In [ ]:

In [ ]:

input function in python

In [ ]:
```
The input() function allows user input.
```

In [2]:
```
name = input("Please enter your name: ")
print("Hello, " + name + "!")
print(type(name))
```

```
Hello, swati!
<class 'str'>
```

In [8]:
```
num1 = input("Enter first number: ")
num2 = input("Enter second number: ")

# Converting input from string to integer
num1 = int(num1)
num2 = int(num2)

sum_result = num1 + num2
print("Sum of numbers:", sum_result)
```

```
Sum of numbers: 8
```

pow function

In [11]:
```
result = pow(2, 3)
print(result)
```

```
8
```

In [13]:
```
result = pow(12, 3.5)
print(result)
```

```
5985.96759095804
```

In [15]:
```
print(12**3.5)
```

```
5985.96759095804
```

abs Function

In [18]:
```
num1 = -10
result1 = abs(num1)
print(result1)
```

```
10
```

In [20]:
```
num2 = 20
result2 = abs(num2)
```

```
print(result2)
```

```
20
```

In [22]:
```
num3 = -15.7
result3 = abs(num3)
print(result3)
```

```
15.7
```

In [24]:
```
z = -3 + 4j
absolute_value_z = abs(z)
print("Absolute value of", z, "is", absolute_value_z)
```

```
Absolute value of (-3+4j) is 5.0
```

len function

In [27]:
```
text = "Hello, World!"
length_of_text = len(text)
print("Length of the string:", length_of_text)
```

```
Length of the string: 13
```

In [29]:
```
my_list = [1, 2, 3, 4, 5]
length_of_list = len(my_list)
print("Length of the list:", length_of_list)
```

```
Length of the list: 5
```

Bin Function

In [ ]:
```
The bin() function in Python is used to
convert an integer number to its
binary representation as a string prefixed with '0b'.
It takes an integer as an argument
and returns its binary representation.
```

In [32]:
```
# Convert an integer to its binary representation
binary_representation = bin(10)
print("Binary Representation of 10:", binary_representation)
```

```
Binary Representation of 10: 0b1010
```

In [34]:
```
y = bin(36)
print(y)
```

```
0b100100
```

float Function

In [37]:
```
num_int = 10
print(type(num_int))
num = float(num_int)
print(num)
print(type(num))
```

```
<class 'int'>
10.0
<class 'float'>
```

In [39]:
```
str_num = "3.14"
print(type(str_num))
num1 = float(str_num)
print(num1)
print(type(num1))
```

```
<class 'str'>
3.14
<class 'float'>
```

int Function

In [42]:
```
a = 5.68952
print(type(a))
b = int(a)
print(b)
print(type(b))
```

```
<class 'float'>
5
<class 'int'>
```

In [44]:
```
a1 = "22"
print(type(a1))
b1 = int(a1)
print(b1)
print(type(b1))
```

```
<class 'str'>
22
<class 'int'>
```

str Function

In [47]:
```
# Convert an integer to a string
a3 = 68
print(type(a3))
b3 = str(a3)
print(b3)
print(type(b3))
```

```
<class 'int'>
68
<class 'str'>
```

In [49]:
```
# Convert a float to a string
a4 = 8.5624
print(type(a4))
b4 = str(a4)
print(b4)
print(type(b4))
```

```
<class 'float'>
8.5624
<class 'str'>
```

In [51]:
```
# Convert a boolean to a string
bool_value = True
print(type(bool_value))
bool_str = str(bool_value)
print(bool_str)
print(type(bool_str))
```

```
<class 'bool'>
True
<class 'str'>
```

In [53]:
```
# Convert a list to a string
my_list = [1, 2, 3]
print(type(my_list))
list_str = str(my_list)
print(list_str)
print(type(list_str))
```

```
<class 'list'>
[1, 2, 3]
<class 'str'>
```

complex Function

In [56]:
```
# Create a complex number with real and imaginary parts
complex_number = complex(2, 3)
print( complex_number)
```

```
(2+3j)
```

In [58]:
```
y = complex(2.5, 3.96)
print(y)
```

```
(2.5+3.96j)
```

eval function

In [ ]:
```
eval(): Evaluates a Python expression stored in a string
```

In [61]:
```
x = 10
y = 5
expression = 'x + y * 2'
result = eval(expression)
print(result)  # Output: 20
```

```
20
```

In [63]:
```
expression = '5 / 2 + 3.5'
result = eval(expression)
print(result)  # Output: 5.0
```

```
6.0
```

In [65]:
```
expression = '(10 + 2) * 3 - 5'
result = eval(expression)
print(result)  # Output: 31
```

```
31
```

In [67]:
```
x = 5
y = 3
expression = 'x * y + 2 * x - y'
result = eval(expression)
print(result)  # Output: 22
```

```
22
```

In [ ]:
```
Help function in Python
```

In [ ]:
```
The Python help function is used to display the
documentation of modules, functions, classes, keywords, etc
```

In [69]:
```
help(print)
```

```
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```

In [ ]:
```
ou are given a string that represents a mathematical expression (only containing

Take the input expression as a string.
Calculate the result of the expression using eval().
Find the absolute value of the result using abs().
Calculate the length of the input expression (the number of characters, includin
Return
The absolute value of the result of the expression.
The length of the input string (number of characters).
```

In [75]:
```
# Take the input string containing a mathematical expression
input_expression = input()

# Calculate the result of the expression using eval()
result = eval(input_expression)

# Calculate the absolute value of the result
absolute_result = abs(result)

# Find the length of the input expression
length_of_expression = len(input_expression)

# Return a tuple with the absolute value of the result and the length of the inp
```

```
print(absolute_result)
print(length_of_expression)
```

```
1
5
```

In [ ]:
```
You are managing the salaries of 5 employees. Your task is to:

Take the salary of 5 employees as input (one by one).
Calculate the average salary of the 5 employees.
Find the maximum salary among the 5 employees.
Find the minimum salary among the 5 employees.
```

In [81]:
```
# Take the salary of 5 employees as input
salary1 = int(input())
salary2 = int(input())
salary3 = int(input())
salary4 = int(input())
salary5 = int(input())

# Calculate the average salary
average_salary = sum([salary1, salary2, salary3, salary4, salary5]) / 5

# Find the maximum and minimum salary
max_salary = max(salary1, salary2, salary3, salary4, salary5)
min_salary = min(salary1, salary2, salary3, salary4, salary5)

# Return a tuple with the average, maximum, and minimum salary
print("average salary is ",average_salary)
print("maximum salary is ",max_salary)
print("minimum salary is",min_salary)
```

```
average salary is  66600.0
maximum salary is  95000
minimum salary is 45000
```

In [ ]:
```
Problem:

You are given a string representing a mathematical expression with integers and

Take the mathematical expression as input.
Evaluate the expression using eval().
Find the absolute value of the result of the expression.
Find the length of the expression (including spaces and operators).
Return a tuple containing:
The absolute value of the result.
The length of the expression.
```

In [85]:
```
# Take the mathematical expression as input
expression = input()

# Evaluate the expression using eval()
result = eval(expression)

# Calculate the absolute value of the result
abs_result = abs(result)

# Find the length of the expression
length_of_expression = len(expression)
```

```
# Return a tuple with the absolute value and length of the expression
print("absolute value is",abs_result)
print("the length of the expression is ",length_of_expression)
```

```
absolute value is 3
the length of the expression is  13
```

In [ ]: