Basic If-Else Statements:

1. Write a Python program to check if a given number is positive or negative.

In [93]:
```
num=int(input("Enter the number:"))
if num>=0:
    print(f"{num} is positive")
else:
    print(f"{num} is negative")
```

```
Enter the number:-89
-89 is negative
```

2. Create a program that determines if a person is eligible to vote based on their age.

In [14]:
```
age=int(input("Enter your age:"))
if age >= 18:
    print("You are eligible to vote")
else:
    print("You are not eligible to vote")
```

```
Enter your age:21
You are eligible to vote
```

3. Develop a program to find the maximum of two numbers using if-else statements.

In [21]:
```
num1=int(input("Enter a number:"))
num2=int(input("Enter another number:"))
if num1>num2:
    max=num1
else:
    max=num2
print(f"maximum of {num1} and {num2} is {max}")
```

```
Enter a number:78
Enter another number:56
maximum of 78 and 56 is 78
```

4. Write a Python script to classify a given year as a leap year or not.

In [31]:
```
year=int(input("Enter the year:"))
if (year%400==0) or (year%4==0 and year%100!=0):
    print(f"{year} is a leap year")
else:
    print(f"{year} is not a leap year")
```

```
Enter the year:2100
2100 is not a leap year
```

5. Create a program that checks whether a character is a vowel or a consonant

In [35]:
```
char=input("Enter a character:")
vowels="aeiouAEIOU"
if char in vowels:
    print(f"'{char}' is a vowel")
else:
    print(f"'{char}' is a consonant")
```

```
Enter a character:a
'a' is a vowel
```

6. Implement a program to determine whether a given number is even or odd.

In [39]:
```
num=int(input("Enter a number:"))
if num%2==0:
    print(f"{num} is even")
else:
    print(f"{num} is odd")
```

```
Enter a number:84
84 is even
```

7. Write a Python function to calculate the absolute value of a number without using the `abs()` function

In [42]:
```
def absolute_value(num):
    if num>=0:
        return num
    else:
        return num*-1

num=int(input("Enter a number:"))
abs_value=absolute_value(num)
print(f"Absolute value of {num} is {abs_value}")
```

```
Enter a number:-54
Absolute value of -54 is 54
```

8. Develop a program that determines the largest of three given numbers using if-else statements

In [60]:
```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))

if a>=b and a>=c:
    max=a
elif b>=c and b>=a:
    max=b
else:
    max=c
print(f"\nmaximum of {a},{b},{c} is {max}")
```

```
Enter the first number: 45
Enter the second number: 45
Enter the third number: 21

maximum of 45,45,21 is 45
```

9. Create a program that checks if a given string is a palindrome.

In [65]:
```
string=input("Enter a string:")
string=string.lower()
if string==string[::-1]:
    print(f"'{string}' is a palindrome string")
else:
    print(f"'{string}' is a not palindrome string")
```

```
Enter a string:madam
'madam' is a palindrome string
```

10. Write a Python program to calculate the grade based on a student's score.

In [76]:
```
marks=float(input("Enter your marks out of 100:"))
if marks>=33:
    grade='P'
else:
    grade='F'
print("Your grade is:",grade)
```

```
Enter your marks out of 100:90
Your grade is: P
```

Nested If-Else Statements:

11. Write a program to find the largest among three numbers using nested if-else statements.

In [72]:
```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))

if a >= b:
    if a >= c:
        largest = a
    else:
        largest = c
else:
    if b >= c:
        largest = b
    else:
        largest = c
print(f"\nmaximum of {a},{b},{c} is {largest}")
```

```
Enter the first number: 45
Enter the second number: 47
Enter the third number: 85

maximum of 45,47,85 is 85
```

12. Implement a program to determine if a triangle is equilateral, isosceles, or scalene.

In [3]:
```
side1=float(input("Enter the first side:"))
side2=float(input("Enter the second side:"))
side3=float(input("Enter the third side:"))

if side1+side2>side3 and side1+side3>side2 and side2+side3>side1:
    if side1==side2==side3:
        print("This is an equilateral triangle")
    elif side1==side2 or side2==side3 or side3==side1:
        print("This is an isosceles triangle")
    else:
        print("This is a scelene triangle")
else:
    print("The sides don't form a triangle")
```

```
Enter the first side:10
Enter the second side:12
Enter the third side:15
This is a scelene triangle
```

13. Develop a program that checks if a year is a leap year and also if it is a century year.

In [8]:
```
year=int(input("Enter the year:"))
if (year%400==0) or (year%4==0 and year%100!=0):
    if year%100==0:
        print(f"{year} is a leap year as well as century year")
    else:
        print(f"{year} is a leap year but not a century year")
else:
    print(f"{year} is not a leap year")
```

```
Enter the year:2000
2000 is a leap year as well as century year
```

14. Write a Python script to determine if a number is positive, negative, or zero.

In [92]:
```
num=int(input("Enter the number:"))
if num>=0:
    if num==0:
        print(f"{num} is zero")
    else:
        print(f"{num} is positive")
else:
    print(f"{num} is negative")
```

```
Enter the number:13
13 is positive
```

15. Create a program to check if a person is a teenager (between 13 and 19 years old

In [20]:
```
age=int(input("Enter your age:"))

if 13<=age<=19:
    print("You are a teenager")
else:
    print("You are not a teenager")
```

```
Enter your age:17
You are a teenager
```

16. Develop a program that determines the type of angle based on its measure (acute, obtuse, or right).

In [33]:
```
angle=float(input("Enter the angle:"))
if 0<angle<90:
    print(f"{angle} is an acute angle")
elif angle==90:
    print(f"{angle} is an right angle")
elif 90<angle<=180:
    print(f"{angle} is an obtuse angle")
else:
    print(f"{angle} is invalid angle")
```

```
Enter the angle:90
90.0 is an right angle
```

17. Write a Python program to calculate the roots of a quadratic equation

In [42]:
```
import math
print("Enter the coefficients 'a','b','c' of the quadratic equation of the
a=int(input("Enter a:"))
b=int(input("Enter b:"))
c=int(input("Enter c:"))
d=b**2-(4*a*c)
if d>0:
    root1=(-b+math.sqrt(d))/2*a
    root2=(-b-math.sqrt(d))/2*a
    print(f"Roots of the quadratic equation are real,unique and they are: {
elif d==0:
    root=-b/(2*a)
    print(f"Roots of the quadratic equation are real,equal and they are: {r
else:
    real_part = -b / (2*a)
    imaginary_part = math.sqrt(abs(d)) / (2*a)
    root1 = complex(real_part, imaginary_part)
    root2 = complex(real_part, -imaginary_part)
    print(f"The roots are complex: {root1} and {root2}")
```

```
Enter the coefficients 'a','b','c' of the quadratic equation of the form a
x^2+bx+c.
Enter a:1
Enter b:-4
Enter c:4
Roots of the quadratic equation are real,equal and they are: 2.0
```

18. Implement a program to determine the day of the week based on a user-provided number (1 for Monday, 2 for Tuesday, etc.).

