

“Motivated” Numerical Science Problems: Problems, Solutions, Output from GPT4 + Wolfram Alpha and GPT4 + Code Interpreter

Scott Aaronson
University of Texas at Austin
aaronson@cs.utexas.edu

Ernest Davis
New York University
davise@cs.nyu.edu

August 15, 2023

This document includes 21 college-level numerical math and physics problems with problems, answers, and outputs from GPT4 with the Wolfram Alpha plug-in (henceforth GPT4+WA) and with the Code Interpreter plug-in (henceforeth GPT4+CI). The experiments on the two systems were carried out between July 27 and August ?? 2023.

For each problem, we give here:

1. The problem.
2. The correct solution, with an explanation of how it was derived. In some cases, we have included the Matlab code used to compute the correct solution.
3. The answer produced by GPT4+WA.
4. An edited form of the output of GPT4+WA. Editorial comments in this output are in *italics*.
5. A comment, including whether the answer was right, wrong, or got partial credit.
6. The answer produced by GPT4+CI.
7. In some cases, where it has features of interest, an edited form of the output of GPT4+CI. Editorial comments in this output are in *italics*.
8. A comment, as in (5).
9. Occasionally, we have also included the output of GPT-3.5 or GPT-4 without plug-ins, for comparison.

The excerpts from the outputs have been extracted unsystematically to show features of the systems reasoning that seemed to us interesting. The excerpts from GPT4+WA are mostly fairly complete; those from GPT4+CI are more brief, except where that had features of particular interest.

1. **Question:** If the earth collapsed to a black hole, how big would the black hole be? Please include all calculations.

Answer: 8.87 millimeters.

GPT4+WA Answer: 8.87 millimeters

GPT4+WA Calculation:

If the earth collapsed to a black hole, how big would the black hole be? Please include all calculations.

ChatGPT Used Wolfram

REQUEST TO WOLFRAM { "input": "Schwarzschild radius of Earth" }

If the Earth were to collapse into a black hole, the size of the black hole would be determined by the Schwarzschild radius. The Schwarzschild radius is the radius of the event horizon of a non-rotating black hole. It is given by the formula:

$$r = \frac{2GM}{c^2}$$

where:

r is the Schwarzschild radius (the size of the black hole),

G is the gravitational constant, approximately $6.674 \cdot 10^{11} \text{ m}^3/\text{kg} \cdot \text{s}$

M is the mass of the object, in this case, the Earth, which is approximately $5.97 \cdot 10^{24} \text{ kg}$.

c is the speed of light, approximately $2.998 \cdot 10^8 \text{ m/s}^2$.

Comment: Correct. A simple Google search gives the same answer, quoting an article from Forbes.

Google

X

Images
Bes
News
Videos
Books
Shopping
Maps
Flights
Finance

About 34,400,000 results (0.65 seconds)

After just a few minutes more — 21 to 22 minutes total — the entire mass of the Earth would have collapsed into a black hole just 1.75 centimeters (0.69") in diameter: the inevitable result of an Earth's mass worth of material collapsing into a black hole.

Oct 15, 2020

Forbes
<https://www.forbes.com/startswithabang/2020/10/15/> ::

GPT4+CI Answer: 8.87 millimeter. Calls CI to perform same calculation.

Comment: Correct.

GPT-3.5 with no plug-ins:

The Earth cannot collapse into a black hole under any realistic circumstances. Black holes form when massive stars undergo a gravitational collapse at the end of their life cycle. These stars need to have a significantly larger mass than that of the Earth to create a black hole.

For a black hole to form, the mass of the collapsing object needs to be compressed within a small enough volume, known as the Schwarzschild radius. For the Earth to become a black hole, it would need to be compressed to a sphere with a radius of about 9 millimeters (0.35 inches). This is known as the Schwarzschild radius for the mass of the Earth.

In reality, this collapse scenario is not possible because the Earth is held together by various forces, mainly the pressure resulting from nuclear reactions in its core and the electromagnetic forces between its atoms. It is not massive enough to overcome these forces and undergo gravitational collapse to form a black hole.

So, rest assured, the Earth will not collapse into a black hole anytime soon. It will continue to orbit the Sun as it has been for billions of years.

Comment: Correct, but somewhat misses the point.

2. If you fell into the black hole at the center of the Milky Way, how long would you have before hitting the singularity? Please include all calculations.

Correct Answer: 66.7 seconds.

GPT4+WA Answer: 66.7 seconds

GPT4+WA Output:

Used Wolfram

REQUEST TO WOLFRAM

{ “input”: “Schwarzschild radius of the black hole at the center of the Milky Way” }

The Schwarzschild radius of the black hole at the center of the Milky Way, known as Sagittarius A*, is approximately $1.273 \cdot 10^{10}$ meters or 12.73 million kilometers. This radius is also known as the event horizon, which is the point of no return for anything falling into a black hole.

The time it would take for an object to fall from the event horizon to the singularity of a black hole can be calculated using the formula for the proper time of free fall from the event horizon to the singularity of a Schwarzschild black hole. This formula is given by:

$$\tau = \frac{\pi r_s}{2c}$$

where

τ is the proper time experienced by the falling object,

r_s is the Schwarzschild radius (or the radius of the event horizon),

c is the speed of light.

Let's calculate this.

Used Wolfram

The time it would take for an object to fall from the event horizon to the singularity of the black hole at the center of the Milky Way, as experienced by the falling object, is approximately 66.7 seconds.

Comment: Correct.

GPT4+CI Answer: 63.44 seconds

GPT4+CI Calculations:

The time it would take for an object to fall into a black hole can be calculated using a simplified model, assuming a non-rotating, non-charged black hole (a Schwarzschild black hole) and neglecting any energy loss due to gravitational radiation. This is known as the proper time, the time experienced by the falling object. Note that this doesn't account for the extreme time dilation effects near the event horizon as observed by a distant observer.

Call to CI:

```
import math

# Constants
G = 6.67430e-11 # m^3 kg^-1 s^-2, gravitational constant
c = 2.998e8 # m/s, speed of light
M_sun = 1.9885e30 # kg, solar mass

# Mass of the black hole at the center of the Milky Way
M = 4.1e6 * M_sun # in kg
```

```

# Calculate the Schwarzschild radius
r_s = 2 * G * M / c**2 # in meters

# Calculate the proper time to fall from the event horizon to the singularity
tau = math.pi * r_s / (2 * c) # in seconds

```

Comment: The two answers differ by about 5%, presumably reflecting uncertainty in the mass of the black hole.

3. Consider the first binary black hole system discovered by LIGO. Roughly how close to that system would a person have to have been, before they were killed by the gravitational waves? Please include all calculations.

Answer: So close that you would be killed by the tidal forces rather than by the gravitational waves.

GPT4+WA Answer: The energy carried by gravitational waves is tiny. For instance, the gravitational waves detected by LIGO from the merger of two black holes about 1.3 billion light-years away caused a change in length of less than 1/1000 the diameter of a proton over a 4-km distance in the LIGO detectors.

The effect of gravitational waves is a stretching and squeezing of space, but this effect is incredibly small. It would not be harmful to a human body, which is already quite capable of withstanding much larger stretching and squeezing forces.

Even if you were close to the black holes when they merged, the gravitational waves themselves would not harm you. The intense gravity of the black holes, on the other hand, would be deadly long before you got close enough for the gravitational waves to have any significant effect.

To give you a sense of scale, the event horizon of a black hole (the point of no return) is typically a few kilometers for a stellar black hole. The gravitational waves produced by the merger of two black holes would have an amplitude orders of magnitude smaller than this, and the amplitude decreases with distance from the source.

So, to answer your question, a person would have to be incredibly close to the black hole merger for the gravitational waves to be lethal, so close in fact that the intense gravity of the black holes themselves would have already torn you apart. This is a process known as spaghettification, where the difference in gravitational pull between your head and feet (if you're falling in feet-first) stretches you out into a thin strand like a piece of spaghetti.

In conclusion, it's not the gravitational waves from a black hole merger that would kill you, but the intense gravity of the black holes themselves.

Comment: Correct. GPT did not call WA.

GPT4+CI: Similar answer. Again, GPT did not call CI.

4. How far off from $10!$ is Stirling's approximation? Please include all calculations.

Answer: 30,104.

GPT4+WA Answer: Approximately 30,100

GPT4+WA Output: *GPT4 had to call Wolfram Alpha 6 times: three successful calls to compute $10!$, the Stirling approximation, and the difference; and three unsuccessful calls attempting to compute the Stirling approximation.⁴*

GPT4+CI Answer: Approximately 30104.38.

```
import math
```

```

# Exact value of 10!
exact_10_factorial = math.factorial(10)

# Stirling's approximation for 10!
stirling_approximation = math.sqrt(2 * math.pi * 10) * ((10 / math.e) ** 10)

# Difference between the exact value and the approximation
difference = abs(exact_10_factorial - stirling_approximation)

exact_10_factorial, stirling_approximation, difference

```

RESULT

(3628800, 3598695.6187410373, 30104.381258962676)

Comment: Correct.

