

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFltXafIhkcZ\_MONQLkC2#scrollTo=TmeDvLm95b9H

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Google Account  
Vaishnavi Mishra  
vaishnavimishra7021@gmail.com

Share Gemini

```
[1] import time
import tracemalloc # For memory usage
import os
import psutil
import py_compile
import math

# Function to calculate Binomial Expansion of (a + b)^n
def binomial_expansion(a, b, n):
    result = ""
    for k in range(n + 1):
        coeff = math.comb(n, k) # Combination nCk
        term = f"{coeff}"
        if n - k > 0:
            term += f"*{a}^{n - k}" # Add power of a
        if k > 0:
            term += f"*{b}^{k}" # Add power of b
        result += term + " + "
    return result.rstrip(" + ") # Remove last plus

# ----- Performance Measurement ----- #

# Measure execution time
start_time = time.perf_counter()
value = binomial_expansion("x", "2", 5) # Example: (x + 2)^5
end_time = time.perf_counter()
execution_time = end_time - start_time

# Measure memory usage
tracemalloc.start()
```

What can I help you build?

Light rain  
At night

Search

9:13 AM Python 3

ENG IN 09:17 09-08-2025

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=TmeDvLm95b9H

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

[1] # Measure memory usage  
tracemalloc.start()  
binomial\_expansion("x", "2", 5) # Run again for memory tracking  
current, peak = tracemalloc.get\_traced\_memory()  
tracemalloc.stop()  
  
# Save current cell's code to a temp file to measure file size and compile time  
temp\_filename = "temp\_code.py"  
with open(temp\_filename, "w") as f:  
 f.write("""import math  
def binomial\_expansion(a, b, n):  
 result = ""  
 for k in range(n + 1):  
 coeff = math.comb(n, k)  
 term = f"{coeff}"  
 if n - k > 0:  
 term += f"\*{a}^{n - k}"  
 if k > 0:  
 term += f"\*{b}^{k}"  
 result += term + "  
 return result.rstrip(" + ")  
""")  
  
# Measure code size (in KB)  
file\_size = os.path.getsize(temp\_filename)  
  
# Measure compilation time  
compile\_start = time.perf\_counter()  
py\_compile.compile(temp\_filename, cfile="temp\_comp  
compile\_end = time.perf\_counter()

What can I help you build?

Variables Terminal

Light rain At night

Search

9:13 AM Python 3

ENG IN 09:18 09-08-2025

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=TmeDvLm95b9H

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

RAM Disk

[1] # Measure code size (in KB)  
file\_size = os.path.getsize(temp\_filename)  
  
# Measure compilation time  
compile\_start = time.perf\_counter()  
py\_compile.compile(temp\_filename, cfile="temp\_compiled.pyc")  
compile\_end = time.perf\_counter()  
compilation\_time = compile\_end - compile\_start  
  
# Measure CPU usage percentage  
process = psutil.Process(os.getpid())  
\_ = process.cpu\_percent(interval=0.1) # First call for initialization  
cpu\_usage\_percent = process.cpu\_percent(interval=1) # Actual measurement  
  
# ----- Output Results ----- #  
print(f"Binomial Expansion for (x + 2)^5: {value}")  
print(f"Execution Time: {execution\_time:.6f} seconds")  
print(f"Current Memory Usage: {current / 1024:.2f} KB")  
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")  
print(f"Code Size: {file\_size / 1024:.2f} KB")  
print(f"Compilation Time: {compilation\_time:.6f} seconds")  
print(f"CPU Usage: {cpu\_usage\_percent}%")

Binomial Expansion for (x + 2)^5: 1\*x^5 + 5\*x^4\*2^1 + 10\*x^3\*2^2 + 10\*x^2\*2^3 + 5\*x^1\*2^4 + 1\*2^5  
Execution Time: 0.000121 seconds  
Current Memory Usage: 1.04 KB  
Peak Memory Usage: 11.77 KB  
Code Size: 0.31 KB  
Compilation Time: 0.002272 seconds  
CPU Usage: 0.0%

What can I help you build?

