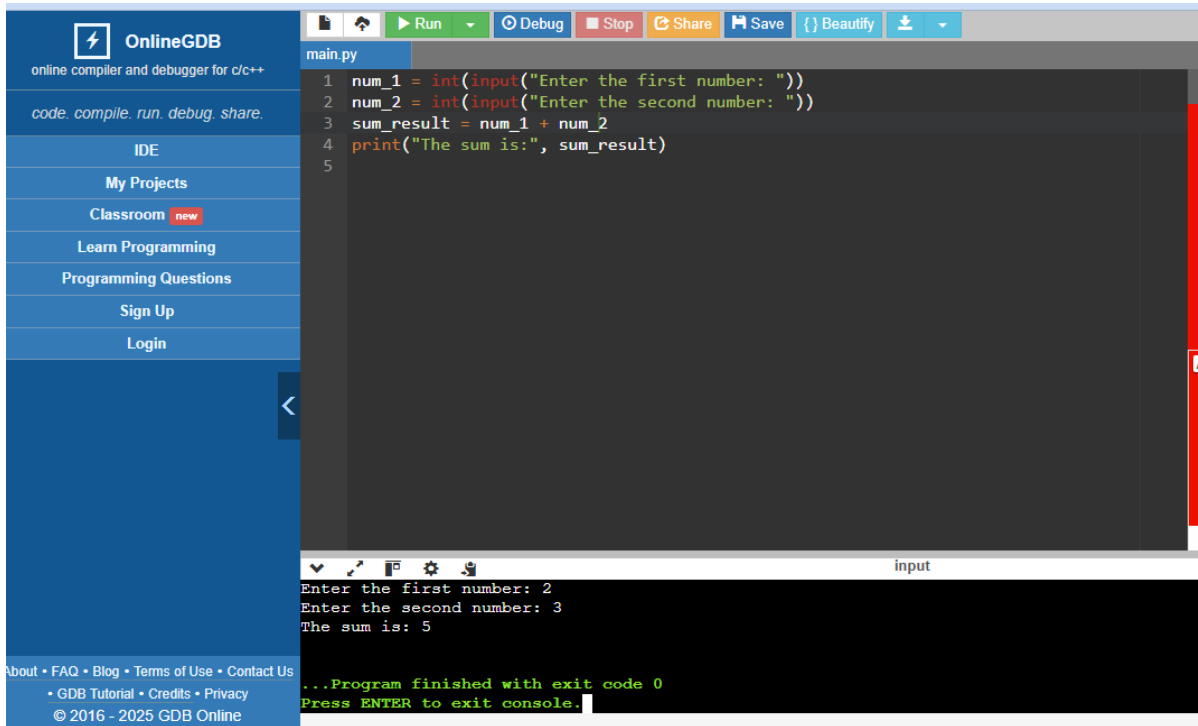


TASK:1

PYTHON INTERNSHIP

1. The sum of Two Numbers

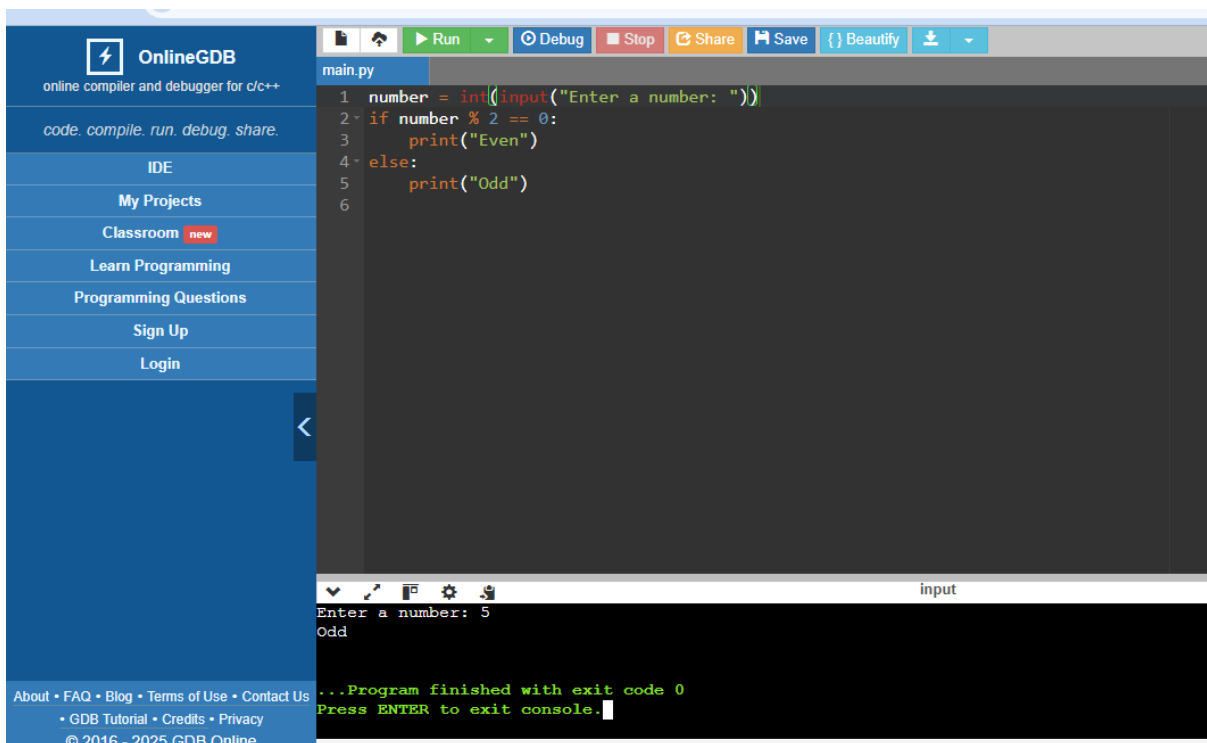


The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with navigation links: OnlineGDB, code, compile, run, debug, share, IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a Python file named 'main.py' with the following code:

```
1 num_1 = int(input("Enter the first number: "))
2 num_2 = int(input("Enter the second number: "))
3 sum_result = num_1 + num_2
4 print("The sum is:", sum_result)
5
```

Below the editor is a console window showing the program's execution. It prompts for two numbers, receives '2' and '3', and outputs 'The sum is: 5'. The console also shows the program finished with exit code 0 and a prompt to press ENTER to exit.