In [47]:
```
day_number=int(input("Enter a number from 1 to 7:"))
if 1<=day_number<=7:
    week=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Su
    day=week[day_number-1]
    print(f"The day of the week is {day}")
else:
    print("Invalid choice, enter number from 1 to 7")
```

```
Enter a number from 1 to 7:7
The day of the week is Sunday
```

19. Create a program that determines if a year is a leap year and also if it is evenly divisible by 400

In [52]:
```
year=int(input("Enter the year:"))

if (year%400==0) or (year%4==0 and year%100!=0):
    print(f"{year} is a leap year")
    if year%400==0:
        print(f"{year} is evenly divisible by 400")
    else:
        print(f"{year} is not evenly divisible by 400")
else:
    print(f"{year} is not a leap year")
```

```
Enter the year:2000
2000 is a leap year
2000 is evenly divisible by 400
```

20. Develop a program that checks if a given number is prime or not using nested if-else statements.

In [74]:
```
number = int(input("Enter a number: "))

if number > 1:
    is_prime = True
    for i in range(2, number):
        if number % i == 0:
            is_prime = False
            break

    if is_prime:
        print(f"{number} is a prime number.")
    else:
        print(f"{number} is not a prime number.")
else:
    print(f"{number} is not a prime number (prime numbers are greater than
```

```
Enter a number: 5
5 is a prime number.
```

Elif Statements:

21. Write a Python program to assign grades based on different ranges of scores using elif statements.

In [77]:
```
marks=float(input("Enter your marks out of 100:"))
if marks>=90:
    grade='A+'
elif marks >=80:
    grade='A'
elif marks>=70:
    grade='B'
elif marks>=60:
    grade='C'
elif marks>=50:
    grade='D'
else:
    grade='F'
print("Your grade is:",grade)
```

```
Enter your marks out of 100:94.65
Your grade is: A+
```

22. Implement a program to determine the type of a triangle based on its angles.

In [80]:
```
angle1 = int(input("Enter the first angle of the triangle: "))
angle2 = int(input("Enter the second angle of the triangle: "))
angle3 = int(input("Enter the third angle of the triangle: "))

if angle1 + angle2 + angle3 == 180:
    if angle1 == angle2 == angle3:
        triangle_type = "Equilateral"
    elif angle1 == angle2 or angle1 == angle3 or angle2 == angle3:
        triangle_type = "Isosceles"
    else:
        triangle_type = "Scalene"
else:
    triangle_type = "Not a valid triangle"


print(f"The triangle is a {triangle_type} triangle.")
```

```
Enter the first angle of the triangle: 45
Enter the second angle of the triangle: 45
Enter the third angle of the triangle: 90
The triangle is a Isosceles triangle.
```

23. Develop a program to categorize a given person's BMI into underweight, normal, overweight, or obese using elif statements.

In [85]:
```
height=float(input("Enter your height in cm:"))
weight=float(input("Enter your weight in kg:"))
bmi=weight/((height/100)**2)
print("Your BMI is:",bmi)
if bmi<=18.5:
    print("You are underweight")
elif 18.5<bmi<25:
    print("You are normal")
elif 25<bmi<30:
    print("You are overweight")
else:
    print("You are obese")
```

```
Enter your height in cm:165
Enter your weight in kg:60
Your BMI is: 22.03856749311295
You are normal
```

24. Create a program that determines whether a given number is positive, negative, or zero using elif statements.

In [95]:
```
num=int(input("Enter the number:"))
if num>0:
    print(f"{num} is positive")
elif num==0:
    print(f"{num} is zero")
else:
    print(f"{num} is negative")
```

```
Enter the number:0
0 is zero
```

25. Write a Python script to determine the type of a character (uppercase, lowercase, or special) using elif statements

In [103]:
```
char=input("Enter a character:")
if char.isalpha():
    if char.isupper():
        print(f"'{char}' is uppercase")
    elif char.islower():
        print(f"'{char}' is lowercase")
else:
    print(f"'{char}' is special")
```

```
Enter a character:*
'*' is special
```

26. Implement a program to calculate the discounted price based on different purchase amounts using elif statements

In [107]:
```
org_amount=float(input("Enter your purchase amount:"))
discount_rate=0
discounted_price=org_amount

if org_amount>=10000:
    discount_rate=.4 #40% discount for purchase over 10000
elif org_amount>=7000:
    discount_rate=.3 #30% discount for purchase over 7000
elif org_amount>=4000:
    discount_rate=.2 #20% discount for purchase over 4000
elif org_amount>=2000:
    discount_rate=.1 #10% discount for purchase over 2000
if discount_rate>0:
    discounted_price=org_amount-(org_amount*discount_rate)

print(f"Original Price: ₹{org_amount:}")
print(f"You get {discount_rate*100}% discount and Discounted Price: ₹{disco
```

```
Enter your purchase amount:15000
Original Price: ₹15000.0
You get 40.0% discount and Discounted Price: ₹9000.0
```

27. Develop a program to calculate the electricity bill based on different consumption slabs using elif statements.

In [111]:
```
consumption = float(input("Enter your electricity consumption (in kWh): "))

bill_amount = 0.0
rate_per_kwh = 0.0
if consumption <= 50:
    rate_per_kwh = 3.00
elif consumption <= 100:
    rate_per_kwh = 4.50
elif consumption <= 200:
    rate_per_kwh = 6.00
else:
    rate_per_kwh = 7.50

bill_amount = consumption * rate_per_kwh

print(f"Electricity Consumption: {consumption} kWh")
print(f"Rate per kWh: ₹{rate_per_kwh:.2f}")
print(f"Electricity Bill: ₹{bill_amount:.2f}")
```

```
Enter your electricity consumption (in kWh): 154
Electricity Consumption: 154.0 kWh
Rate per kWh: ₹6.00
Electricity Bill: ₹924.00
```