Variables

Terminal

Light rain  
At night

Search

ENG  
IN

9:13 AM  
09-08-2025

The screenshot displays a Google Colab notebook titled "Untitled12.ipynb". The interface includes a top navigation bar with tabs for "Welcome To Colab - Colab", "Untitled12.ipynb - Colab", "Introducing ChatGPT | OpenAI", and "Google Colab code update". The notebook's menu bar shows "File", "Edit", "View", "Insert", "Runtime", "Tools", and "Help". A "Share" button and a "Gemini" icon are also present. The left sidebar contains icons for file explorer, search, and other tools. The main area shows a Python script with the following code:

```
import time
import tracemalloc # For memory usage
import os
import psutil
import py_compile

# Function to check if a number is an Armstrong number
def is_armstrong(num):
    digits = str(num) # Convert number to string
    power = len(digits) # Number of digits
    total = sum(int(d) ** power for d in digits) # Sum of each digit raised to 'power'
    return total == num # True if Armstrong number

# ----- Performance Measurement ----- #
test_number = 153 # Example test number

# Measure execution time
start_time = time.perf_counter()
result = is_armstrong(test_number)
end_time = time.perf_counter()
execution_time = end_time - start_time

# Measure memory usage
tracemalloc.start()
is_armstrong(test_number) # Run again for memory tracking
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()

# Save current function code to a temp file for si
temp_filename = "temp_code.py"
```

At the bottom of the script, there is a prompt: "What can I help you build?". The bottom status bar shows "Variables", "Terminal", "9:13 AM", and "Python 3". The Windows taskbar at the very bottom includes a weather widget showing "Light rain At night", a search bar, and various application icons.

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=jXfXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

RAM Disk

What can I help you build?

```
import time
import tracemalloc # For memory usage
import os
import psutil
import py_compile

# Function to check if a number is an Armstrong number
def is_armstrong(num):
    digits = str(num) # Convert number to string
    power = len(digits) # Number of digits
    total = sum(int(d) ** power for d in digits) # Sum of each digit raised to 'power'
    return total == num # True if Armstrong number

# ----- Performance Measurement ----- #
test_number = 153 # Example test number

# Measure execution time
start_time = time.perf_counter()
result = is_armstrong(test_number)
end_time = time.perf_counter()
execution_time = end_time - start_time

# Measure memory usage
tracemalloc.start()
is_armstrong(test_number) # Run again for memory tracking
current, peak = tracemalloc.get_traced_memory()
tracemalloc.stop()

# Save current function code to a temp file for si
temp_filename = "temp_code.py"
```

Variables Terminal

Light rain At night

9:13 AM Python 3

ENG IN 09:18 09-08-2025

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTu48ShFltXafhlkcZ\_MONQLkC2#scrollTo=jXfXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

RAM  
Disk

↑ ↓ ⚡ 🔗 🗨 ⚙ 📄 🗑 ⋮

▶

```
# Save current function code to a temp file for size and compile time measurement
temp_filename = "temp_code.py"
with open(temp_filename, "w") as f:
    f.write("""def is_armstrong(num):
    digits = str(num)
    power = len(digits)
    total = sum(int(d) ** power for d in digits)
    return total == num
""")

# Measure code size (in KB)
file_size = os.path.getsize(temp_filename)

# Measure compilation time
compile_start = time.perf_counter()
py_compile.compile(temp_filename, cfile="temp_compiled.pyc")
compile_end = time.perf_counter()
compilation_time = compile_end - compile_start

# Measure CPU usage percentage
process = psutil.Process(os.getpid())
_ = process.cpu_percent(interval=0.1) # First call just to initialize
cpu_usage_percent = process.cpu_percent(interval=1) # Actual measurement

# ----- Output Results ----- #
print(f"{test_number} is an Armstrong number: {result}")
print(f"Execution Time: {execution_time:.6f} seconds")
print(f"Current Memory Usage: {current / 1024:.2f} KB")
print(f"Peak Memory Usage: {peak / 1024:.2f} KB")
print(f"Code Size: {file_size / 1024:.2f} KB")
print(f"Compilation Time: {compilation_time:.6f} seconds")
```

What can I help you build?

Variables

Terminal

Light rain  
At night

Search

🐱

📁

🌐

📁

🔄

🌐

📁

📁

9:13 AM

Python 3

ENG  
IN

09:18  
09-08-2025

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=jXfIXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

file\_size = os.path.getsize(temp\_filename)

# Measure compilation time

compile\_start = time.perf\_counter()

py\_compile.compile(temp\_filename, cfile="temp\_compiled.pyc")

compile\_end = time.perf\_counter()

compilation\_time = compile\_end - compile\_start

# Measure CPU usage percentage

process = psutil.Process(os.getpid())

\_ = process.cpu\_percent(interval=0.1) # First call just to initialize

cpu\_usage\_percent = process.cpu\_percent(interval=1) # Actual measurement

# ----- Output Results ----- #

print(f"{test\_number} is an Armstrong number: {result}")

print(f"Execution Time: {execution\_time:.6f} seconds")

print(f"Current Memory Usage: {current / 1024:.2f} KB")

print(f"Peak Memory Usage: {peak / 1024:.2f} KB")

print(f"Code Size: {file\_size / 1024:.2f} KB")

print(f"Compilation Time: {compilation\_time:.6f} seconds")

print(f"CPU Usage: {cpu\_usage\_percent}%")

153 is an Armstrong number: True  
Execution Time: 0.001199 seconds  
Current Memory Usage: 0.97 KB  
Peak Memory Usage: 11.69 KB  
Code Size: 0.13 KB  
Compilation Time: 0.000831 seconds  
CPU Usage: 0.0%

What can I help you build?