GPT-4 with no plug-ins Answer: 3,151,918.53

Comment: With no plug-ins, GPT-4 was able to set up the formulas correctly and to compute $10!$ (many times in the training set), but messed both in calculating the values of the Stirling approximation (it got a value of 476,881.471) and in doing the subtraction.

5. How high would an airplane have to be, before you could notice 10 degrees of the earth's curvature when looking out the window? Please include all calculations.

Answer: 98.4 km.

Matlab

```

r = 6378 % Radius of earth in kilometers
d = r/cos(10*pi/180) % distance from center of earth to airplane
answer = d-r %height above earth's surface

```

GPT4+WA Answer: 24 km

GPT4+WA Calculation:

To answer this question, we need to understand the geometry of the Earth and how the curvature is perceived from a certain height.

The Earth is approximately a sphere with a radius of about 6,371 kilometers. When you're at a certain height above the Earth, you can see a certain portion of the Earth's surface. The angle that this portion subtends at your eye is the angle of view.

The relationship between the angle of view (in radians), the radius of the Earth, and the height above the Earth is given by the formula:

$$\theta = 2 \cdot \arcsin \left(\frac{r}{r+h} \right)$$

where:

θ is the angle of view (in radians),

r is the radius of the Earth (in kilometers), and

h is the height above the Earth (in kilometers).

We want to find the height h when the angle of view is 10 degrees. We can rearrange the formula to solve for h :

$$h = \frac{r}{\sin\left(\frac{\theta}{2}\right)} - r$$

Let's plug in the values $r = 6371$ km and $\theta = 10$ degrees (which we need to convert to radians) into this formula to find h .

Note: To convert degrees to radians, we multiply by 180π . So $\theta = 10$ degrees is $\theta = 10\pi/180$ radians.

Let's do the calculation.

```
Request to Wolfram
{
  "input": "r = 6371; theta = 10 * Pi / 180; h = r / Sin[theta / 2] - r"
}
Response from Wolfram
"-6371 + 6371 Csc[Pi/36]"
```

The result of the calculation is given in a symbolic form. The function $\text{Csc}[x]$ is the reciprocal of the sine function, i.e., $\text{Csc}[x] = 1/\text{Sin}[x]$.

Let's evaluate this expression to get a numerical value for the height h .

```
Request to Wolfram
{
  "input": "h = -6371 + 6371 * Csc[Pi/36]; N[h]"
}
Response from Wolfram
"66728.02708816265"
```

The height above the Earth's surface at which you would notice 10 degrees of the Earth's curvature when looking out the window is approximately 66,728 kilometers.

Please note that this is a theoretical calculation and assumes a perfect spherical Earth, which is not entirely accurate due to the Earth's equatorial bulge and other factors. Also, this calculation assumes that the observer has an unobstructed view of the horizon, which might not be the case in a real-world scenario.

Comment: GPT invokes a completely wrong formula — the correct formula is $\theta = \arccos(r + h/r)$ — and gets a completely wrong answer.

GPT4+CI Answer: 96.3 km.

GPT4+CI Computation:

To determine the height at which an airplane would need to fly to observe a 10-degree arc of the Earth's curvature, we can use some basic geometry and the knowledge that the Earth is approximately a sphere.

The total angle for a sphere is 360° , and since we want to see 10° we'll use that fraction to determine the corresponding fraction of the Earth's circumference that we'll be able to see.

Let's denote:

R as the Earth's radius, approximately 6,371 km,

h as the height we are looking for,

θ as the angle of the Earth's curvature we want to see, which is 10 degrees.

We can use the formula for the length of an arc of a circle:

$$\text{Arc length} = R \cdot \theta$$

where θ is in radians. The arc length is also the distance of the line of sight from the airplane to the horizon, which forms a right triangle with the radius of the Earth and the line from the airplane straight down to Earth. Therefore, we can use the Pythagorean theorem:

$$(R + h)^2 = R^2 + \text{Arc length}^2$$

From this equation, we can solve for h . Let's do the calculations.

```
import math

# Define constants
R = 6371 # Earth's radius in km
theta_degrees = 10 # Angle of curvature in degrees

# Convert the angle to radians
theta_radians = math.radians(theta_degrees)

# Calculate the arc length
arc_length = R * theta_radians

# Use the Pythagorean theorem to solve for h
h = math.sqrt(R**2 + arc_length**2) - R

h
RESULT
```

96.30795394412053

The airplane would need to fly at an altitude of approximately 96.3 km (or about 60 miles) in order for a person to see a 10-degree arc of the Earth's curvature when looking out the window.

Please note that this is a theoretical answer. In reality, the Earth's atmosphere becomes extremely thin above about 100 km (known as the Krmn line), which is commonly used to define the boundary between Earth's atmosphere and outer space. Commercial airplanes

Comment: GPT4+CI is making a geometrical approximation: that the distance from the point on the horizon to the airplane is equal to the length of the curve on the earth's surface from the horizon to the point underneath the airplane. This is valid in the small angle limit, but here leads to a 2% error over all. Half credit.

6. Approximately how much time would a commercial airliner save in going from New York to Tel Aviv, if it could go in a straight line, through a tunnel in the earth, at the same speed as usual? Please include all calculations. Please include all calculations.

Answer: About 52 minutes, assuming a flight speed of 500 mph.

Matlab:

```
nyclat = (40+42/60+46/3600)*pi/180
nyclong = -(74+22/3600)*pi/180
talat = (32+8/60)*(pi/180)
talong = (34+78/60)*(pi/180)
earthRadius = 3959 % miles
```

```

nycDirection = [cos(nyclat)*sin(nyclong), cos(nyclat)*cos(nyclong), sin(nyclat)] % unit vector
taDirection = [cos(talat)*sin(talong), cos(talat)*cos(talong), sin(talat)]
tunnel = earthRadius*norm(taDirection-nycDirection)
flight = earthRadius*acos(dot(nycDirection,taDirection))
timeSaved = (flight-tunnel)/550 % assuming 550 mph.

```

GPT4+WA answer: About 49 minutes.⁴

GPT4+WA calculation: Three calls to Wolfram Alpha:

- Retrieve the distance from New York to Tel Aviv on the earth surface. The many measurements of this returned by Wolfram Alpha (kilometers, meters, light in fiber, light in vacuum etc.) include, helpfully, the facts that this is 0.23 of the earth's circumference and that the airplane travel time is 10 hours 20 minutes. circumference,
- Retrieve the earth's diameter
- Compute the straight line distance using the formula $d = 2r \sin(f * 2\pi)$ r is the radius of the earth and f is the fraction (0.23) of the earth's circumference of the path on the earth's surface.

Comment: Correct. (The difference in the answer is OK; it may well just correspond to different choices of reference points within New York and Tel Aviv.)

GPT4+CI Answer: The tunnel would save no time over the flight.

GPT4+CI Calculation:

```

import math

# Constants
r_earth = 6371 # Earth's radius in km
speed_plane = 900 # Plane speed in km/h

# Coordinates in degrees
lat1, lon1 = 40.7128, -74.0060 # New York
lat2, lon2 = 32.0853, 34.7818 # Tel Aviv

# Convert degrees to radians
lat1, lon1 = math.radians(lat1), math.radians(lon1)
lat2, lon2 = math.radians(lat2), math.radians(lon2)

# Haversine formula
dlat = lat2 - lat1
dlon = lon2 - lon1
a = math.sin(dlat/2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon/2)**2
c = 2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))
distance_straight = r_earth * c # Straight line distance in km

# Time to travel straight line distance
time_straight = distance_straight / speed_plane # Time in hours

# Real-world flight distance and time
distance_real = 9100 # Real-world flight distance in km
time_real = distance_real / speed_plane # Time in hours

```

```

# Time saved
time_saved = time_real - time_straight # Time saved in hours

distance_straight, time_straight, time_saved

```

The straight-line distance through the Earth between New York and Tel Aviv is approximately 9115 km. At a speed of 900 km/h, a plane would take about 10.13 hours to fly this distance. However, the real-world flight distance is approximately 9100 km, which also takes about 10.13 hours at the same speed.

So, surprisingly, in this case, there would be virtually no time saved by travelling through a straight tunnel through the Earth, assuming the same speed. In fact, the straight-line path is slightly longer than the real-world flight path due to the Earth's oblate spheroid shape, meaning it's wider at the equator than at the poles. This is a somewhat unusual situation and specific to the cities chosen for the example.

Comment: Wrong. GPT-4 tries to calculate the distance directly from the latitudes and longitudes of the two cities and somewhere messes up. GPT-4 does notice that the fact that the straight line distance is greater than the real-world flight path is "surprising" but doesn't realize that it is impossible, and makes up a bogus explanation about how this explained by the fact that the Earth is not a perfect sphere.

7. For what fraction of the lifetime of the universe has there been life in it? Give upper and lower bounds. Please include all calculations.

Answer: Between 25% and 99.93%.

Explanation: The age of the universe is taken to be 13.8 billion years. It is generally accepted that life emerged on earth at least 3.5 billion years ago, and it is speculated that it might have been possible for life to appear in the universe about 10 years after the Big Bang.

GPT4+WA Answer: Between 74% and 99.93%.