2. Odd or Even

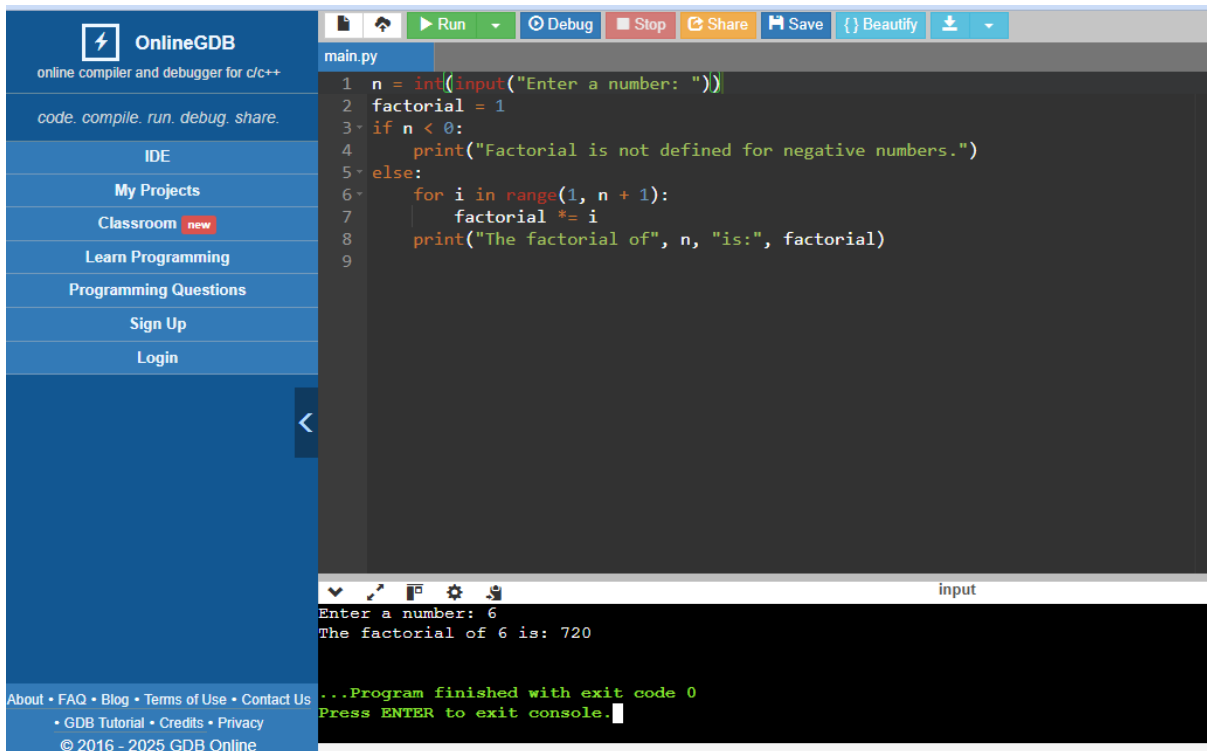


The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with navigation links: OnlineGDB, code, compile, run, debug, share, IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a Python file named 'main.py' with the following code:

```
1 number = int(input("Enter a number: "))
2 if number % 2 == 0:
3     print("Even")
4 else:
5     print("Odd")
6
```

Below the editor is a console window showing the program's execution. It prompts for a number, receives '5', and outputs 'Odd'. The console also shows the program finished with exit code 0 and a prompt to press ENTER to exit.

