for π given above come from? It's not important for the challenge, but it is equivalent to

$$\pi = \sum_{n=0}^{\infty} \frac{3 \cdot \binom{2n}{n}}{16^n (2n+1)},$$

and it is derived from the fact that

$$\sin\left(\frac{\pi}{6}\right) = \frac{1}{2}$$

and the series expansion for arcsin. Don't worry if that doesn't make sense to you. On the other hand, if you've had calculus, then it should be clear.

For those scratching their heads over the mysterious Σ and some kind of fraction-looking thing, here is an explanation. The "fraction-looking thing" is called the choice function. By definition,

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

The exclamation point is used to mean a product of integers, and it's called the *factorial*. So, $m!$ or "m factorial" is

$$m! = 1 \cdot 2 \cdot 3 \cdots (m-1) \cdot m,$$

where m is a positive whole number. It's not relevant to the π challenge, but it so happens that $\binom{n}{k}$ is equal to the number of ways in which one could choose k items among n possibilities. Given six jelly beans with different colors, the number of ways in which you could choose two of them is

$$\begin{aligned}\binom{6}{2} &= \frac{6!}{2!(6-2)!} \\ &= \frac{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6}{(1 \cdot 2)(1 \cdot 2 \cdot 3 \cdot 4)} \\ &= \frac{5 \cdot 6}{1 \cdot 2} \\ &= 15\end{aligned}$$

Going back to the formula for π, the Σ means to add the fractions up, and the $n = 0$ and ∞ above and below Σ mean to start with $n = 0$ and keep going "forever." What about the $\binom{2n}{n}$? We have

$$\binom{2n}{n} = \frac{1 \cdot 2 \cdot 3 \cdots (2n-1) \cdot (2n)}{n! \cdot (2n-n)!}.$$

Rearrange the numerator, putting the odd numbers with the odd numbers, and the even numbers with the even numbers:

$$\binom{2n}{n} = \frac{(1 \cdot 3 \cdot 5 \cdots (2n-1)) \cdot (2 \cdot 4 \cdot 6 \cdots (2n))}{n! \cdot n!}.$$

Since the even numbers are all divisible by 2, this becomes

$$\begin{aligned}\binom{2n}{n} &= \frac{(1 \cdot 3 \cdot 5 \cdots (2n-1)) \cdot 2^n \cdot (1 \cdot 2 \cdot 3 \cdot \cdots \cdot n)}{n! \cdot n!} \\ &= \frac{(1 \cdot 3 \cdot 5 \cdots (2n-1)) \cdot 2^n n!}{n! \cdot n!} \\ &= 2^n \frac{1 \cdot 3 \cdot 5 \cdots (2n-1)}{n!}.\end{aligned}$$

The Fine Print

The contest is open to residents of Maine. We need your name, a physical address, email, and phone number. Submit as many entries as you'd like. Use different algorithms or different languages – the more, the merrier!

This is meant to be in good fun, but we do want the contest to be fair. For that to be possible, we have to be able to run your entry on a standard machine. Although you need not prepare your entry on a Raspberry Pi 4, that is what we will use to time the entries. We encourage offbeat entries, but be prepared to help us install the relevant software.

Entries that use old or unusual hardware or software will be accepted in the Freestyle category, but it will be impossible for the judges to compare with other entries. Maybe there's an old TRS-80 in your basement, you want to use a programmable calculator, or you'd like to build a Raspberry Pi supercomputer. You could even build a machine that throws darts at a circular target, and work out π that way. Those are fun ideas, and we'd love to see what you've done!

Depending on the number and nature of the entries, the judges may choose to break the contest up into further sub-categories.

To place the materials in the public domain, each file must include the statement

> This is the sole work of [*your name*], and I place it in the public domain.

CONTACT

`events@bangormakerspace.org`

FOR INFORMATION ON HOW TO SUBMIT YOUR ENTRY.

DEADLINE: MARCH 31, 2022.

AWARDS AND PIE EATING: APRIL 17, 2022, IN HERMON.