Variables Terminal

Light rain  
At night

Search

09:19  
09-08-2025

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=jXfIXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

✓ [3] import time  
import tracemalloc # For memory usage tracking  
import os  
import psutil  
import py\_compile # For compilation time  
  
# ----- Leap Year Function ----- #  
def is\_leap\_year(year):  
 """  
 Determines if the given year is a leap year.  
 A year is a leap year if:  
 - It is divisible by 4, AND  
 - (It is not divisible by 100 OR it is divisible by 400)  
 """  
 if year % 4 == 0: # Check divisibility by 4  
 if (year % 100 != 0) or (year % 400 == 0): # Check 100 and 400 rules  
 return True  
 return False  
  
# ----- Performance Measurement ----- #  
year = 2024 # Example year (replace with your test year)  
  
# Measure execution time  
start\_time = time.perf\_counter()  
is\_leap = is\_leap\_year(year)  
end\_time = time.perf\_counter()  
execution\_time = end\_time - start\_time  
  
# Measure memory usage  
tracemalloc.start()  
is\_leap\_year(year) # Run again for memory tracking

What can I help you build?

Variables

Terminal

9:13 AM

Python 3

Light rain  
At night

Search

ENG  
IN

09:19  
09-08-2025



colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=jXfIXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Q Commands + Code + Text ▶ Run all

RAM Disk

[3] # Measure memory usage  
tracemalloc.start()  
is\_leap\_year(year) # Run again for memory tracking  
current, peak = tracemalloc.get\_traced\_memory()  
tracemalloc.stop()  
  
# Save current function code to a temp file for size and compile time measurement  
temp\_filename = "temp\_code.py"  
with open(temp\_filename, "w") as f:  
 f.write("""def is\_leap\_year(year):  
 if year % 4 == 0:  
 if (year % 100 != 0) or (year % 400 == 0):  
 return True  
 return False  
 """)  
  
# Measure code size (in KB)  
file\_size = os.path.getsize(temp\_filename)  
  
# Measure compilation time  
compile\_start = time.perf\_counter()  
py\_compile.compile(temp\_filename, cfile="temp\_compiled.pyc")  
compile\_end = time.perf\_counter()  
compilation\_time = compile\_end - compile\_start  
  
# Measure CPU usage percentage  
process = psutil.Process(os.getpid())  
\_ = process.cpu\_percent(interval=0.1) # Warm-up call  
cpu\_usage\_percent = process.cpu\_percent(interval=1)

What can I help you build?

Variables Terminal

9:13 AM Python 3

Light rain  
At night

Search

ENG  
IN

09:19  
09-08-2025

Welcome To Colab - Colab

Untitled12.ipynb - Colab

Introducing ChatGPT | OpenAI

Google Colab code update

colab.research.google.com/drive/1-AOTHu48ShFlitXafhlkcZ\_MONQLkC2#scrollTo=jXfIXP3O4ZKw

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

RAM

Disk

[3] # Measure CPU usage percentage

process = psutil.Process(os.getpid())

\_ = process.cpu\_percent(interval=0.1) # Warm-up call

cpu\_usage\_percent = process.cpu\_percent(interval=1)

# ----- Output Results ----- #

if is\_leap:

print(f"{year} is a leap year.")

else:

print(f"{year} is not a leap year.")

print(f"Execution Time: {execution\_time:.6f} seconds")

print(f"Current Memory Usage: {current / 1024:.2f} KB")

print(f"Peak Memory Usage: {peak / 1024:.2f} KB")

print(f"Code Size: {file\_size / 1024:.2f} KB")

print(f"Compilation Time: {compilation\_time:.6f} seconds")

print(f"CPU Usage: {cpu\_usage\_percent}%")

2024 is a leap year.

Execution Time: 0.000076 seconds

Current Memory Usage: 1.11 KB

Peak Memory Usage: 11.84 KB

Code Size: 0.13 KB

Compilation Time: 0.000735 seconds

CPU Usage: 0.0%

What can I help you build?

Variables

Terminal

Light rain

At night

Search

9:13 AM

Python 3

ENG

IN

09:19

09-08-2025