3. Factorial Calculation



The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with the OnlineGDB logo and navigation links: code, compile, run, debug, share. The main editor displays a Python file named main.py with the following code:

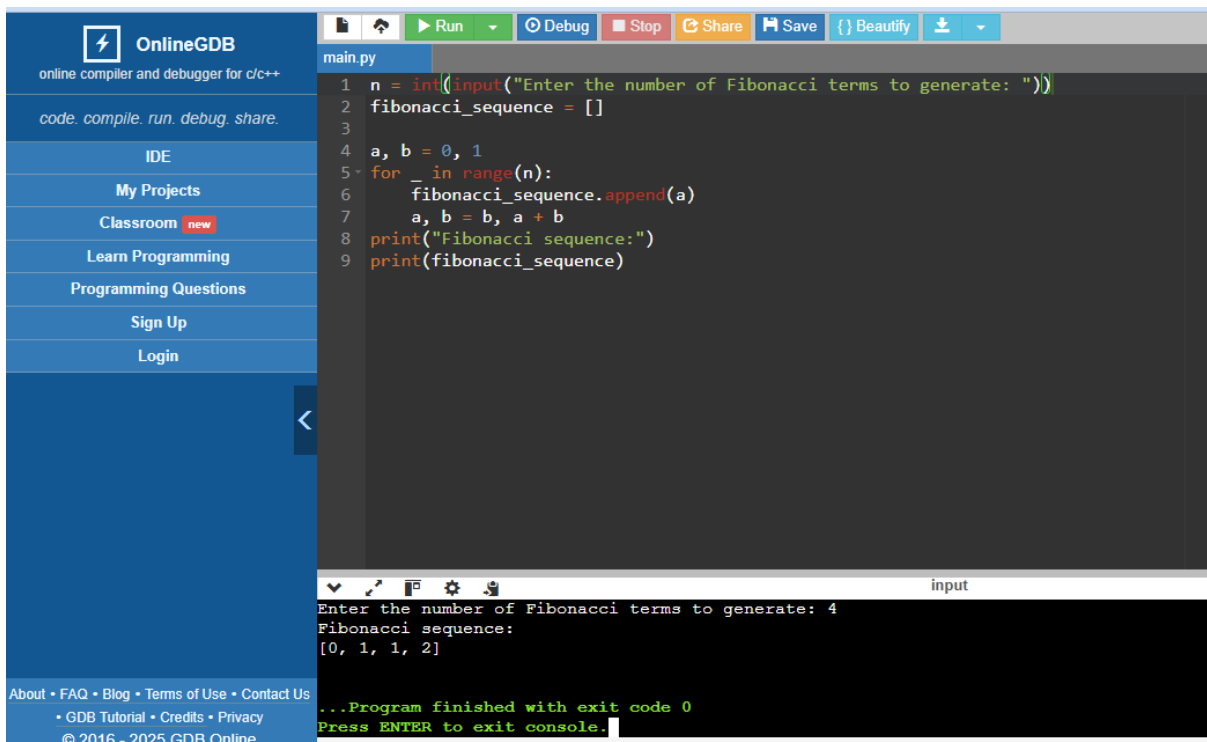
```
1 n = int(input("Enter a number: "))
2 factorial = 1
3 if n < 0:
4     print("Factorial is not defined for negative numbers.")
5 else:
6     for i in range(1, n + 1):
7         factorial *= i
8     print("The factorial of", n, "is:", factorial)
9
```

