python-codes

* Write program that reads a number N and prints square of N rows and N columns using numbers starting from 1

```python
N = int(input())
num = 1
for i in range(N):
    for j in range(N):
        print(num, end=" ")
        num += 1
    print()
```

* Write a program that reads a number N and prints two Right

```python
N = int(input())
for i in range(1, N+1):
    print("* " * i)
print()
for i in range(1, N+1):
    print("* " * i)
```

* Angled Triangles of N rows, using numbers starting from 1.

```python
N = int(input())
num = 1
for i in range(1, N+1):
    for j in range(i):
        print(num, end=" ")
        num += 1
    print()
```

* Write a program to print the factorial of N Factorial is the product of all positive integers less than or equal to N.

```python
N = int(input())
fact = 1
for i in range(1, N+1):
    fact *= i
print(fact)
```

* Write a program that reads a string and prints the count of vowels in the string.

```python
s = input()
count = 0
for ch in s:
    if ch.lower() in "aeiou":
        count += 1
print(count)
```

* Given two numbers X and N, write a program to print the sum of N terms in the given series.
Series:
$(x)^2$, $(xx)^2$, $(xxx)^2$, N terms

```
X = int(input())
N = int(input())
total = 0
num = 0
for i in range(N):
    num = num * 10 + X
    total += num ** 2
print(total)
```

* given a string write a program to print alphabets only alphabets in the given string

```
s = input()
for ch in s:
    if ch.isalpha():
        print(ch, end="")
print()
```

* given string write program that prints all the uppercases letters of the string

```
s = input()
for ch in s:
    if ch.isupper():
        print(ch, end="")
print()
```

* see about built in functions and what is the use case of builtin functions its advantages and where do we use it

Built-in functions are functions that come predefined in Python and are ready to use without importing any module. Examples: len(), sum(), max(), min(), sorted().
Use case & advantages:
Quick and easy to use without writing code from scratch
Reliable because they are tested and optimized
Saves time and improves code readability
Example:

```
numbers = [5, 10, 2, 7]
print(max(numbers))
```

* try to do a simple calculator using python

```
a = float(input())
b = float(input())
op = input()
if op == "+":
```

```python
    print(a + b)
elif op == "-":
    print(a - b)
elif op == "*":
    print(a * b)
elif op == "/":
    print(a / b)
else:
    print("Invalid operator")
```

* Find perimeter of a rectangle
```python
length = float(input("Enter length of rectangle: "))
width = float(input("Enter width of rectangle: "))
perimeter_rectangle = 2 * (length + width)
print("Perimeter of rectangle:", perimeter_rectangle)
```

* Triangle:
```python
a = float(input("Enter side a of triangle: "))
b = float(input("Enter side b of triangle: "))
c = float(input("Enter side c of triangle: "))
perimeter_triangle = a + b + c
print("Perimeter of triangle:", perimeter_triangle)
```

* Make a mini calculator get input for 2 numbers a and b get input add,sum,div,mul then if user select add them add 2 numbers and print the result
```python
x = float(input("Enter first number: "))
y = float(input("Enter second number: "))
operation = input("Choose operation (add, sub, mul, div): ")
if operation == "add":
    print("Result:", x + y)
elif operation == "sub":
    print("Result:", x - y)
elif operation == "mul":
    print("Result:", x * y)
elif operation == "div":
    print("Result:", x / y)
else:
    print("Invalid operation")
```