GPT4+WA Calculation: To answer this question, we need to know two things:

The age of the universe. The earliest possible time life could have appeared in the universe. As of my knowledge cutoff in September 2021, the age of the universe is estimated to be about 13.8 billion years. However, I can use the Wolfram Alpha tool to get the most recent estimate. Regarding the earliest possible time life could have appeared in the universe, this is a subject of ongoing scientific debate. Some theories suggest that life could have appeared as early as 10 million years after the Big Bang, while others suggest it could have taken billions of years. The earliest evidence of life on Earth dates back to about 3.5 billion years ago, but life could have potentially existed elsewhere in the universe before this time.

Let's first get the most recent estimate for the age of the universe.

```

Request to Wolfram
{
  "input": "age of the universe"
}
Response from Wolfram
Query:
"age of the universe"

```

Input interpretation:

Value:
13.8 billion years

The most recent estimate for the age of the universe is approximately 13.8 billion years.

Now, let's consider the earliest possible time life could have appeared in the universe. As I mentioned earlier, this is a subject of ongoing scientific debate. For the purpose of this calculation, I'll use two estimates:

The optimistic estimate: Life could have appeared as early as 10 million years after the Big Bang. This is based on theories suggesting that life could arise quickly under the right conditions. The conservative estimate: Life first appeared on Earth about 3.5 billion years ago. This is based on the earliest evidence of life we have found so far.

Let's calculate the fractions:

For the optimistic estimate, the fraction of the universe's lifetime that has contained life is $(13.8 \text{ billion years} - 10 \text{ million years}) / 13.8 \text{ billion years}$. For the conservative estimate, the fraction is $(13.8 \text{ billion years} - 3.5 \text{ billion years}) / 13.8 \text{ billion years}$.

Let's perform these calculations. Used Wolfram

```
Request to Wolfram
{
  "input": "optimisticFraction = (13.8*10^9 - 10*10^6) / (13.8*10^9);
  conservativeFraction = (13.8*10^9 - 3.5*10^9) / (13.8*10^9);
{optimisticFraction, conservativeFraction}"
}
Response from Wolfram
"0.9992753623188406, 0.746376811594203"
```

Based on these calculations:

The optimistic estimate suggests that life has existed for approximately 99.93% of the lifetime of the universe.

The conservative estimate suggests that life has existed for approximately 74.64% of the lifetime of the universe.

So, the fraction of the lifetime of the universe that has contained life lies somewhere between 74.64% and 99.93%, depending on when life first appeared.

Comment: The lower bound is miscalculated. Half credit.

GPT4+CI Answer: Between 25% and 31%.

GPT4+CI Computation: To estimate the fraction of the universe's lifetime during which life has existed, we first need to establish:

The age of the universe.

The earliest point at which life may have appeared on Earth.

The latest point at which life may have appeared on Earth.

As of my knowledge cutoff in September 2021, here is the information we have:

The age of the universe: The most recent estimate is about 13.8 billion years.

The earliest potential evidence for life on Earth: This is a bit more uncertain. Some controversial evidence might suggest life existed on Earth as early as 4.28 billion years ago, based on carbon isotopes found in ancient rocks.

More widely accepted evidence suggests that life on Earth existed around 3.5 billion years ago.

Please note that these are estimates and the actual values could be different. The uncertainties primarily come from the difficulty in interpreting ancient geological records, the complexity of biological evolution, and the cosmological measurements required to determine the age of the universe.

Let's calculate the fractions now.

```
# Variables
age_of_universe = 13.8 # in billion years
earliest_life = 4.28 # in billion years
accepted_life = 3.5 # in billion years

# Calculate fractions
earliest_fraction = earliest_life / age_of_universe
accepted_fraction = accepted_life / age_of_universe

earliest_fraction, accepted_fraction
```

RESULT

(0.3101449275362319, 0.25362318840579706)

The calculations suggest:

The earliest potential evidence for life on Earth indicates that life may have been present for about 31.0% of the universe's lifetime. More widely accepted evidence suggests that life has been present for about 25.4% of the universe's lifetime.

These percentages give us the lower and upper bounds, respectively, for the fraction of the universe's lifetime during which life has existed. Please note that these are estimates and the actual values could be different due to the reasons mentioned earlier.

Comment: GPT misinterpreted the question as “life on earth” rather than “life somewhere in the universe”. Half credit.

8. Approximately how long would it take to transmit an entire human genome over a standard WiFi connection? Please include all calculations.

Answer: 4 minutes.

Explanation: The human genome has about 3 billion base pairs = 6 billion bits. Assuming a 25Mbps WiFi connection, that gives 240 seconds = 4 minutes.

GPT4+WA Answer: 4 minutes

GPT4+WA Calculation: To answer this question, we need to know two things:

The size of the human genome in terms of digital storage.

The speed of a standard WiFi connection.

The human genome consists of about 3 billion base pairs. Each base pair can be one of four types (A, T, C, G), so we can represent each base pair with 2 bits of information (since 2 bits can represent 4 different states). Therefore, the size of the human genome in bits is $4 \cdot 10^9$ base pairs * 2 bits/base pair = $6 \cdot 10^9$. To convert this to gigabytes, we divide by $8 \cdot 10^9$ (since there are 8 bits in a byte and 10^9 bytes in a gigabyte), giving us approximately 0.75 gigabytes.

The speed of a standard WiFi connection can vary widely depending on many factors, but a typical speed might be around 25 Mbps (megabits per second). To convert this to gigabytes per second, we divide by $8 \cdot 10^9$ (since there are 8 bits in a byte and 10^3 megabits in a gigabit), giving us approximately 0.003125 gigabytes per second.

To find out how long it would take to transmit the human genome over this WiFi connection, we divide the size of the genome by the speed of the connection:

$$\frac{0.75\text{GB}}{0.003123\text{GB/s}} \approx 240\text{s} \approx 4 \text{ minutes}$$

So it would take approximately 4 minutes to transmit the entire human genome over a standard WiFi connection. Please note that this is a rough estimate and the actual time could be longer due to factors such as network congestion, signal strength, and other variables.

Comment: Right.

GPT4+CI Answer: 4 minutes.

GPT4+CI Calculation: Same as GPT4+WA.

Comment: Right.

9. Approximately how large would an asteroid have to be, in diameter (assume it's approximately spherical), before an Olympic high jumper could no longer reach escape velocity by jumping off it? Please include all calculations.

Answer About 10 km.

Explanation: The kinetic energy needed for escape velocity from the surface of a planet is inversely proportional to the radius and proportional to the mass of the planet, which in turn is proportional to the density and radius cubed. So over planets of constant density energy is proportional to the radius squared and density. So the velocity is proportional to the radius and square root of the density. So the radius is proportional to the velocity and inversely proportional to the square root of the density.

The world record for high jump is 2.45 meters, set by Javier Sotomayor in 1993. The jumper's center of mass starts about half way up his body, around .9 m. (Sotomayor is 193 cm tall.) At the apex of the jump, it is perhaps .3m above the bar. So the jump raises the center of mass by about 1.8 meters. The initial velocity is therefore $v = \sqrt{2gh} \approx 6 \text{ m/s}$.

Escape velocity from earth is 11 km/s and the diameter of earth is 12,750 km. The densities of asteroids is not at all well known (see the Wikipedia article, "Standard asteroid physical characteristics" but a reasonable estimate might be about 4/9 the density of earth.

So the diameter of the asteroid would be about $12750 \cdot (6/11000) \cdot (3/2) \approx 10 \text{ km}$.

GPT4+WA Answer: 10.81 km

GPT4+WA Calculation:

The escape velocity of an object is given by the formula

$$v_c = \sqrt{\frac{2GM}{r}}$$

where

v_c is the escape velocity

G is the gravitational constant ($6.67430 \cdot 10^{-11} \text{m}^2 \text{kg}^{-1} \text{s}^{-2}$)

M is the mass of the object (in this case, the asteroid), and

r is the radius of the object (half the diameter of the asteroid).

We can rearrange this formula to solve for the radius r :

$$r = \frac{2GM}{v_c^2}$$

The mass M of the asteroid can be expressed in terms of its volume and density. For a spherical asteroid, the volume V is given by $V = 4\pi r^3/3$.

Assuming the asteroid is made of a typical asteroid material such as rock, we can use a typical density ρ of about 3000kg/m^3 . Therefore, the mass M is $M = \rho V = \rho(4/3)\pi r^3$.

Substituting this into our formula for r , we get:

$$r = \frac{2G\rho\frac{4}{3}\pi r^3}{v_c^2}$$

Solving this equation for r will give us the radius of the asteroid for which the escape velocity is equal to the jump speed of an Olympic high jumper.

The world record for the high jump is about 2.45 m, which corresponds to a speed of about 7 m/s (assuming the jumper's center of mass starts from rest and reaches its maximum height at this speed due to the conversion of kinetic energy to potential energy).

Let's plug these values into the equation and solve for r .

```
Request to Wolfram
{
  "input": "G = 6.67430*10^-11; rho = 3000; ve = 7;
  Solve[2*G*rho*(4/3)*Pi*r^3 == ve^2*r, r]"
}
Response from Wolfram
"{{r -> -5404.7454967406165}, {r -> 0.}, {r -> 5404.7454967406165}}"
```

The solution to the equation gives three values for the radius r : -5404.75 m, 0 m, and 5404.75 m. We can disregard the negative value and the zero value as they are not physically meaningful in this context.

So, the radius of the asteroid for which the escape velocity is equal to the jump speed of an Olympic high jumper is approximately 5404.75 m.

Since the diameter is twice the radius, the diameter of the asteroid would be approximately $2 \cdot 5404.75 = 10809.49$ m or about 10.81 kilometers.

Comment: We count this as correct. GPT4 took the density to be 3000 kg/m^3 , which is reasonable. Its calculation of the jumper's velocity is based on the assumption that the center of mass rises by the entire 2.45 meters, ignoring the fact that it starts from about 1 meter off the ground. However, since the error thereby introduced is smaller than the uncertainty due to density, we'll give it a pass on that.

GPT4+CI Answer: 74.2 km

GPT4+CI Calculation: *It set up the calculation in the same way as GPT4+WA, and assumes the same value for density. However, it makes a mistake in manipulating the equation, as shown here:*

we can substitute M into the equation for v_{esc} to get:

$$v_{\text{esc}} = \sqrt{\frac{2G\rho \cdot \frac{4}{3}\pi r^3}{r}} = \sqrt{\frac{8}{3}\pi G\rho r^2}$$

Setting this equal to the initial speed of the jumper v_{jump} and solving for r , we find:

$$r = \frac{v_{\text{jump}}^2}{\sqrt{\frac{8}{3}\pi G\rho}}$$

which leads to an erroneous value.

Comment: Wrong, as discussed above.

10. What is the length of the $y = x^2$ parabola in the region $-1 \leq x \leq 1$? Please include all calculations.

Answer:

$$\int_{-1}^1 \sqrt{1+4x^2} dx = \sqrt{5} + \frac{1}{2} \sinh^{-1}(2) = 2.95789$$

GPT4+WA Answer: 2.95789

GPT4+WA Calculation: It set up the integral and output both the exact symbolic expression above and its numerical value.

GPT4+CI Answer: Approximately 2.958

GPT4+CI Calculation: It set up the integral and output the numerical value.

Comment: Both correct.

11. Approximately how large a supply of antimatter would be needed, in order to propel a space-craft with the mass of the International Space Station into orbit around Proxima Centauri, in one year as experienced by its crew? Please include all calculations.

Answer: 706,000 kg.

Explanation:

We'll assume that the Space Station is accelerated rapidly to its cruising speed, and we'll ignore the need to decelerate when it reaches Proxima Centauri.

The first step is to compute the velocity. We'll use the speed of light as the unit of speed and a year as the unit of time. Proxima Centauri is 4.25 light years away. If the spaceship travels at speed v , then as measured from earth, the time required $\tau = 4.25/v$. Because of time dilation, this appears to the crew as $\tau\sqrt{1-v^2}$. So we have the equation $1 = (4.25/v)\sqrt{1-v^2}$, so $1 = (4.25^2/v^2)(1-v^2)$, so $v^2 = 18.0625(1-v^2)$, so $v^2 = 18.0625/19.062 = 0.9475$ and $v = 0.9734$.

If the Space Station has mass M , then its total energy at speed v is $M/\sqrt{1-v^2}$, so the additional energy is $M/\sqrt{1-v^2} - M$ (c is still 1). If we annihilate mass m of antimatter with mass m of matter, then that liberates $2m$ of energy. We have $2m = M \cdot ((1/\sqrt{1-v^2}) - 1) = 3.3661M$ so $m = 1.683M$. Taking m to be 420,000 kg, that gives a value of 706,000 kg of antimatter.

GPT4+WA: This is impossible.

GPT4+WA Output:

This is a complex question that involves several steps to answer. Here's a general outline of the process:

Determine the mass of the International Space Station (ISS). Determine the distance to Proxima Centauri. Calculate the energy required to travel this distance in one year, assuming a constant acceleration journey (half the trip accelerating, half decelerating). Determine the energy produced by a unit of antimatter reacting with matter. Calculate the amount of antimatter needed to produce the required energy.

Let's start with the first two steps. I'll use Wolfram Alpha to find the mass of the ISS and the distance to Proxima Centauri.

```
Request to Wolfram
{
  "input": "mass of the International Space Station"
}
```

```
RESPONSE FROM WOLFRAM
Result:
990000 lb (pounds) (without fuel)
```

```
Unit conversions:
450000 kg (kilograms)
```

```
Request to Wolfram
{
  "input": "distance to Proxima Centauri"
}
Response from Wolfram
Query:
"distance to Proxima Centauri"
```

```
Input interpretation:
Proxima Centauri | distance from Earth
```

```
Result:
4.247 ly (light years)
```

The mass of the International Space Station (ISS) is approximately 450,000 kg, and the distance to Proxima Centauri is approximately $4.018 \cdot 10^{16}$ meters.

The next step is to calculate the energy required to travel this distance in one year, assuming a constant acceleration journey (half the trip accelerating, half decelerating). This is a relativistic problem, as the speeds involved are a significant fraction of the speed of light.

The relativistic rocket equation, which takes into account the effects of special relativity, can be used to calculate the energy required. The equation is:

$$\Delta v = c \cdot \tanh \left(\frac{Ft}{mc} \right)$$

where:

Δv is the change in velocity,
 c is the speed of light,
 \tanh^{-1} is the hyperbolic tangent function,
 F is the force,
 t is the time, and
 m is the mass of the rocket.

We can rearrange this equation to solve for F , the force required:

$$F = \frac{m \cdot c \cdot \tanh^{-1} \left(\frac{\Delta v}{c} \right)}{t}$$

We know that $\Delta v = 2 \cdot c$ (since we need to accelerate to near the speed of light and then decelerate from near the speed of light), $m = 450,000$ kg, $c = 3 \cdot 10^8$ m/s and $t = 0.5$ years $= 0.5 \cdot 365.25 \cdot 24 \cdot 60 \cdot 60$ s $= 0.5$ years $= 0.5365.25246060$ s (since we spend half the journey accelerating).

Let's calculate F .

```

Request to Wolfram
{
  "input": "m = 450000; c = 3*10^8; deltaV = 2*c; t = 0.5*365.25*24*60*60;
F = m*c*ArcTanh[deltaV/c]/t; F"
}
Response from Wolfram
"4.69974456137966 ^6 - 1.3439393624186315 ^7 I"

```

The result is a complex number, which indicates that we've run into a problem with our calculations. The issue is that we've assumed a change in velocity (ΔV) of $2c$, which is greater than the speed of light. This is not possible according to the laws of physics as we currently understand them.

Instead, let's aim for a maximum velocity of $0.99c$ during the journey. This means that the spaceship will accelerate to $0.99c$ and then decelerate from $0.99c$, so $\Delta v = 0.99c + 0.99c = 1.98c$. Let's recalculate the force required with this new Δv

```

Request to Wolfram
{
  "input": "deltaV = 1.98*c; F = m*c*ArcTanh[deltaV/c]/t; F"
}
Response from Wolfram
"((0.5560630039174163 - 1.5707963267948966 I) c m)/t"

```

Again, we've obtained a complex number, which indicates that we're still trying to achieve a change in velocity that's not possible according to the laws of physics. The issue is that the hyperbolic tangent function \tanh is only defined for inputs between -1 and 1, and we're trying to take the inverse hyperbolic tangent of a number greater than 1.

This suggests that even with a maximum velocity of $0.99c$, we can't make the journey to Proxima Centauri in one year as experienced by the crew of the spaceship. The journey would take longer than a year, or it would require a higher maximum velocity, which is not possible according to our current understanding of physics.

However, we can calculate the maximum velocity that would be achievable in one year of constant acceleration. The equation for the velocity of an object under constant acceleration is $v = at$, where a is the acceleration and t is the time. We can rearrange this to solve for a : $a = v/t$.

```

Request to Wolfram
{
  "input": "v = c; a = v/t; a"
}
Response from Wolfram
"c/t"

```

The acceleration required to reach the speed of light in half a year is c/t , where c is the speed of light and t is half a year.

However, it's important to note that reaching the speed of light is not possible according to our current understanding of physics. This calculation gives us an upper limit on the acceleration that we could theoretically achieve if there were no speed limit.

The force required to achieve this acceleration can be calculated using the equation $F = ma$, where m is the mass of the spaceship and a is the acceleration. Let's calculate the force.

```

Request to Wolfram
{
  "input": "F = m*a; F"
}
Response from Wolfram
"a m"

```

The force required to achieve the maximum possible acceleration is $F = ma$, where m is the mass of the spaceship and a is the acceleration.

The energy required to apply this force over the distance to Proxima Centauri can be calculated using the equation $E = Fd$, where E is the energy, F is the force, and d is the distance. Let's calculate the energy.

```

Request to Wolfram
{
  "input": "d = 4.018*10^16; E = F*d; E"
}
Response from Wolfram
"Your code made an assignment to the protected built-in symbol \"E\",
built-in symbols are protected and new values cannot be assigned to them."