Below the editor is a console window with the input and output:

```
Enter a number: 6
The factorial of 6 is: 720

...Program finished with exit code 0
Press ENTER to exit console.
```

4. Fibonacci Sequence



The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with the OnlineGDB logo and navigation links: code, compile, run, debug, share. The main editor displays a Python file named main.py with the following code:

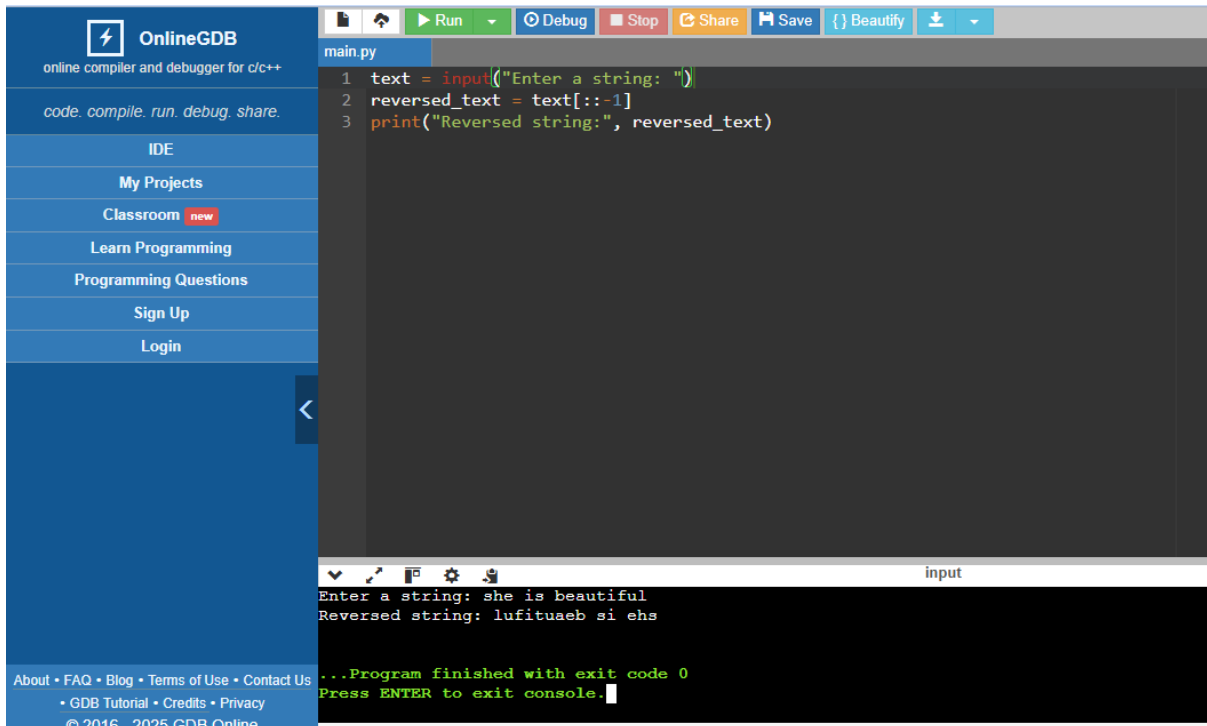
```
1 n = int(input("Enter the number of Fibonacci terms to generate: "))
2 fibonacci_sequence = []
3
4 a, b = 0, 1
5 for _ in range(n):
6     fibonacci_sequence.append(a)
7     a, b = b, a + b
8 print("Fibonacci sequence:")
9 print(fibonacci_sequence)
```

Below the editor is a console window with the input and output:

```
Enter the number of Fibonacci terms to generate: 4
Fibonacci sequence:
[0, 1, 1, 2]

...Program finished with exit code 0
Press ENTER to exit console.
```

5. Reverse a String

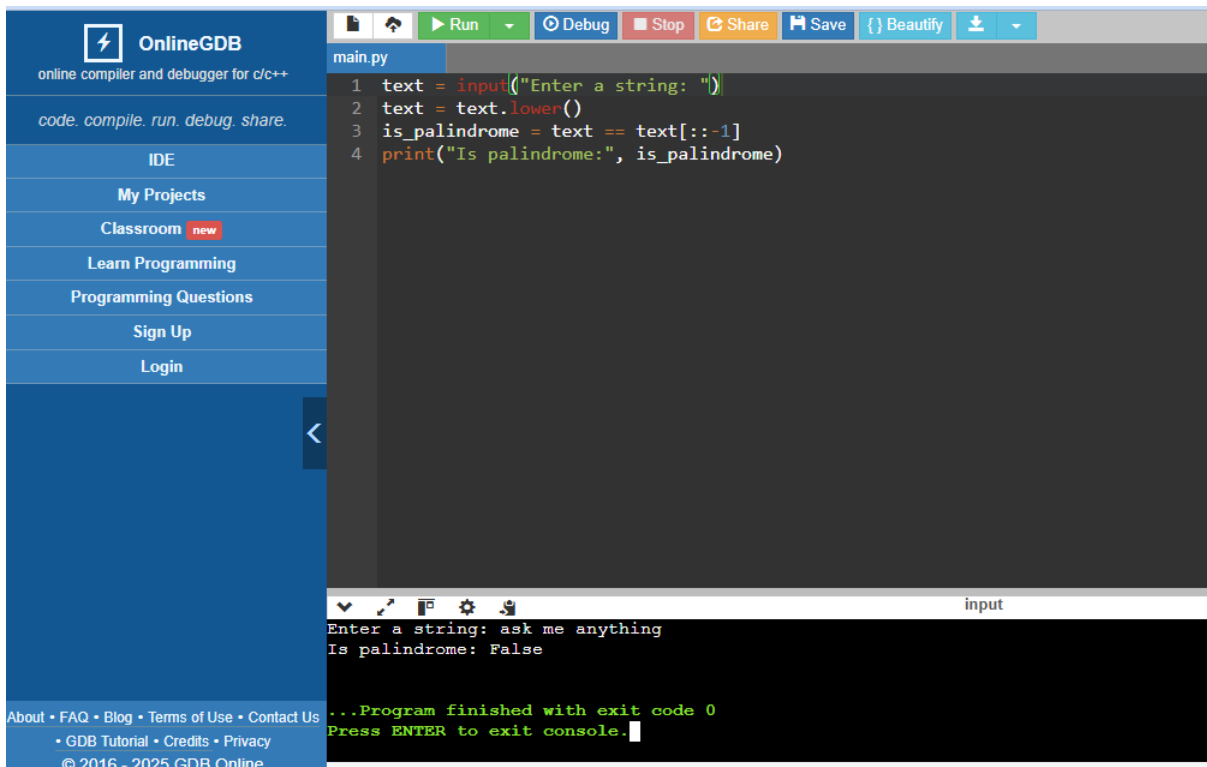


The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with navigation links: OnlineGDB, code, compile, run, debug, share, IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a Python file named 'main.py' with the following code:

```
1 text = input("Enter a string: ")
2 reversed_text = text[::-1]
3 print("Reversed string:", reversed_text)
```

Below the editor is a console window. It shows the program's execution: the prompt 'Enter a string: ' is followed by the user input 'she is beautiful'. The program then outputs 'Reversed string: lufituaeb si ehs'. The console also shows the program finished with exit code 0 and a prompt to press ENTER to exit the console.

6. Palindrome Check

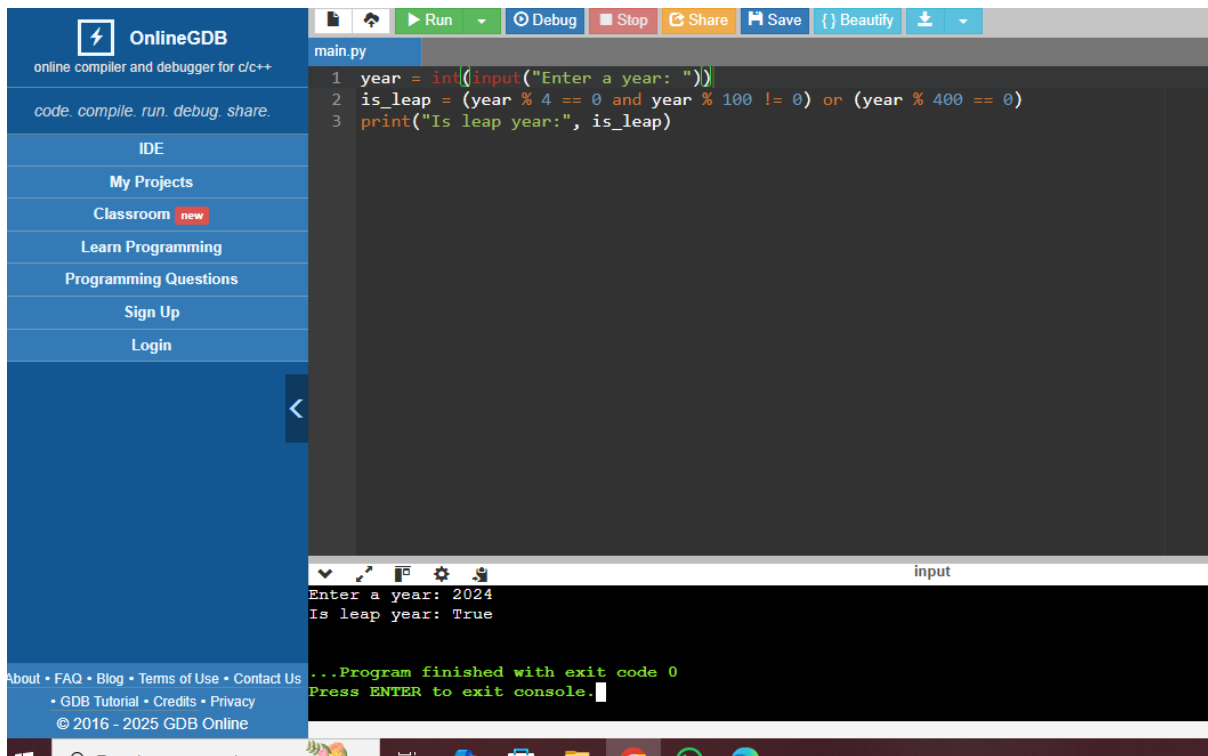


The screenshot shows the OnlineGDB IDE interface. On the left is a sidebar with navigation links: OnlineGDB, code, compile, run, debug, share, IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a Python file named 'main.py' with the following code:

```
1 text = input("Enter a string: ")
2 text = text.lower()
3 is_palindrome = text == text[::-1]
4 print("Is palindrome:", is_palindrome)
```

Below the editor is a console window. It shows the program's execution: the prompt 'Enter a string: ' is followed by the user input 'ask me anything'. The program then outputs 'Is palindrome: False'. The console also shows the program finished with exit code 0 and a prompt to press ENTER to exit the console.