28. Create a program to determine the type of quadrilateral based on its angles and sides using elif statements.

In [1]:
```
#input of angles

angle1 = float(input("Enter the first angle (in degrees): "))
angle2 = float(input("Enter the second angle (in degrees): "))
angle3 = float(input("Enter the third angle (in degrees): "))
angle4 = float(input("Enter the fourth angle (in degrees): "))

#input of sides

side1 = float(input("Enter the length of the first side: "))
side2 = float(input("Enter the length of the second side: "))
side3 = float(input("Enter the length of the third side: "))
side4 = float(input("Enter the length of the fourth side: "))

if angle1+angle2+angle3+angle4==360:
    if angle1 == angle2 == angle3 == angle4 == 90:
        if side1==side2==side3==side4:
            quadrilateral_type = "Square"
        elif side1==side3 and side2==side4:
            quadrilateral_type = "Rectangle"
    elif side1==side2==side3==side4:
        if angle1 == angle3 and angle2 == angle4:
            quadrilateral_type = "Rhombus"
    elif side1==side3 and side2==side4:
        quadrilateral_type = "Parallelogram"
    else:
        quadrilateral_type = "Quadrilateral"
    print(f"The quadrilateral is a {quadrilateral_type}.")
else:
    print("The entered angles do not form a quadrilateral")
```

```
Enter the first angle (in degrees): 90
Enter the second angle (in degrees): 90
Enter the third angle (in degrees): 90
Enter the fourth angle (in degrees): 90
Enter the length of the first side: 10
Enter the length of the second side: 15
Enter the length of the third side: 10
Enter the length of the fourth side: 15
The quadrilateral is a Rectangle.
```

29. Write a Python script to determine the season based on a user-provided month using elif statements.

In [9]:
```
'''
season           months
Summer Season   March to May
Monsoon Season  June to September
Autumn Season   October to November
Winter Season   December to February
'''

month = input("Enter a month: ").lower()

if month in ["december", "january", "february"]:
    season = "Winter"
elif month in ["march", "april", "may"]:
    season = "Summer"
elif month in ["june", "july", "august","september"]:
    season = "Monsoon"
elif month in ["october", "november"]:
    season = "Autumn"
else:
    season = "Invalid Month"

print(f"The season for {month.capitalize()} is {season}.")
```

```
Enter a month: july
The season for July is Monsoon.
```

30. Implement a program to determine the type of a year (leap or common) and month (30 or 31 days) using elif statements.

In [11]:
```
year = int(input("Enter a year: "))

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    year_type = "Leap Year"
else:
    year_type = "Common Year"

month = int(input("Enter a month (1-12): "))

if month == 2:
    if year_type == "Leap Year":
        days_in_month = 29
    else:
        days_in_month = 28
elif month in [4, 6, 9, 11]:
    days_in_month = 30
else:
    days_in_month = 31
month_name=["January", "February", "March", "April","May", "June", "July","
print(f"{year} is a {year_type}.")
print(f"Month {month} i.e {month_name[month-1]} has {days_in_month} days.")
```

```
Enter a year: 2002
Enter a month (1-12): 7
2002 is a Common Year.
Month 7 i.e July has 31 days.
```

Basic Level:

1. Write a Python program that checks if a given number is positive, negative, or zero.

In [12]:
```
num=int(input("Enter the number:"))
if num>0:
    print(f"{num} is positive")
elif num==0:
    print(f"{num} is zero")
else:
    print(f"{num} is negative")
```

```
Enter the number:45
45 is positive
```

2. Create a program to determine if a person is eligible to vote based on their age

In [13]:
```
age=int(input("Enter your age:"))
if age >= 18:
    print("You are eligible to vote")
else:
    print("You are not eligible to vote")
```

```
Enter your age:21
You are eligible to vote
```

3. Write a program to find the maximum of two given numbers using conditional statements.

In [15]:
```
num1=int(input("Enter a number:"))
num2=int(input("Enter another number:"))
if num1>num2:
    max=num1
else:
    max=num2
print(f"maximum of {num1} and {num2} is {max}")
```

```
Enter a number:12
Enter another number:-78
maximum of 12 and -78 is 12
```

4. Develop a program that calculates the grade of a student based on their exam

score.

In [16]:
```
marks=float(input("Enter your marks out of 100:"))
if marks>=90:
    grade='A+'
elif marks >=80:
    grade='A'
elif marks>=70:
    grade='B'
elif marks>=60:
    grade='C'
elif marks>=50:
    grade='D'
else:
    grade='F'
print("Your grade is:",grade)
```

```
Enter your marks out of 100:85
Your grade is: A
```

5. Create a program that checks if a year is a leap year or not.

In [18]:
```
year=int(input("Enter the year:"))
if (year%400==0) or (year%4==0 and year%100!=0):
    print(f"{year} is a leap year")
else:
    print(f"{year} is not a leap year")
```

```
Enter the year:2004
2004 is a leap year
```

6. Write a program to classify a triangle based on its sides' lengths.

In [20]:
```
side1=float(input("Enter the first side:"))
side2=float(input("Enter the second side:"))
side3=float(input("Enter the third side:"))

if side1+side2>side3 and side1+side3>side2 and side2+side3>side1:
    if side1==side2==side3:
        print("This is an equilateral triangle")
    elif side1==side2 or side2==side3 or side3==side1:
        print("This is an isosceles triangle")
    else:
        print("This is a scelene triangle")
else:
    print("The sides don't form a triangle")
```

```
Enter the first side:20
Enter the second side:20
Enter the third side:20
This is an equilateral triangle
```

7. Build a program that determines the largest of three given numbers.

In [21]:
```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))

if a>=b and a>=c:
    max=a
elif b>=c and b>=a:
    max=b
else:
    max=c
print(f"\nmaximum of {a},{b},{c} is {max}")
```

```
Enter the first number: 15
Enter the second number: 45
Enter the third number: -85

maximum of 15,45,-85 is 45
```

8. Develop a program that checks whether a character is a vowel or a consonant.

In [23]:
```
char=input("Enter a character:")
vowels="aeiouAEIOU"
if char in vowels:
    print(f"'{char}' is a vowel")
else:
    print(f"'{char}' is a consonant")
```

```
Enter a character:b
'b' is a consonant
```