```

Apologies for the confusion. I made a mistake in the code. The symbol "E" is a reserved symbol in the Wolfram Language, representing the base of the natural logarithm. I'll use a different symbol for energy and recalculate.

```

Request to Wolfram
{
  "input": "energy = F*d; energy"
}
Response from Wolfram
"d F"

```

The energy required to apply the maximum possible force over the distance to Proxima Centauri is $E = Fd$, where E is the energy, F is the force, and d is the distance.

The next step is to determine the energy produced by a unit of antimatter reacting with matter. When a particle of antimatter reacts with a particle of matter, they annihilate each other, producing energy according to the equation $E = mc^2$, where E is the energy, m is the mass of the particles, and c is the speed of light.

The energy produced by one kilogram of antimatter reacting with one kilogram of matter is therefore $E = (1kg + 1kg)c^2 = 2c^2J$. Let's calculate this.

```
Request to Wolfram
{
  "input": "m = 2; E = m*c^2; E"
}
Response from Wolfram
"Your code made an assignment to the protected built-in symbol \"E\",
built-in symbols are protected and new values cannot be assigned to them."
```

Apologies for the confusion. I made the same mistake again. I'll use a different symbol for energy and recalculate.

```
Request to Wolfram
{
  "input": "energyPerKg = m*c^2; energyPerKg"
}
Response from Wolfram
"c^2 m"
```

The energy produced by one kilogram of antimatter reacting with one kilogram of matter is $E = mc^2$, where E is the energy, m is the mass of the particles, and c is the speed of light.

The final step is to calculate the amount of antimatter needed to produce the required energy. This can be calculated using the equation $m = E/E_{per\ kg}$ where m is the mass of antimatter, E is the required energy, and $E_{per\ kg}$ is the energy produced per kilogram of antimatter. Let's calculate this.

Comment: Completely wrong. GPT4 starts out on the wrong track, not taking the time-dilation into account. This leads it to an impossible equation. It then first tries modifying its assumption, then changing the problem, and finally gives up in mid-calculation. One thing that is particularly strange here are the last three “successful” interactions with Wolfram Alpha which are completely useless, such as this one.

```
Request to Wolfram
{
  "input": "F = m*a; F"
}
Response from Wolfram
"a m"
```

GPT4+CI Answer: 676,200 kg

GPT4+CI Calculation:

We can start by calculating the energy required to reach Proxima Centauri within a year from the perspective of the spacecraft's crew. To do this, we'll need to take into account both the relativistic effects of time dilation and the energy required for propulsion.

First, we need to know the distance to Proxima Centauri. As of my last training data in September 2021, the closest star to the Sun is Proxima Centauri, which is approximately 4.22 light-years away.

Second, we need to consider the relativistic time dilation. If the crew of the spacecraft experiences the journey as taking one year, they will be traveling at a speed where time dilation is significant. This means we need to solve the Lorentz transformation for time dilation, which is given by:

$$\Delta t' = \frac{\Delta t}{\sqrt{1 - \frac{v^2}{c^2}}}$$

where

$\Delta t'$ is the dilated time, which is one year for the crew,
 Δt is the time in the stationary frame, which we want to calculate,
 v is the velocity of the spacecraft, and
 c is the speed of light.

Third, we need to calculate the kinetic energy required to reach this speed, which can be done using the relativistic kinetic energy formula:

$$KE = mc^2 \left(\frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} - 1 \right)$$

where:

KE is the kinetic energy,
 m is the mass of the spacecraft, and
 c is the speed of light.

Finally, we need to know the energy yield of antimatter. When a particle of antimatter comes into contact with a particle of matter, they annihilate each other, releasing energy equivalent to twice the mass of the particle (once for the matter and once for the antimatter) times the speed of light squared, according to Einstein's mass-energy equivalence principle $E = mc^2$. This means that to calculate the amount of antimatter required, we need to divide the total energy required by the energy yield per kilogram of antimatter, which is $2c^2$.

Let's start with the first step and solve the Lorentz transformation for time dilation to find Δt , the time in the stationary frame, in years. Then we'll use this to calculate the velocity v . After that, we can calculate the kinetic energy and then the amount of antimatter required. We'll assume the mass of the International Space Station (ISS) to be approximately 420,000 kg.

Calls CI:

```
import sympy as sp

# Constants

c = 299792458 # speed of light in m/s
mass_iss = 420000 # mass of the ISS in kg
distance_to_proxima = 4.22 # distance to Proxima Centauri in light-years
```

```

delta_t_prime = 1 # time experienced by the crew in years

# Solve Lorentz transformation for time dilation to find delta_t
v = sp.symbols('v')
delta_t = delta_t_prime / sp.sqrt(1 - v**2/c**2)
delta_t_solution = sp.solve(delta_t - distance_to_proxima, v)

# Only one solution will be real and less than c, that's the one we want
v_solution = [sol.evalf() for sol in delta_t_solution if sol.is_real]

# Calculate the kinetic energy
KE = mass_iss*c**2*(1/sp.sqrt(1 - v_solution**2/c**2) - 1)

# Calculate the amount of antimatter required
antimatter_required = KE / (2*c**2)

v_solution, KE, antimatter_required

```

RESULT

```
(-291253692.018633, 1.21547650372367e+23, 676199.999999999)
```

It seems like I've made a mistake in my calculation, as the velocity came out to be negative. My apologies for that. The issue might be due to the way the equation is being solved. Let's try again by using a different approach to solve for the velocity v .

The Lorentz transformation for time dilation can also be rearranged to solve for v as:

$$v = c \sqrt{1 - \left(\frac{\Delta t}{\Delta t'}\right)^2}$$

Let's use this formula instead to calculate v . Then, we'll proceed to calculate the kinetic energy and the amount of antimatter required as before.

Call to CI:

```

# Calculate v using the rearranged Lorentz transformation
v_solution = c * sp.sqrt(1 - (delta_t_prime/distance_to_proxima)**2)

# Calculate the kinetic energy
KE = mass_iss*c**2*(1/sp.sqrt(1 - v_solution**2/c**2) - 1)

# Calculate the amount of antimatter required
antimatter_required = KE / (2*c**2)

v_solution.evalf(), KE.evalf(), antimatter_required.evalf()

RESULT

```

```
(291253692.018633, 1.21547650372367e+23, 676199.999999999)
```

The velocity required for the spacecraft to reach Proxima Centauri in one year (as experienced by its crew) is approximately $12.91 \cdot 10^8$ m/s. This is just slightly below the speed of light, as expected due to the significant time dilation. t

The kinetic energy required to reach this velocity is approximately $1.22 \cdot 10^{23}$ Joules.

To provide this energy, approximately 676,200 kg of antimatter would be required. This is assuming perfect efficiency in the antimatter-matter reaction and the conversion of the resulting energy into kinetic energy for the spacecraft, which is highly unlikely in reality. In a real-world scenario, the amount of antimatter required would likely be significantly higher due to energy losses in the propulsion system and the need for extra energy to decelerate the spacecraft once it reaches Proxima Centauri.

Comment: GPT4+CI almost got this right. It correctly accounted for the time dilation. However, it failed to correct for the fact that, since the spaceship is travelling at less than the speed of light, the travel time for an earth observer is more than 1 year. Therefore, the time dilation is greater than GPT computed and thus the speed is faster and more antimatter is needed. In the end, the relative error is about 4%.

The difference between the two systems here is more likely to be chance behavior of GPT than an actual difference between the two plug-ins and their interfaces.

12. Approximately how many errors will a standard laptop suffer over its lifetime, due to cosmic rays hitting the microchip? Please include all calculations.

Answer: Estimates vary widely, but one commonly cited figure is about 1 error per 256 megabytes of RAM per month. Assuming an 8 GByte laptop and a 5 year lifespan, that will total 1920 errors.

GPT4+WA Answer:

GPT+WA Explanation:

REQUEST TO WOLFRAM

```
{ "input": "cosmic ray flux at sea level" }
```

RESPONSE FROM WOLFRAM

Wolfram|Alpha could not understand: cosmic ray flux at sea level.

Things to try instead: cosmic ray sea level

REQUEST TO WOLFRAM

```
{ "input": "cosmic ray intensity at sea level" }
```

RESPONSE FROM WOLFRAM

Wolfram|Alpha could not understand: cosmic ray intensity at sea level.

Things to try instead: cosmic ray intensity sea level

I apologize for the inconvenience. It seems that I'm having difficulty retrieving the specific data on cosmic ray flux at sea level from my usual sources. However, I can provide a general explanation and calculation based on typical values used in the field.

Cosmic rays are high-energy particles from space that can cause errors in electronic devices when they strike a microchip. This is known as a single event upset (SEU). The rate of SEUs depends on many factors, including the altitude (cosmic ray intensity increases with altitude), the shielding provided by the device's casing, and the design of the microchip itself.

A commonly used estimate for the flux of cosmic rays at sea level is about 1 cosmic ray per square centimeter per minute. This is a rough estimate and the actual value can vary.

Let's consider a microchip that has an area of 1 square centimeter (this is on the larger side for a microchip, but it makes the calculation easier). This means that it will be hit by approximately 1 cosmic ray per minute.

