

Source Code:

```
num1 = float(input("Enter num1: "))
num2 = float(input("Enter num2: "))
charecter = input("Enter +, -, *, /, %: ")
press = input("Enter =: ")

# Performing the operation when the '=' is pressed
if press == '=':
    if charecter == '+':
        print(f"{num1} + {num2} = {num1 + num2}")
    elif charecter == '-':
        print(f"{num1} - {num2} = {num1 - num2}")
    elif charecter == '*':
        print(f"{num1} * {num2} = {num1 * num2}")
    elif charecter == '/':
        if num2 != 0:
            print(f"{num1} / {num2} = {num1 / num2}")
        else:
            print("Division by zero is not allowed.")
    elif charecter == '%':
        print(f"{num1} % {num2} = {int(num1) % int(num2)}")
    else:
        print("Invalid operator")
```

Output:

```
Enter num1: 6
Enter num2: 3
Enter +, -, *, /, %: *
Enter =: =
```

6.0 * 3.0 = 18.0

Source Code:

```
n = int(input("Enter number: "))
# Initializing a vector with 2*n elements
vec = [0] * (2 * n)

print("Enter numbers:")
for i in range(2 * n):
    vec[i] = float(input())

# Taking the operator input
ch = input("Enter operator (+, -, *, /): ")

# Performing operations based on the input operator
if ch == '+':
    for i in range(0, 2 * n, 2):
        print(f"{vec[i]} + {vec[i+1]} = {vec[i] + vec[i+1]}")
elif ch == '-':
    for i in range(0, 2 * n, 2):
        print(f"{vec[i]} - {vec[i+1]} = {vec[i] - vec[i+1]}")
elif ch == '*':
    for i in range(0, 2 * n, 2):
        print(f"{vec[i]} * {vec[i+1]} = {vec[i] * vec[i+1]}")
elif ch == '/':
    for i in range(0, 2 * n, 2):
        if vec[i+1] != 0:
            print(f"{vec[i]} / {vec[i+1]} = {vec[i] / vec[i+1]}")
        else:
            print(f"Division by zero is not allowed for {vec[i]} / {vec[i+1]}")
else:
    print("Invalid operator")
```

Output:

Enter number: 3

Enter numbers:

4

5

6

7

8

9

Enter operator (+, -, *, /): +

4.0 + 5.0 = 9.0

6.0 + 7.0 = 13.0

8.0 + 9.0 = 17.0

Source Code:

```
# Function to check if input is a palindrome
def is_palindrome(value):
    value = str(value) # Convert input to string
    return value == value[::-1] # Check if string is the same when reversed

# Main function to take input and show the result
def main():
    user_input = input("Enter a number or string: ") # Take input from user
    if is_palindrome(user_input): # Check if input is palindrome
        print(f"{user_input} is a palindrome.")
    else:
        print(f"{user_input} is not a palindrome.")

# Run the program
if __name__ == "__main__":
    main()
```

Output:

Enter a number or string: naim

naim is not a palindrome.

Source Code:

```
num = int(input("Enter a number: "))

# Calculating factorial using a for loop
fact = 1
for i in range(1, num + 1):
    fact *= i
print(f"Factorial using for loop: {fact}")

# Calculating factorial using a while loop
fact = 1
i = 1
while i <= num:
    fact *= i
    i += 1
print(f"Factorial using while loop: {fact}")
```

Output:

Enter a number: 6

Factorial using for loop: 720

Factorial using while loop: 720

Source Code:

```
def sum_array(arr1):
    i = 0
    total_sum = 0
    while i < len(arr1):
        total_sum += arr1[i]
        i += 1
    print(f"Sum: {total_sum}")
def avg_array(arr1):
    i = 0
    total_sum = 0
    while i < len(arr1):
        total_sum += arr1[i]
        i += 1
    print(f"Average: {total_sum / len(arr1)}")

# Main program
def main():
    n = int(input("Enter array size: "))
    arr = []

    print("Enter array elements: ")
    for i in range(n):
        arr.append(int(input())) # Adding elements to the array

    print()
    sum_array(arr) # Calling sum function
    print()
    avg_array(arr) # Calling average function

if __name__ == "__main__":
    main()
```

Output:

Enter array size: 5
Enter array elements:

2
3
5
6
7

Sum: 23

Average: 4.6

Source Code:

```
import os
class ExistClass:
    @staticmethod
    def get_department_name():
        print("Information and Communication Engineering")
def main():
    try:
        if 'ExistClass' in globals():
            print("Class Found inside module.\n-----")
        else:
            raise ImportError("ClassNotFoundException")
    except ImportError:
        print("ClassNotFoundException")

    print("EOF exception for output")
    try:
        with open('input.txt', 'rb') as f:
            while True:
                try:
                    ch = f.read(1) # Reading one byte at a time
                    if not ch: # If end of file is reached, break the loop
                        break
                    print(ch.decode(), end=") # Decoding and printing the byte as a character
                except EOFError:
                    print("\nEnd of file reached")
                    break
            except FileNotFoundError:
                print("File not found: input.txt")
            except Exception as e:
                print(f"An error occurred: {e}")

if __name__ == "__main__":
    main()
```

Output:

EOF exception for output
Department Name: Information and Communication Engineering.
Process finished with exit code 0

Source Code:

```
import sys
sys.stdin = open('input.txt', 'r')
sys.stdout = open('output.txt', 'w')

# Read the number of test cases
testcase = int(input())
v = []

# Reading input pairs and storing in a list
for _ in range(testcase):
    Num1, Num2 = map(int, input().split())
    v.append(Num1)
    v.append(Num2)

# Performing operations and writing results to output.txt
print(f"Sum of Two numbers: {v[0] + v[1]}")
print(f"Subtraction of Two numbers: {v[2] - v[3]}")
print(f"Multiplication of Two numbers: {v[4] * v[5]}")
print(f"Division of Two numbers: {v[6] // v[7]}") #
```

Output:

```
3
5 3
10 7
6 2
```

```
Sum of Two number : 8
Subtraction of Two number: 3
Multiplication of Two number: 42
Division of Two number : 3
```