9. Create a program to calculate the total cost of a shopping cart based on discounts.

In [24]:
```
org_amount=float(input("Enter your purchase amount:"))
discount_rate=0
discounted_price=org_amount

if org_amount>=10000:
    discount_rate=.4 #40% discount for purchase over 10000
elif org_amount>=7000:
    discount_rate=.3 #30% discount for purchase over 7000
elif org_amount>=4000:
    discount_rate=.2 #20% discount for purchase over 4000
elif org_amount>=2000:
    discount_rate=.1 #10% discount for purchase over 2000
if discount_rate>0:
    discounted_price=org_amount-(org_amount*discount_rate)

print(f"Original Price: ₹{org_amount:}")
print(f"You get {discount_rate*100}% discount and Discounted Price: ₹{disco
```

```
Enter your purchase amount:15420
Original Price: ₹15420.0
You get 40.0% discount and Discounted Price: ₹9252.0
```

10. Write a program that checks if a given number is even or odd.

In [25]:
```
num=int(input("Enter a number:"))
if num%2==0:
    print(f"{num} is even")
else:
    print(f"{num} is odd")
```

```
Enter a number:13
13 is odd
```

Intermediate Level:

11. Write a program that calculates the roots of a quadratic equation .

In [42]:
```
import math
print("Enter the coefficients 'a','b','c' of the quadratic equation of the
a=int(input("Enter a:"))
b=int(input("Enter b:"))
c=int(input("Enter c:"))
d=b**2-(4*a*c)
if d>0:
    root1=(-b+math.sqrt(d))/2*a
    root2=(-b-math.sqrt(d))/2*a
    print(f"Roots of the quadratic equation are real,unique and they are: {
elif d==0:
    root=-b/(2*a)
    print(f"Roots of the quadratic equation are real,equal and they are: {r
else:
    real_part = -b / (2*a)
    imaginary_part = math.sqrt(abs(d)) / (2*a)
    root1 = complex(real_part, imaginary_part)
    root2 = complex(real_part, -imaginary_part)
    print(f"The roots are complex: {root1:.3f} and {root2:.3f}")
```

```
Enter the coefficients 'a','b','c' of the quadratic equation of the form a
x^2+bx+c.
Enter a:1
Enter b:4
Enter c:2
Roots of the quadratic equation are real,unique and they are: -0.586 and -
3.414
```

12. Create a program that determines the day of the week based on the day number (1-7).

In [44]:
```
day_number=int(input("Enter a number from 1 to 7:"))
if 1<=day_number<=7:
    week=["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Su
    day=week[day_number-1]
    print(f"The day of the week is {day}")
else:
    print("Invalid choice, enter number from 1 to 7")
```

```
Enter a number from 1 to 7:5
The day of the week is Friday
```

13. Develop a program that calculates the factorial of a given number using recursion.

In [4]:
```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
num=int(input("Enter the number whose factorial is to be found:"))
if num>=0:
    fact=factorial(num)
    print(f"factorial of {num} is: {fact}")
else:
    print("Factorial can be found for numbers greater than or equal to 0")
```

```
Enter the number whose factorial is to be found:6
factorial of 6 is: 720
```

14. Write a program to find the largest among three numbers without using the `max()` function.

In [5]:
```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))

if a>=b and a>=c:
    max=a
elif b>=c and b>=a:
    max=b
else:
    max=c
print(f"\nmaximum of {a},{b},{c} is {max}")
```

```
Enter the first number: 50
Enter the second number: 12
Enter the third number: 75

maximum of 50,12,75 is 75
```

15. Create a program that simulates a basic ATM transaction menu

In [8]:
```
account_balance = 10000

# Function to display account balance
def check_balance():
    print(f"Your account balance is ₹{account_balance:.2f}")

# Function to deposit money
def deposit():
    global account_balance
    amount = float(input("Enter the amount to deposit: ₹"))
    if amount > 0:
        account_balance += amount
        print(f"Successfully deposited ₹{amount:.2f}")
        check_balance()
    else:
        print("Invalid amount. Please enter a positive number.")

# Function to withdraw money
def withdraw():
    global account_balance
    amount = float(input("Enter the amount to withdraw: ₹"))
    if amount > 0 and amount <= account_balance:
        account_balance -= amount
        print(f"Withdrew ₹{amount:.2f}")
        check_balance()
    elif amount > account_balance:
        print("Insufficient funds.")
    else:
        print("Invalid amount. Please enter a positive number.")

# Main ATM menu
while True:
    print("\n********Welcome to ATM Service**************")
    print("1. Check Balance")
    print("2. Deposit")
    print("3. Withdraw")
    print("4. Exit")
    print("*****************************************")

    choice = input("Enter your choice (1/2/3/4): ")

    if choice == "1":
        check_balance()
    elif choice == "2":
        deposit()
    elif choice == "3":
        withdraw()
    elif choice == "4":
        print("Thank you for using the ATM. Goodbye!")
        break
    else:
        print("Invalid choice. Please select a valid option (1/2/3/4).")
```

```
********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 5
Invalid choice. Please select a valid option (1/2/3/4).

********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 1
Your account balance is ₹10000.00

********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 2
Enter the amount to deposit: ₹502
Successfully deposited ₹502.00
Your account balance is ₹10502.00

********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 3
Enter the amount to withdraw: ₹1002
Withdrew ₹1002.00
Your account balance is ₹9500.00

********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 1
Your account balance is ₹9500.00

********Welcome to ATM Service**************
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
***********************************************
Enter your choice (1/2/3/4): 4
Thank you for using the ATM. Goodbye!
```

16. Build a program that checks if a given string is a palindrome or not.

In [9]:
```
string=input("Enter a string:")
string=string.lower()
if string==string[::-1]:
    print(f"'{string}' is a palindrome string")
else:
    print(f"'{string}' is a not palindrome string")
```

```
Enter a string:madam
'madam' is a palindrome string
```

17. Write a program that calculates the average of a list of numbers, excluding the smallest and largest values.

In [12]:
```
sample_list=[22,23,45,78,95,45,12,3,5,4]
print("Original list:\n",sample_list)
if len(sample_list) < 3:
    print("The list should have at least three numbers.")
sample_list.remove(min(sample_list))
sample_list.remove(max(sample_list))
average = sum(sample_list) / len(sample_list)
print("List excluding the smallest and largest values:\n",sample_list)
print("Average of list of number excluding smallest and largest values:",av
```

```
Original list:
 [22, 23, 45, 78, 95, 45, 12, 3, 5, 4]
List excluding the smallest and largest values:
 [22, 23, 45, 78, 45, 12, 5, 4]
Average of list of number excluding smallest and largest values: 29.25
```

18. Develop a program that converts a given temperature from Celsius to Fahrenheit.

In [20]:
```
celsius=float(input("Enter the temperature in degree Celsius:"))
fahrenheit=32+(9*celsius)/5
print(f"{celsius}°C = {fahrenheit}°F")
```

```
Enter the temperature in degree Celsius:95
95.0°C = 203.0°F
```

19. Create a program that simulates a basic calculator for addition, subtraction, multiplication, and division

In [28]:
```
num1=float(input("Enter the first number:"))
num2=float(input("Enter the second number:"))
print("\nPress 1 for addition")
print("Press 2 for subtraction")
print("Press 3 for multiplication")
print("Press 4 for division")
choice=int(input())
if choice==1:
    print(f"{num1}+{num2}={num1+num2}")
elif choice==2:
    print(f"{num1}-{num2}={num1-num2}")
elif choice==3:
    print(f"{num1}*{num2}={num1*num2}")
elif choice==4:
    if num2==0:
        print("Can't divide by 0")
    else:
        print(f"{num1}/{num2}={num1/num2}")
else:
    print("Invalid choice")
```

```
Enter the first number:15
Enter the second number:12

Press 1 for addition
Press 2 for subtraction
Press 3 for multiplication
Press 4 for division
2
15.0-12.0=3.0
```

20. Write a program that determines the roots of a cubic equation using the Cardano formula

In [43]:
```
import math

def cubic_roots(a, b, c, d):
    if a == 0:
        return "Coefficient 'a' cannot be zero."

    p = (3 * a * c - b**2) / (3 * a**2)
    q = (2 * b**3 - 9 * a * b * c + 27 * a**2 * d) / (27 * a**3)

    discriminant = (q**2 / 4) + (p**3 / 27)

    if discriminant > 0:
        u = (-q / 2 + math.sqrt(discriminant))**(1/3)
        v = (-q / 2 - math.sqrt(discriminant))**(1/3)
        root1 = u + v - b / (3 * a)
        return [root1]

    elif discriminant == 0:
        root1 = (-q / 2)**(1/3) - b / (3 * a)
        root2 = (-q / 2)**(1/3) - b / (3 * a)
        return [root1, root2]

    else:  # Discriminant is less than zero
        r = math.sqrt(-(p**3) / 27)
        theta = math.acos(-q / (2 * r))
        root1 = 2 * math.sqrt(-p / 3) * math.cos(theta / 3) - b / (3 * a)
        root2 = 2 * math.sqrt(-p / 3) * math.cos((theta + 2 * math.pi) / 3)
        root3 = 2 * math.sqrt(-p / 3) * math.cos((theta - 2 * math.pi) / 3)
        return [root1, root2, root3]

try:
    a = float(input("Enter the coefficient of x^3 (a): "))
    b = float(input("Enter the coefficient of x^2 (b): "))
    c = float(input("Enter the coefficient of x (c): "))
    d = float(input("Enter the constant term (d): "))

    roots = cubic_roots(a, b, c, d)

    print("Roots of the cubic equation:")
    for i, root in enumerate(roots):
        print(f"Root {i+1}: {root:.6f}")

except ValueError:
    print("Invalid input. Please enter valid numerical coefficients.")
```

```
Enter the coefficient of x^3 (a): 1
Enter the coefficient of x^2 (b): -23
Enter the coefficient of x (c): 142
Enter the constant term (d): -120
Roots of the cubic equation:
Root 1: 12.000000
Root 2: 1.000000
Root 3: 10.000000
```

Advanced Level:

21. Create a program that calculates the income tax based on the user's income and tax brackets.

In [4]:
```
'''
AMOUNT                                  INCOME TAX RATE
Up to ₹2,50,000                          0%
₹2,50,001 – ₹5,00,000                  5% above ₹2,50,000
₹5,00,001 – ₹7,50,000                  10% above ₹5,00,000 + ₹12,500
₹7,50,001 – ₹10,00,000                 15% above ₹7,50,000 + ₹37,500
₹10,00,001 – ₹12,50,000               20% above ₹10,00,000 + ₹75,000
₹12,50,001 – ₹15,00,000               25% above ₹12,50,000 + ₹1,25,000
Above ₹15,00,001                          30% above ₹15,00,000 + ₹1,87,500
'''


def income_tax(income):
    if income<=250000:
        tax=0
    elif income<=500000:
        tax=(income-250000)*.05
    elif income<=750000:
        tax=(income-500000)*.1+12500
    elif income<=1000000:
        tax=(income-750000)*.15+37500
    elif income<=1250000:
        tax=(income-1000000)*.2+75000
    elif income<=1500000:
        tax=(income-1250000)*.25+125000
    else:
        tax=(income-1500000)*.3+187500
    return tax

income=float(input("Enter your annual income:₹"))
tax=income_tax(income)
print(f"You need to pay ₹{tax:.3f} on your annual income of ₹{income:.3f}")
```

```
Enter your annual income:₹720000
You need to pay ₹34500.000 on your annual income of ₹720000.000
```

22. Write a program that simulates a rock-paper-scissors game against the computer

In [1]:
```
import random

def get_user_choice():
    while True:
        user_choice = input("Enter your choice (rock, paper, or scissors):
        if user_choice in ["rock", "paper", "scissors"]:
            return user_choice
        else:
            print("Invalid choice. Please choose rock, paper, or scissors."

def get_computer_choice():
    choices = ["rock", "paper", "scissors"]
    return random.choice(choices)

def determine_winner(user_choice, computer_choice):
    if user_choice == computer_choice:
        return "It's a tie!"
    elif user_choice == "rock" and computer_choice == "scissors":
        return "Rock smashes scissors!,You win!"
    elif user_choice == "paper" and computer_choice == "rock":
        return "Paper covers rock! You win!"
    elif user_choice == "scissors" and computer_choice == "paper":
        return "Scissors cuts paper! You win!"
    else:
        return "Computer wins!"

while True:
    user_choice = get_user_choice()
    computer_choice = get_computer_choice()
    print(f"You chose '{user_choice}'.")
    print(f"The computer chose '{computer_choice}'.")

    result = determine_winner(user_choice, computer_choice)
    print(result)

    play_again = input("Do you want to play again? (yes/no): ").lower()
    if play_again != "yes":
        break

print("Thanks for playing!")
```

```
Enter your choice (rock, paper, or scissors): rock
You chose 'rock'.
The computer chose 'scissors'.
Rock smashes scissors!,You win!
Do you want to play again? (yes/no): no
Thanks for playing!
```