Not every cosmic ray hit will cause an error. The probability of an error occurring from a single hit depends on the design of the microchip. A commonly used estimate is that 1 in every 1000 hits will cause an error.

Let's also assume that the laptop is used for 8 hours a day.

So, the number of errors per day due to cosmic rays would be:

$$1 \text{ cosmic ray/minute} * 60 \text{ minutes/hour} * 8 \text{ hours/day} * (1 \text{ error} / 1000 \text{ cosmic rays}) = 0.48 \text{ errors/day}$$

If we assume a laptop has a lifespan of about 5 years, the total number of errors over its lifetime would be:

$$0.48 \text{ errors/day} * 365 \text{ days/year} * 5 \text{ years} = 876 \text{ errors}$$

This is a rough estimate and the actual number could be higher or lower depending on the factors I mentioned earlier. It's also worth noting that many of these errors will be corrected by error-checking mechanisms in the computer, so they won't all lead to noticeable problems.

Comment: Correct. A completely different approach to estimation gave a significantly different result, but there is no reason to suppose that this is less accurate. Note that the calls to WA were useless; the work was all done by GPT4.

GPT4+CI Answer: 1920 errors.

GPT4+CI Calculation: The error rate due to cosmic rays (also known as soft errors) is largely dependent on the specific microchip, its size, the altitude and geographical location where the device is used, and the shielding provided by the device. However, we can make a rough estimation using some average values.

...

Cosmic rays can cause soft errors by creating a high-energy particle shower when they strike the Earth's atmosphere. These high-energy particles can then strike a microchip, causing a bit to flip from a 0 to a 1 or vice versa.

The average error rate due to cosmic rays is typically cited as 1 error per 256 megabytes of RAM per month at sea level. However, this number can vary widely. For example, at higher altitudes or in space, the error rate can be much higher due to the increased intensity of cosmic rays.

Let's take the following assumptions for our calculation:

Average laptop lifespan: 5 years

Average laptop RAM: 8 gigabytes (GB)

Error rate: 1 error per 256 megabytes of RAM per month at sea level

The laptop is used at sea level

Note: Here we are only considering errors in RAM due to cosmic rays. The error rate might be different for other parts of a microchip, such as the CPU or GPU.

Now, let's calculate the number of errors:

```
pythonCopy code
# Constants
average_laptop_lifespan_years = 5
average_laptop_ram_gb = 8
error_rate_per_256mb_per_month = 1

# Conversion factors
```

```
gb_to_mb = 1024
months_per_year = 12

# Calculate total memory in MB
total_memory_mb = average_laptop_ram_gb * gb_to_mb

# Calculate error rate for the total memory
error_rate_per_month = error_rate_per_256mb_per_month * (total_memory_mb / 256)

# Calculate total errors over the laptop's lifespan
total_errors = error_rate_per_month * average_laptop_lifespan_years *
months_per_year
total_errors

RESULT
```

RESULT

1920.0

Comment: Correct.

13. What is the approximate probability that a randomly-chosen 100-digit integer is prime? Please include all calculations.

Answer: Approximately $1 / \log_e(5.5 \cdot 10^{99}) = 1 / (\log_e(5.5) + 99 \cdot \log_e(10)) = 0.00435$. That should be accurate to within 1%.

GPT4+WA Answer: $1.01 \cdot 10^{-22}$

GPT4+WA Computation: The probability that a randomly chosen n -digit number is prime can be approximated using the Prime Number Theorem, which states that the density of primes around a large number n is approximately $1/\ln(n)$, where \ln is the natural logarithm.

A 100-digit number is between 10^{99} and 10^{100} . So, we can estimate the number of 100-digit primes by subtracting the number of primes less than 10^{99} from the number of primes less than 10^{100} .

The Prime Number Theorem gives us an approximation for the number of primes i less than n as $n/\ln(n)$. So, the number of primes less than 10^{100} is approximately $10^{100}/\ln(10^{100})$ and the number of primes less than 10^{99} is approximately $10^{99}/\ln(10^{99})$.

The total number of 100-digit numbers is $10^{100} - 10^{99}$.

So, the probability P that a randomly chosen 100-digit number is prime is approximately:

$$P = \frac{10^{100}/\ln(10^{100}) - 10^{99}/\ln(10^{99})}{10^{100} - 10^{99}}$$

Let's calculate this.

The probability that a randomly chosen 100-digit number is prime is approximately $1.01 \cdot 10^{-22}$. This is a very small probability, indicating that it is extremely unlikely for a randomly chosen 100-digit number to be prime.

Comment: Wrong. GPT passed a correct expression to WA, but WA returned an answer in the form of an arithmetic expression over 100 digit numbers. (I presume correct, but I haven't checked.) Not surprisingly, GPT had trouble dealing with that.

GPT4+CI Answer: 0.00435705463248646

GPT4+CI Calculation: GPT-4 set up the following Python to execute:

```
import sympy as sp

# Define the variable
x = sp.symbols('x')

# Define the limits
lower_limit = 10**99
upper_limit = 10**100

# The number of 100-digit numbers
total_numbers = upper_limit - lower_limit

# The probability density function
pdf = 1/sp.log(x)

# The number of primes is approximated by the integral of the pdf
num_primes = sp.integrate(pdf, (x, lower_limit, upper_limit))

# The probability that a 100-digit number is prime
probability = num_primes / total_numbers

probability.evalf()
```

Comment: Correct, to many more decimal places than our crude estimate. By accident, we ran it the same problem a second time with GPT4+CI. On this second run, it got the analysis nearly correct, but made the mistake of assuming that $\log(10)=1$, so its answer was off by a factor of 2.3.

14. What is the probability that a randomly-chosen 100×100 matrix, over the finite field F_2 , is invertible? Please include all calculations.

Answer: 0.289

Explanation: Calculate the probability that the 2nd row is linearly independent from the first, then the probability that the 3rd row is outside the span of the first two, the probability that the 4th row is outside the span of the first three, etc. and multiplying them all together. In backward order this is

$$(1/2) * (3/4) * (7/8) * (15/16) * \dots \approx 0.289.$$

GPT4+WA answer: $2.58 \cdot 10^{301} / 1.58 \cdot 10^{302}$

GPT4+WA computation:

```
Request to Wolfram
{
  "input": "n = 100; q = 2; prob = Product[(q^n - q^k)/(q^n), {k, 0, n - 1}]; prob"
}
Response from Wolfram
```

Incredibly, Wolfram Alpha responded with the *exact fraction*, in the form of a 1521 digit numerator over a 1522 digit denominator.

GPT continued: The probability that a randomly-chosen 100x100 matrix, over the finite field F_2 , is invertible is approximately $4.59 \cdot 10^{301} / 1.59 \cdot 10^{302}$.

This is a very small number, close to zero. The exact value is too long to be displayed in its entirety, but it's a fraction with a numerator of approximately $4.59 \cdot 10^{301}$ and a denominator of approximately $1.59 \cdot 10^{302}$.

The calculation is based on the formula for the probability that a randomly chosen square matrix over a finite field is invertible. The formula is

$$\prod_{k=0}^{n-1} \frac{q^n - q^k}{q^n}$$

Comment: This is in fact the correct value, but we deduct 2 points out of 10 for the comment that it is a very small number. Why GPT turns the 1521/1522 digit numbers generated by WA into 301/302 digit numbers is mystifying. One can sympathize with GPT not wishing to count 1522 digits — counting is not its strong suit in any case — but since it didn't count, how did it manage to correctly determine that the denominator has one more digit than the numerator?

GPT4+CI answer: Approximately 0.289 or 28.9%.

GPT4+CI Computation:

To determine the number of invertible matrices, we need to count the number of sets of 100 linearly independent rows. The first row can be any nonzero row, of which there are $3^{100} - 1$ (subtracting 1 because the zero row is not allowed). The second row can be any row not in the span of the first row. Since the first row spans a 1-dimensional subspace of F_2^{100} there are $2^{100} - 2^1$ possibilities for the second row. The third row can be any row not in the span of the first two rows. Since the first two rows span a 2-dimensional subspace, there are $2^{100} - 2^2$ possibilities for the third row. This process continues until the 100th row, for which there are $2^{100} - 2^{99}$ possibilities.

The total number of 100×100 matrices is $2^{100 \times 100}$, since each entry can be 0 or 1. Therefore, the probability that a randomly chosen matrix is invertible is given by

$$\frac{(2^{100} - 1) \cdot (2^{100} - 2) \cdot \dots \cdot (2^{100} - 2^{99})}{2^{100 \cdot 100}}$$

Call to CI

```
import sympy as sp
```

```

numerator = sp.Mul(*[(2**100 - 2**i) for i in range(100)])
denominator = 2**100*100

probability = numerator / denominator
probability.evalf()

```

RESULT

0.288788095086602

The probability that a randomly chosen 100×100 matrix over the field F_2 is invertible is approximately 0.289, or 28.9%.

Comment: Right.

15. How does the total weight of all the uranium that humans mined, compare to the total weight of all the gold that they've mined? Please include all calculations.

Answer: It is estimated that humans have mined about 209,000 tones of gold and about 2.8 million tons of uranium, a difference of a factor of about 13.

