1.

```
import re
def track_login_attempts():
    for i in range(5):
        username = input("Enter username: ")
        pattern = r'^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9]).{8,}$'
        if re.match(pattern, username):
            print("Login Succeeded")
            break
        else:
            print(f"Login Failed,{5-i-1} attempts more")
    else:
        print("oops!no more attempts")
track_login_attempts()
```

2.

```
n=int(input("Enter the number of elements in the list:"))
list1=[]
for i in range(n):
    list1.append(int(input(f"enter the element")))
list2=[]
dup=[]
for i in list1:
    if i not in list2:
        list2.append(i)
    elif i in list2:
        dup.append(i)
print("the list after removing duplicates is:",list2)
list2.sort(reverse=True)
print("the list that is sorted in descending order is:",list2)
```

3.

```
num1=int(input("enter the number of elements to be appended in list
1"))
```

```
num2=int(input("enter the number of elements to be appended in list
2"))
lst1=[]
lst2=[]
for i in range(num1):
    lst1.append(int(input("enter the element to be appenden in list1")))
for j in range(num2):
    lst2.append(int(input("enter the element to be appenden in
list1")))
common=[]
for i in lst1:
    if i in lst2:
        common.append(i)
print(f"the common elements in the two lists are {common}")


4.
num=int(input("Enter a number:"))
count=0
for i in str(num):
    count+=int(i)
print(f"the sum of digits in the given number is: {count}")


5.
user_input = input("Enter a string: ")
words = user_input.split()
word_count = len(words)
print(f"The number of words in the given string is: {word_count}")


6.
class BankAccount:
    def __init__(self, account_holder, initial_balance=0):
        self.account_holder = account_holder
```

```
        self.balance = initial_balance
    def deposit(self, amount):
        if amount > 0:
            self.balance += amount
            print(f"Deposited: {amount}. New Balance: {self.balance}")
        else:
            print("Deposit amount must be positive.")
    def withdraw(self, amount):
        if amount > self.balance:
            print("Insufficient balance.")
        elif amount <= 0:
            print("Withdrawal amount must be positive.")
        else:
            self.balance -= amount
            print(f"Withdrawn: {amount}. New Balance: {self.balance}")
    def check_balance(self):
        print(f"Current Balance: {self.balance}")
account = BankAccount("KRISHNA", 1000)
account.check_balance()
account.deposit(500)
account.withdraw(300)
account.withdraw(1500)


7.
import re
email = input("Enter an email address: ")
pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'
if re.match(pattern, email):
    print("Valid email address.")
else:
    print("Invalid email address.")


8.
```

```
import re
text = input("Enter the text: ")
pattern = r'\b\d{10}\b|\b(?:\d{3}-\d{3}-\d{4})\b|\b(?:\(\d{3}\)
\d{3}-\d{4})\b'
phone_numbers = re.findall(pattern, text)
if phone_numbers:
    print("Extracted phone numbers:", phone_numbers)
else:
    print("No phone numbers found.")
```

9.

```
import re
text = input("Enter the text: ")
pattern = r'#\w+'
hashtags = re.findall(pattern, text)
if hashtags:
    print("Extracted hashtags:", hashtags)
else:
    print("No hashtags found.")
```

10.

```
numbers = [5, 2, 9, 1, 5, 6]
for i in range(len(numbers)):
    for j in range(len(numbers) - i - 1):
        if numbers[j] > numbers[j + 1]:
            numbers[j], numbers[j + 1] = numbers[j + 1], numbers[j]
print("Sorted list:", numbers)
```