23. Develop a program that generates a random password based on user preferences (length, complexity).

In [4]:
```
import random
import string

print("******************Welcome to the Random Password Generator!********

length = int(input("Enter the desired password length: "))
use_lowercase = input("Include lowercase letters? (y/n): ").lower() == 'y'
use_uppercase = input("Include uppercase letters? (y/n): ").lower() == 'y'
use_digits = input("Include digits? (y/n): ").lower() == 'y'
use_special_chars = input("Include special characters? (y/n): ").lower() ==

characters = ''

if use_lowercase:
    characters += string.ascii_lowercase
if use_uppercase:
    characters += string.ascii_uppercase
if use_digits:
    characters += string.digits
if use_special_chars:
    characters += string.punctuation

if len(characters) == 0:
    print("Please select at least one character type.")
else:
    password = ''.join(random.choice(characters) for _ in range(length))
    print(f"\nGenerated Password: {password}")
```

```
******************Welcome to the Random Password Generator!**************
************
Enter the desired password length: 12
Include lowercase letters? (y/n): y
Include uppercase letters? (y/n): y
Include digits? (y/n): y
Include special characters? (y/n): y

Generated Password: OV!06x["$l<i
```

24. Create a program that implements a simple text-based adventure game with branching scenarios.

In [8]:
```
answer_yes = ["Yes", "Y", "yes", "y"]
answer_no = ["No", "N", "no", "n"]

print("""
WELCOME! LET'S START THE ADVENTURE

You are standing outside of your house and you see a man running towards yo

Will you provide shelter to him. (Yes / No)
""")

ans1 = input(">>")

if ans1 in answer_yes:
    print("\nAfter 2 minutes, the Police came to your house, and ask you th

    ans2 = input(">>")

    if ans2 in answer_yes:
        print("\nYou are an honest person. He was a thief & You won the Gam

    elif ans2 in answer_no:
        print("\nYou helped a thief. Now, go to Jail. GAME OVER")

    else:
        print("\nYou typed the wrong input. GOODBYE!")

elif ans1 in answer_no:
    print("\nNow, he is trying to kill you. Will, you knock him down? (Yes

    ans3 = input(">>")

    if ans3 in answer_yes:
        print("\nCongrats! He was a thief & You helped the police to catch

    elif ans3 in answer_no:
        print("\nSorry! You are dead. He was a thief & He killed you. GAME

    else:
        print("\nYou typed the wrong input. GOODBYE!")

else:
    print("\nYou typed the wrong input. GOODBYE!")
```

```
WELCOME! LET'S START THE ADVENTURE

You are standing outside of your house and you see a man running towards y
ou and asking for urgent shelter.

Will you provide shelter to him. (Yes / No)

>>no

Now, he is trying to kill you. Will, you knock him down? (Yes / No)

>>yes

Congrats! He was a thief & You helped the police to catch him with your br
avery.
```

25. Build a program that solves a linear equation for x, considering different cases.

In [13]:
```python
print("**********************Linear Equation solver**********************")
print("Enter the coefficients'a' and 'b' in the equation of form ax+b=0")
a=float(input("a>>"))
b=float(input("b>>"))
if a==0:
    if b==0:
        print(f"The equation {a}x + {b} = 0 has infinitely many solutions."
    else:
        print(f"The equation {a}x + {b} = 0 has no solution.")
else:
    x=-b/a
    print(f"The solution of the equation {a}x + {b} = 0 is x={x}")
```

```
**********************Linear Equation solver**********************
Enter the coefficients'a' and 'b' in the equation of form ax+b=0
a>>25
b>>2
The solution of the equation 25.0x + 2.0 = 0 is x=-0.08
```

26. Write a program that simulates a basic quiz game with multiple-choice questions and scoring.

In [29]:
```
print("****************Welcome to the Quiz game********************")
print("\n\nThere will be 4 options a,b,c,d, you have to select the correct
options={1:'b',2:'b',3:'b',4:'c',5:'c',6:'c',7:'b',8:'d',9:'a',10:'d'}
score=0
def check(q,choice):
    global score
    if choice==options[q]:
        score+=1
        print("Correct answer!")
    else:
        print("Wrong answer!")


print("""
1._______ is the smallest unit of data in a computer ?

a)Gigabyte
b)Bit
c)Byte
d)Terabyte
""")
choice=input("your choice>>").lower()
check(1,choice)

print("""
2.Which of the following is NOT an anti-virus software ?

a)Avast
b)Linux
c)Norton
d)Kaspersky
""")
choice=input("your choice>>").lower()
check(2,choice)

print("""
3.In the context of digital computer, which of the following pairs of digit

a)3 and 4
b)0 and 1
c)2 and 3
d)1 and 2
""")
choice=input("your choice>>").lower()
check(3,choice)

print("""
4.Which unit of the computer is considered as the brain of the computer ?

a)Memory unit
b)Input unit
c)CPU
d)Output unit
""")
choice=input("your choice>>").lower()
check(4,choice)

print("""
5.What is the full form of PROM ?

a)Program read-only memory
```

```
b)Primary read-only memory
c)Programmable read-only memory
d)Program read-output memory

""")
choice=input("your choice>>").lower()
check(5,choice)

print("""
6.In the context of computing, what is the full form of URL ?

a)Undistributed Resource Locator
b)Unified Resource Locator
c)Uniform Resource Locator
d)Uniform Region Locator

""")
choice=input("your choice>>").lower()
check(6,choice)

print("""
7.Which of the following is an input device used to enter motion data in co

a)Monitor
b)Trackball
c)Plotter
d)Joystick

""")
choice=input("your choice>>").lower()
check(7,choice)

print("""
8.In the context of computing, a byte is equal to _____ bits ?

a)4
b)16
c)24
d)8
""")
choice=input("your choice>>").lower()
check(8,choice)

print("""
9._____ is a small, portable flash memory card that plugs into a computer's

a)Flash drive
b)CD-RW
c)DVD-ROM
d)CD-ROM
""")
choice=input("your choice>>").lower()
check(9,choice)

print("""
10.Which of the following devices is NOT used to enter data into a computer

a)Mouse
b)Keyboard
c)Scanner
d)Monitor
```

```
""")
choice=input("your choice>>").lower()
check(10,choice)
print(f"\nYour score is {score}/10")
```

```
***************Welcome to the Quiz game*******************


There will be 4 options a,b,c,d, you have to select the correct option.

1._______ is the smallest unit of data in a computer ?

a)Gigabyte
b)Bit
c)Byte
d)Terabyte

your choice>>b
Correct answer!

2.Which of the following is NOT an anti-virus software ?

a)Avast
b)Linux
```

27. Develop a program that determines whether a given year is a prime number or not.

In [33]:
```
year=int(input("Enter the year>>"))
count=0
for i in range(1,int(year/2)):
    if year%i==0:
        count+=1
if count==1:
    print(f"{year} is a prime number")
else:
    print(f"{year} is not a prime number")
```

```
Enter the year>>2113
2113 is a prime number
```

28. Create a program that sorts three numbers in ascending order using conditional statements.

In [36]:
```
num1=int(input("Enter first number>>"))
num2=int(input("Enter second number>>"))
num3=int(input("Enter third number>>"))
if num1 <= num2 <= num3:
    smallest = num1
    middle = num2
    largest = num3
elif num1 <= num3 <= num2:
    smallest = num1
    middle = num3
    largest = num2
elif num2 <= num1 <= num3:
    smallest = num2
    middle = num1
    largest = num3
elif num2 <= num3 <= num1:
    smallest = num2
    middle = num3
    largest = num1
elif num3 <= num1 <= num2:
    smallest = num3
    middle = num1
    largest = num2
else:
    smallest = num3
    middle = num2
    largest = num1
print(f"Sorted numbers in ascending order: {smallest} < {middle} < {largest
```

```
Enter first number>>45
Enter second number>>2
Enter third number>>89
Sorted numbers in ascending order: 2 < 45 < 89
```

29. Build a program that determines the roots of a quartic equation using numerical methods.

In [37]:
```
import math
print("Enter the coefficients 'a','b','c' of the quadratic equation of the
a=int(input("Enter a:"))
b=int(input("Enter b:"))
c=int(input("Enter c:"))
d=b**2-(4*a*c)
if d>0:
    root1=(-b+math.sqrt(d))/2*a
    root2=(-b-math.sqrt(d))/2*a
    print(f"Roots of the quadratic equation are real,unique and they are: {
elif d==0:
    root=-b/(2*a)
    print(f"Roots of the quadratic equation are real,equal and they are: {r
else:
    real_part = -b / (2*a)
    imaginary_part = math.sqrt(abs(d)) / (2*a)
    root1 = complex(real_part, imaginary_part)
    root2 = complex(real_part, -imaginary_part)
    print(f"The roots are complex: {root1} and {root2}")
```

```
Enter the coefficients 'a','b','c' of the quadratic equation of the form a
x^2+bx+c.
Enter a:1
Enter b:-7
Enter c:12
Roots of the quadratic equation are real,unique and they are: 4.0 and 3.0
```

30. Write a program that calculates the BMI (Body Mass Index) and provides health recommendations based on the user's input.

In [38]:
```
height=float(input("Enter your height in cm:"))
weight=float(input("Enter your weight in kg:"))
bmi=weight/((height/100)**2)
print("Your BMI is:",bmi)
if bmi<=18.5:
    print("You are underweight.It's important to maintain a balanced diet a
elif 18.5<bmi<25:
    print("You are normal,You have a healthy weight. Keep up the good work
elif 25<bmi<30:
    print("You are overweight.Consider adopting a healthier lifestyle with
else:
    print("You are obese.It's crucial to consult a healthcare professional
```

```
Enter your height in cm:165
Enter your weight in kg:60
Your BMI is: 22.03856749311295
You are normal,You have a healthy weight. Keep up the good work with a bal
anced diet and regular exercise.
```

Challenge Level:

31. Create a program that validates a password based on complexity rules (length, characters, etc.)

In [41]:
```
import re

def validate_password(password):

    if len(password) < 8:
        return "Password must be at least 8 characters long."


    if not re.search(r'[A-Z]', password):
        return "Password must contain at least one uppercase letter."


    if not re.search(r'[a-z]', password):
        return "Password must contain at least one lowercase letter."

    if not re.search(r'[0-9]', password):
        return "Password must contain at least one digit."

    if not re.search(r'[!@#$%^&*]', password):
        return "Password must contain at least one special character (!@#$%

    return "Password is valid."


password = input("Enter a password: ")


validation_result = validate_password(password)
print(validation_result)
```

```
Enter a password: JB98U5jJHIK&^%&*
Password is valid.
```

32. Develop a program that performs matrix addition and subtraction based on user input

In [57]:
```
def matrix_addition(matrix1, matrix2):
    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None

    result = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix1[0])):
            row.append(matrix1[i][j] + matrix2[i][j])
        result.append(row)
    return result

def matrix_subtraction(matrix1, matrix2):
    if len(matrix1) != len(matrix2) or len(matrix1[0]) != len(matrix2[0]):
        return None

    result = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix1[0])):
            row.append(matrix1[i][j] - matrix2[i][j])
        result.append(row)
    return result

def display_matrix(matrix):
    for row in matrix:
        row_str = ""
        for element in row:
            row_str += str(element) + "  "
        print(row_str)

rows1 = int(input("Enter the number of rows for first matrix: "))
cols1 = int(input("Enter the number of columns for second matrix: "))
rows2 = int(input("Enter the number of rows for second matrix: "))
cols2 = int(input("Enter the number of columns for second matrix: "))
if rows1!=rows2 and cols1!=cols2:
    print("Matrix addition and subtraction is not possible due to incompara
else:
    print("Enter the elements of the first matrix:")
    matrix1 = []
    for i in range(rows1):
        row = []
        for j in range(cols1):
            element = int(input(f"Enter element at row {i + 1}, column {j +
            row.append(element)
        matrix1.append(row)
    print("matrix 1:")
    display_matrix(matrix1)
    print()

    print("Enter the elements of the second matrix:")
    matrix2 = []
    for i in range(rows2):
        row = []
        for j in range(cols2):
            element = int(input(f"Enter element at row {i + 1}, column {j +
            row.append(element)
        matrix2.append(row)
    print("matrix 2:")
    display_matrix(matrix2)
    print()
```

```
    result_addition = matrix_addition(matrix1, matrix2)
    if result_addition is not None:
        print("\nMatrix Addition:")
        display_matrix(result_addition)
    else:
        print("\nMatrix addition is not possible due to different dimension

    result_subtraction = matrix_subtraction(matrix1, matrix2)
    if result_subtraction is not None:
        print("\nMatrix Subtraction:")
        display_matrix(result_subtraction)
    else:
        print("\nMatrix subtraction is not possible due to different dimens
```

```
Enter the number of rows for first matrix: 2
Enter the number of columns for second matrix: 2
Enter the number of rows for second matrix: 2
Enter the number of columns for second matrix: 2
Enter the elements of the first matrix:
Enter element at row 1, column 1: 5
Enter element at row 1, column 2: 6
Enter element at row 2, column 1: 2
Enter element at row 2, column 2: 3
matrix 1:
5  6
2  3

Enter the elements of the second matrix:
Enter element at row 1, column 1: 4
Enter element at row 1, column 2: 5
Enter element at row 2, column 1: 2
Enter element at row 2, column 2: 3
matrix 2:
4  5
2  3


Matrix Addition:
9  11
4  6

Matrix Subtraction:
1  1
0  0
```