Explanation: The figure for gold comes from the World Gold Council. The figure for uranium was estimated by the authors based on the information on this web page of the World Nuclear Association

GPT4+WA Answer: Around 197,576 tonnes of gold and more than 2.5 million tonnes of uranium.

GPT4+WA Output:

GPT called WA three times to get the figure for uranium and three times for gold, but did not get any useful answers:

REQUEST TO WOLFRAM

```
{ "input": "total amount of uranium mined" }
```

RESPONSE FROM WOLFRAM

Wolfram|Alpha could not understand: total amount of uranium mined. You could instead try: amount of uranium

REQUEST TO WOLFRAM

```
{ "input": "amount of uranium mined" }
```

RESPONSE FROM WOLFRAM

Wolfram|Alpha could not understand: amount of uranium mined. You could instead try: amount of uranium

REQUEST TO WOLFRAM

```
{ "input": "amount of uranium" }
```

Long response from WA with all kinds of facts and figures about uranium, but not the amount that has been mined.

I'm sorry, but it seems that the exact amount of uranium mined by humans is not readily available. This could be due to the fact that uranium mining data is often kept confidential for national security reasons. However, we can still try to find the total amount of gold mined by humans for comparison. Let's proceed with that.

Three similar calls to WA for gold, with similar outcomes.

I'm sorry, but it seems that the exact amount of gold mined by humans is also not readily available. This could be due to the fact that gold mining data is often kept confidential for economic reasons.

However, according to the World Gold Council, as of 2021, it is estimated that around 197,576 tonnes of gold has been mined throughout history. This is a rough estimate and the actual number could be slightly higher or lower.

As for uranium, the World Nuclear Association reported that by the end of 2019, cumulative production was more than 2.5 million tonnes.

Please note that these figures are estimates and the actual amounts may vary. The data for uranium is particularly sensitive and may not be fully disclosed due to its use in nuclear energy and weapons.

Comment: Correct.

GPT4+CI Answer:

First, we need to establish the total amount of each element that has been mined.

As of my knowledge cutoff in September 2021, the total amount of gold ever mined by humans was estimated to be around 190,000 metric tonnes (or 190 million kilograms) by the World Gold Council.

The total amount of uranium mined by humans is a bit trickier to pin down. According to the World Nuclear Association, as of 2019, approximately 2.5 million metric tonnes of uranium had been mined in the history of nuclear power.

Let's calculate the total weight of all the uranium and gold mined by humans.

```
pythonCopy code
# Total weight of gold and uranium mined (in kg)
gold_weight_kg = 190_000_000 # kg
uranium_weight_kg = 2_500_000_000 # kg

gold_weight_kg, uranium_weight_kg
```

RESULT

(190000000, 2500000000)

The total weight of all the gold ever mined is approximately 190,000,000 kg (190,000 metric tonnes), while the total weight of all the uranium ever mined is approximately 2,500,000,000 kg (2.5 million metric tonnes).

To compare these amounts, let's calculate the ratio of the weight of uranium mined to the weight of gold mined.

```
pythonCopy code
# Ratio of the weight of uranium mined to the weight of gold mined
uranium_to_gold_ratio = uranium_weight_kg / gold_weight_kg

uranium_to_gold_ratio
```

RESULT

13.157894736842104

The ratio of the weight of uranium mined to the weight of gold mined is approximately 13.16. This means that for every kilogram of gold mined, approximately 13.16 kilograms of uranium have been mined.

Therefore, humans have mined significantly more uranium than gold by weight

Comment: Correct. The first call to Python, in which the output simply echoes the input, is rather strange.

16. What is the Shannon entropy of a positive integer n that's chosen with probability $Pr[n] = 6/(\pi^2 * n^2)$? Please include all calculations.

Answer: 2.362

Matlab:

```
ent=0;
for i=1:1000000
    p = 6/((pi^2)*(i^2));
    ent = ent - p*log(p);
    sum = sum+p;
end
ent/log(2)

% It is easily shown that the sum of the remaining terms is less than 0.001.
```

GP4T+WA Answer: Unable to compute an answer.

GPT4+WA Computation: GPT set up the summation correctly, but Wolfram Alpha was unable to evaluate it:

```
Request to Wolfram
{
    "input": "-Sum[6/(Pi^2 * n^2) * Log2[6/(Pi^2 * n^2)], {n, 1, Infinity}]"
}
Response from Wolfram
"-Sum[(6 Log[6/(n^2 Pi^2)])/(n^2 Pi^2 Log[2]), {n, 1, Infinity}]"
```

GPT then tried an incorrect rewriting

$$H(N) = - \sum_{n=1}^{\infty} Pr[n] \log_2 Pr[n] = - \left(\sum_{n=1}^{\infty} Pr[n] \right) \left(\sum_{n=1}^{\infty} \log_2 Pr[n] \right)$$

However it found that Wolfram Alpha could not compute the second sum.

It concluded “In such cases, it might be necessary to use numerical methods or software specifically designed for handling complex infinite series to get an approximate value.”

Comment: Unable to find an exact closed-form, GPT4+WA did not consider computing a numerical approximation.

GPT4+CI Answer: 0.92653.

GPT4+CI Computation: GPT4 first set up the summation and passed it to sympy, with the specification that it should first compute a closed form expression and then evaluate it.

```

import sympy as sp

# define the variable
n = sp.symbols('n')

# define the probability
pr_n = 6/(sp.pi**2 * n**2)

# define the entropy
entropy = -sp.summation(pr_n * sp.log(pr_n, 2), (n, 1, sp.oo))

entropy.evalf()

```

However, that failed since there is no closed form expression.

GPT-4 then correctly rewrote the expression into the form

$$\log_2(6/\pi^2) + \sum_{n=1}^{\infty} 2PR[n] \log(n)$$

and passed that to Python, but messed up in writing the code, and so got an unevaluated expression returned

Call to CI:

```

# first part of the entropy
entropy_part1 = sp.log(pr_n, 2)

# second part of the entropy
entropy_part2 = 2*sp.summation(pr_n * sp.log(n, 2), (n, 1, sp.oo))

entropy = entropy_part1 + entropy_part2

entropy_part1.evalf(), entropy_part2.evalf(), entropy.evalf()

RESULT

(1.44269504088896*log(6/(pi**2*n**2)),
 1.64455979647526,
 1.44269504088896*log(6/(pi**2*n**2)) + 2.0*Sum(6*log(n)/(pi**2*n**2*log(2)), (n, 1, oo)))

```

GPT's comment about this is rather confused:

It appears that Sympy was unable to calculate the first part of the entropy expression, $\log(6/\pi^2 n^2)$ (as a numerical value. This is because n is a variable that ranges over all positive integers, so Sympy is trying to substitute this infinite sum into the logarithm, which is not computationally feasible.

GPT then corrects the computation:

Call to CI:

```

# first part of the entropy
entropy_part1 = sp.log(6/sp.pi**2, 2)

# total entropy

```

```

entropy = entropy_part1 + entropy_part2

entropy_part1.evalf(), entropy.evalf()

```

The first part of the entropy, $\log(6/\pi^2)$ is approximately -0.71803 .

Adding this to the second part of the entropy, which is approximately 1.64456 , we obtain a total Shannon entropy of approximately 0.92653 for a positive integer n chosen with the probability $\Pr[n]=6/\pi^2 n^n$.

Comment: Wrong. In its final computation, GPT4+CI fails to take the negative of the term $\log(6/\pi^2)$. Otherwise, the computation is correct.

Revised version of problem: On consideration, after running the above, we thought that perhaps the wording of the problem was unfair to GPT. After all, the answer probably has no elegant closed-form expression, and it is not easy to get a highly precise numerical answer (the convergence rate of the naïve summation is $O(\log(n)/n)$). So we tried the two systems with the following revised question:

Compute the Shannon entropy of a positive integer n that's chosen with probability $\Pr[n] = 6/(\pi^2 * n^2)$ to 3 digit accuracy. Please include all calculations.

GPT4+WA Answer: Approximately 2.36.

GPT4+WA Calculation: GPT converted this into the Wolfram Alpha call,

```

Request to Wolfram
{
  "input": "N[Sum[-6/(Pi^2 * n^2) * Log2[6/(Pi^2 * n^2)], {n, 1, Infinity}], 3]"
}
Response from Wolfram
"2.3625895544553516271`3."

```

and returned the correct answer “approximately 2.36 (to three decimal places).”

Comment: Correct. GPT is taking advantage of a built-in feature in Wolfram Alpha for carrying out computations to a specified precision.

GPT4+CI Answer: The entropy is infinite.

GPT4+CI Calculation: GPT calls CI in an attempt to compute the expression directly, which fails. As in the attempt described above, it then splits the summation into two parts which it tries to simplify. Unfortunately in the course of simplification it carries out the incorrect step:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} \log_2(n^2) = 2 \sum_{n=1}^{\infty} \frac{1}{n}$$

which diverges.

Comment: Wrong, due to a faulty symbolic manipulation.

17. Assume that IQs are normally distributed, with a mean of 100 and a standard deviation of 15. For which n does the number of people with IQ n exceed the number of people with IQ $n+1$ by the maximum amount? Please include all calculations.