7. Leap Year Check



OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom **new**
Learn Programming
Programming Questions
Sign Up
Login

main.py

```
1 year = int(input("Enter a year: "))
2 is_leap = (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)
3 print("Is leap year:", is_leap)
```

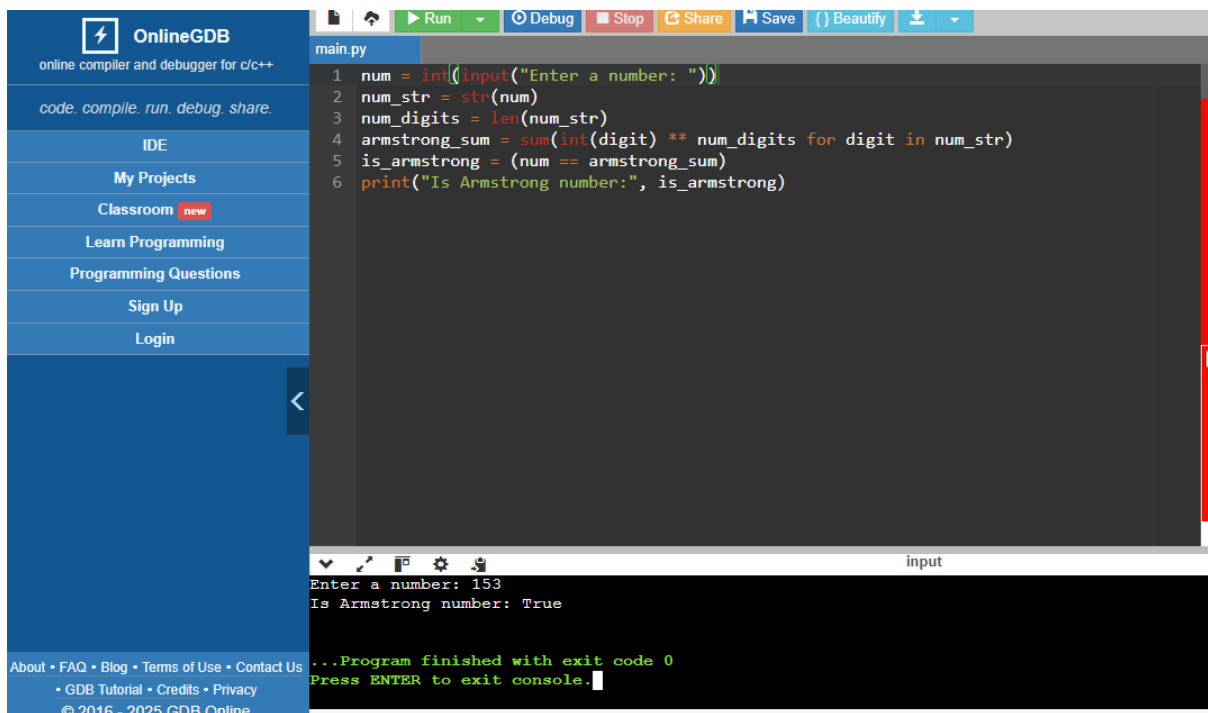
input

```
Enter a year: 2024
Is leap year: True

...Program finished with exit code 0
Press ENTER to exit console.
```

About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

8. Armstrong Number



OnlineGDB
online compiler and debugger for c/c++
code. compile. run. debug. share.

IDE
My Projects
Classroom **new**
Learn Programming
Programming Questions
Sign Up
Login

main.py

```
1 num = int(input("Enter a number: "))
2 num_str = str(num)
3 num_digits = len(num_str)
4 armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)
5 is_armstrong = (num == armstrong_sum)
6 print("Is Armstrong number:", is_armstrong)
```

input

```
Enter a number: 153
Is Armstrong number: True

...Program finished with exit code 0
Press ENTER to exit console.
```

About • FAQ • Blog • Terms of Use • Contact Us
• GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

1. Custom Encryption-Decryption System

>APPROACH: -

The aim of this project is to create an original encryption-decryption system without calling any pre-compiled libraries, like "cryptography" or "hashlib." Three positions in a substitution cipher were shifted by characters. Reversal of the string and case switching was used for an optional multi-layered system to enhance security.

>CHALLENGES HANDELS: -

Both uppercase and lowercase letters also special characters, then spaces, and here comes numbers are handled by the system. Additional questions is provided by an optional time taken encryption layer so, making brute-force attacks harder.

>SKILLS DEMONSTRATED: -Logical reasoning, string manipulation, and algorithm design skills are to be demonstrated by this project. Students gain the ability to make educated decisions regarding set structure, key design, and encryption.

>Code implementation :-

```
def encrypt(text, shift):  
    ... # Shifts characters forward
```

```
def decrypt(ciphertext, shift):  
    ... # Reverses the shift
```

>CODE:-

```
# Caesar Cipher - Custom Encryption-Decryption System
```

```
def encrypt(text, shift):  
    encrypted = ""  
    for char in text:  
        if char.isalpha():  
            base = ord('A') if char.isupper() else ord('a')  
            encrypted += chr((ord(char) - base + shift) % 26 + base)
```


```
elif char.isdigit():
    encrypted += chr((ord(char) - ord('0') + shift) % 10 + ord('0'))
else:
    # Keep special characters unchanged
    encrypted += char
return encrypted

def decrypt(ciphertext, shift):
    return encrypt(ciphertext, -shift)

# --- Example usage ---
message = input("Enter the message: ")
key = int(input("Enter the encryption key (shift): "))

# Encryption
cipher = encrypt(message, key)
print("Encrypted message:", cipher)

# Decryption
original = decrypt(cipher, key)
print("Decrypted message:", original)
```

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

About • FAQ • Blog • Terms of Use • Contact Us

GDB Tutorial • Credits • Privacy

© 2016 - 2025 GDB Online

main.py

1 # Caesar Cipher - Custom Encryption-Decryption System

2

3 def encrypt(text, shift):

4 encrypted = ""

5 for char in text:

6 if char.isalpha():

7 base = ord('A') if char.isupper() else ord('a')

8 encrypted += chr((ord(char) - base + shift) % 26 + base)

9 elif char.isdigit():

10 encrypted += chr((ord(char) - ord('0') + shift) % 10 + ord('0'))

11 else:

12 # Keep special characters unchanged

13 encrypted += char

14 return encrypted

15

16 def decrypt(ciphertext, shift):

17 return encrypt(ciphertext, -shift)

18

19 # --- Example usage ---

20 message = input("Enter the message: ")

21 key = int(input("Enter the encryption key (shift): "))

22

23 # Encryption

24 cipher = encrypt(message, key)

25 print("Encrypted message:", cipher)

26

27 # Decryption

28 original = decrypt(cipher, key)

29 print("Decrypted message:", original)

30

31


Enter the message: hello, world

Enter the encryption key (shift): 3

Encrypted message: kloor, zruog

Decrypted message: hello, world

...Program finished with exit code 0

**OnlineGDB**

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

About • FAQ • Blog • Terms of Use • Contact Us

GDB Tutorial • Credits • Privacy

© 2016 - 2025 GDB Online

main.py

9 elif char.isdigit():

10 encrypted += chr((ord(char) - ord('0') + shift) % 10 + ord('0'))

11 else:

12 # Keep special characters unchanged

13 encrypted += char

14 return encrypted

15

16 def decrypt(ciphertext, shift):

17 return encrypt(ciphertext, -shift)

18

19 # --- Example usage ---

20 message = input("Enter the message: ")

21 key = int(input("Enter the encryption key (shift): "))

22

23 # Encryption

24 cipher = encrypt(message, key)

25 print("Encrypted message:", cipher)

26

27 # Decryption

28 original = decrypt(cipher, key)

29 print("Decrypted message:", original)

30

31

Enter the message: hello, world

Enter the encryption key (shift): 3

Encrypted message: kloor, zruog

Decrypted message: hello, world

...Program finished with exit code 0