Answer: Approximately $n = 14.5$

Explanation: Let $f(x)$ be the probability density function for the normal distribution with mean 100 and standard deviation 15.

$$f(x) = \frac{1}{\sqrt{2\pi} \cdot 15} \exp(-(x - 100)^2 / 2 \cdot 15^2)$$

At the infinitesimal scale, the change between people with IQ x and people with IQ $x + \epsilon$ is maximized when the derivative of the probability distribution is negative and its magnitude is maximal; thus, at a value x greater than the mean where the second derivative is 0.

The first derivative of f , $f'(x) = -((x - 100)/15^2)f(x)$.

Setting the second derivative of f to be zero, we have, $f'' = ((x - 100)^2/15^2)^2 - (1/15^2)f(x)$, so $(x - 100)^2 = 15^2$, which has two roots: $x = 85$, where the first derivative reaches a maximum positive value and $x = 105$ where it reaches a maximum magnitude negative value.

Now, that does not directly give us the value of x that maximizes $f(x) - f(x + 1)$, and finding the true maximum would involve solving an inelegant transcendental equation. However, since the third derivative of f is fairly small and the second derivative is 0, it is safe to assume that picking x and $x + 1$ evenly across $x = 15$ gives values close to the maximum value. Using binary search, one can determine that the true maximum is 114.5028.

GPT4+WA Answer: $n = 100$.

GPT4+WA Computation: The difference between the number of people with IQ n and $n + 1$ is proportional to the difference between the PDF values at n and $n + 1$. This difference is equivalent to the derivative of the PDF, which is the slope of the tangent line to the PDF at a given point.

Therefore, the maximum difference between the number of people with IQ n and $n + 1$ occurs where the derivative of the PDF is at its minimum (since the PDF of a normal distribution decreases as we move away from the mean, its derivative is negative).

The PDF of a normal distribution with mean μ and standard deviation σ is given by:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

The derivative of this function is

$$f'(x) = -\frac{x-\mu}{\sigma^2} f(x)$$

Setting this derivative to zero and solving for s will give us the value of n where the difference between the number of people with IQ n and $n + 1$ is maximized.

Let's calculate this using Wolfram Language.

```
Request to Wolfram
{
  "input": "Solve[-((x - 100)/(15^2)) * (1/(15*sqrt(2*Pi))) *
             Exp[-1/2 * ((x - 100)/15)^2] == 0, x]"
}
Response from Wolfram
"{{x -> 100}}"
```

The solution to the equation is $x = 100$. This means that the difference between the number of people with IQ n and $n + 1$ is maximized at $n = 100$. This is expected, as the mean of the distribution is also 100, and the normal distribution is symmetric about its mean.

Comment: Wrong. GPT finds the zero of the first derivative rather than its maximum.

GPT4+CI Answer: 85 and 115.

GPT4+CI Calculation:

The question asks for the score n such that the number of people with IQ n exceeds the number of people with IQ $n+1$ by the maximum amount. In other words, we're looking for the score at which the rate of decrease in the population is the greatest. This occurs at the inflection points of the normal distribution, which are located one standard deviation away from the mean.

GPT4+CI then proceeds correctly to compute symbolically the second derivative and find its zeros. However, it does not exclude the value 85.

Comment: GPT fails to realize that a zero can be a minimum rather than a maximum. Half credit.

It is notable that both versions went immediately to the differential version of the problem and that neither discussed the actual problem of maximizing $f(x) - f(x+1)$.

18. Suppose a randomized algorithm outputs the correct answer (yes or no) with probability 0.9. If we take the majority answer out of 100 independent runs, with what probability will the answer be wrong? Please include all calculations.

Answer: $3.232 \cdot 10^{-24}$.

Explanation: Assume that if the results are exactly 50/50, then you flip a fair coin. Then the exact value is

$$(1/2) \cdot C(100, 50) \cdot 0.9^{50} \cdot 0.1^{50} + \sum_{i=1}^{50} C(100, 50+i) \cdot 0.9^{50-i} \cdot 0.1^{50+i}$$

However, since this is (approximately) a geometrically decreasing series with ratio of 1/9, it suffices to take the first five terms to get three digit accuracy.

Matlab: (Matlab warns that its value may have a relative error of 10^{-14}).

```
p=nchoosek(100,50)*0.9^50*0.1^50
sum=p/2
for i=1:5
    p=nchoosek(100,50+i)*0.9^(50-i)*0.1^(50+i)
    sum = sum+p
end
sum
```

GPT4+WA Answer: $5.83 \cdot 10^{-24}$.

GPT4+WA Computation: *GPT set up the computation correctly except that it assumes that, if there is a 50-50 split, the correct answer will definitely be chosen.*

```
Request to Wolfram
{
  "input": "Sum[Binomial[100, k] * 0.9^k * (1 - 0.9)^(100 - k), {k, 0, 50}]"
}
Response from Wolfram
"5.8320387857342846 ^-24"
```

Comment: 8/10 credit. Failed to recognize that the 50-50 case is a toss-up.

GPT4+CI Answer: $5.83 \cdot 10^{-24}$.

GPT4+CI Computation: *Same computation as GPT4+WA, same mistake.*

Comment: 8/10 credit.

19. In a Manhattan-like two-dimensional grid, how many distinct shortest paths are there that go from a fixed starting point to a fixed endpoint 5 blocks to the north and 5 blocks to the east? Please include all calculations.

Answer: 252.

Explanation: You have to take 10 steps: 5 of them north and 5 of them east. They can be in any order. So the number is $C(10,5) = 252$.

GPT4+WA Answer: 252. The reasoning was correct and it called WA to carry out the calculation.

Comment: Right.

GPT4+CI Answer: 252. The reasoning was correct and it called CI to carry out the calculation.

Comment: Right. For that matter ChatGPT with GPT3.5 also got the right answer.

20. When 3-dimensional spheres are packed in their densest possible configuration, what percentage of the space is empty? Please include all calculations.

Answer: $1 - \pi/3\sqrt{2} = 0.2595$. I looked it up in the Wikipedia article Kepler conjecture

GPT4+WA Answer: $1 - \pi/3\sqrt{2} = 0.2595$.

GPT4+WA Calculation:

GPT immediately (with no other output text) tried four calls to Wolfram Alpha: "densest packing of spheres", "sphere packing density", "packing density of spheres", and "sphere packing". Wolfram Alpha did not understand the first three, but answered the final one as follows:

Definition:

Define the packing density η of a packing of spheres to be the fraction of a volume filled by the spheres. In three dimensions, there are three periodic packings for identical spheres: cubic lattice, face-centered cubic lattice, and hexagonal lattice. It was hypothesized by Kepler in 1611 that close packing (cubic or hexagonal, which have equivalent packing densities) is the densest possible, and this assertion is known as the Kepler conjecture. The problem of finding the densest packing of spheres (not necessarily periodic) is therefore known as the Kepler problem, where

`_Kepler = (F C C) = (H C P) = pi/(3 sqrt(2))74.048%`

(OEIS A093825; Steinhaus 1999, p. 202; Wells 1986, p. 29; Wells 1991, p. 237).

GPT then echoed the density information, subtracted the density from 1, and returned the correct answer.

Comment: Right. Built into Wolfram Alpha.

GPT4+CI Answer: 81.5% empty.

GPT4+CI Calculation:

The densest packing of spheres in 3 dimensions is achieved in a configuration known as a face-centered cubic (FCC) lattice or a hexagonal close-packed (HCP) lattice. In both cases, the packing fraction (i.e., the percentage of space filled by the spheres) is the same.

The packing fraction of spheres is given by the ratio of the volume of a sphere to the volume of the smallest cube that can contain it.

The volume of a sphere of radius r is given by $\frac{4}{3}\pi r^3$.

In the face-centered cubic lattice, each sphere is in contact with 12 others and can be inscribed in a cube with edge length $2\sqrt{2}r$. The volume of this cube is $(2\sqrt{2}r)^3 = 16\sqrt{2}r^3$

Therefore the packing fraction ϕ is given by

$$\phi = \frac{\text{Volume of sphere}}{\text{Volume of cube}} = \frac{\frac{4}{3}\pi r^3}{16\sqrt{2}r^3}$$

Solving this will give us the packing fraction. The percentage of empty space is then given by $(1 - \phi) \cdot 100\%$. Let's calculate these values.

```
import math

# radius of sphere
r = 1

# volume of a sphere
V_sphere = 4/3 * math.pi * r**3

# volume of the cube
V_cube = 16 * math.sqrt(2) * r**3

# packing fraction
phi = V_sphere / V_cube

# percentage of empty space
empty_space_percent = (1 - phi) * 100

phi, empty_space_percent

RESULT

(0.18512012242326523, 81.48798775767348)
```

The packing fraction, ϕ , for spheres in a face-centered cubic or hexagonal close-packed lattice is approximately 0.185.

This means that about 18.5% of the space is filled with spheres, and therefore, approximately 81.5% of the space remains empty in the densest possible packing of 3-dimensional spheres.

Comment: Wrong. GPT tries to compute the value from the geometry of the face centered cubic packing, but does not understand the geometry involved.