33. Write a program that calculates the greatest common divisor (GCD) of two numbers using the Euclidean algorithm

In [51]:
```
num1=int(input("Enter the first number:"))
num2=int(input("Enter the second number:"))

def euclidean_gcd(a, b):
    while b:
        a, b = b, a % b
    return a

gcd = euclidean_gcd(num1, num2)

print(f"The greatest common divisor (GCD) of {num1} and {num2} is {gcd}.")
```

```
Enter the first number:25
Enter the second number:555
The greatest common divisor (GCD) of 25 and 555 is 5.
```

34. Build a program that performs matrix multiplication using nested loops and conditional statements

In [59]:
```
def matrix_multiplication(matrix1, matrix2):
    if len(matrix1[0]) != len(matrix2):
        return None
    result = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix2[0])):
            element = 0
            for k in range(len(matrix2)):
                element += matrix1[i][k] * matrix2[k][j]
            row.append(element)
        result.append(row)
    return result


def display_matrix(matrix):
    for row in matrix:
        row_str = ""
        for element in row:
            row_str += str(element) + "   "
        print(row_str)


rows1 = int(input("Enter the number of rows for the first matrix: "))
cols1 = int(input("Enter the number of columns for the first matrix: "))
rows2 = int(input("Enter the number of rows for the second matrix: "))
cols2 = int(input("Enter the number of columns for the second matrix: "))

if cols1 != rows2:
    print("Matrix multiplication is not possible due to incompatible dimens
else:
    print("Enter the elements of the first matrix:")
    matrix1 = []
    for i in range(rows1):
        row = []
        for j in range(cols1):
            element = int(input(f"Enter element at row {i + 1}, column {j +
            row.append(element)
        matrix1.append(row)
    print("Matrix 1:")
    display_matrix(matrix1)
    print("\nEnter the elements of the second matrix:")
    matrix2 = []
    for i in range(rows2):
        row = []
        for j in range(cols2):
            element = int(input(f"Enter element at row {i + 1}, column {j +
            row.append(element)
        matrix2.append(row)
    print("Matrix 2:")
    display_matrix(matrix2)

    result = matrix_multiplication(matrix1, matrix2)

    if result is not None:
        print("\nMatrix Multiplication:")
        display_matrix(result)
    else:
        print("\nMatrix multiplication is not possible due to incompatible
```

```
Enter the number of rows for the first matrix: 2
Enter the number of columns for the first matrix: 3
Enter the number of rows for the second matrix: 3
Enter the number of columns for the second matrix: 2
Enter the elements of the first matrix:
Enter element at row 1, column 1: 5
Enter element at row 1, column 2: 6
Enter element at row 1, column 3: 4
Enter element at row 2, column 1: 2
Enter element at row 2, column 2: 3
Enter element at row 2, column 3: 5
Matrix 1:
5   6   4
2   3   5

Enter the elements of the second matrix:
Enter element at row 1, column 1: 6
Enter element at row 1, column 2: 2
Enter element at row 2, column 1: 3
Enter element at row 2, column 2: 5
Enter element at row 3, column 1: 6
Enter element at row 3, column 2: 5
Matrix 2:
6   2
3   5
6   5

Matrix Multiplication:
72   60
51   44
```

35. Create a program that simulates a basic text-based tic-tac-toe game against the computer

In [63]:
```
import random

def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)


def check_winner(board, player):
    for row in board:
        if all([cell == player for cell in row]):
            return True
    for col in range(3):
        if all([board[row][col] == player for row in range(3)]):
            return True
    if all([board[i][i] == player for i in range(3)]) or all([board[i][2 -
        return True
    return False

# Function to make a computer move
def computer_move(board, computer_char):
    empty_cells = [(i, j) for i in range(3) for j in range(3) if board[i][j
    if empty_cells:
        return random.choice(empty_cells)
    else:
        return None

# Initialize the Tic-Tac-Toe board
board = [[" " for _ in range(3)] for _ in range(3)]

# Print the initial board
print("Welcome to Tic-Tac-Toe!")
print_board(board)

# Main game loop
player_char = "X"
computer_char = "O"
while True:
    # Player's move
    while True:
        try:
            row, col = map(int, input("Enter row and column (e.g., 1 2): ")
            if 1 <= row <= 3 and 1 <= col <= 3 and board[row - 1][col - 1]
                board[row - 1][col - 1] = player_char
                break
            else:
                print("Invalid move. Try again.")
        except (ValueError, IndexError):
            print("Invalid input. Enter row and column as numbers, e.g., '1

    # Print the updated board
    print_board(board)

    # Check if the player has won
    if check_winner(board, player_char):
        print("Congratulations! You win!")
        break

    # Check for a tie
    if all(cell != " " for row in board for cell in row):
        print("It's a tie!")
```

```
            break

        # Computer's move
        computer_row, computer_col = computer_move(board, computer_char)
        if computer_row is not None:
            board[computer_row][computer_col] = computer_char

            # Print the updated board
            print("Computer's Move:")
            print_board(board)

            # Check if the computer has won
            if check_winner(board, computer_char):
                print("Computer wins! Better luck next time.")
                break
        else:
            print("It's a tie!")
            break
```

```
Welcome to Tic-Tac-Toe!
  |   |
---------
  |   |
---------
  |   |
---------
Enter row and column (e.g., 1 2): 2 2
  |   |
---------
  | X |
---------
  |   |
---------
Computer's Move:
  |   | O
---------
  | X |
---------
  |   |
---------
Enter row and column (e.g., 1 2): 1 2
  | X | O
---------
  | X |
---------
  |   |
---------
Computer's Move:
  | X | O
---------
  | X |
---------
O |   |
---------
Enter row and column (e.g., 1 2): 2 2
Invalid move. Try again.
Enter row and column (e.g., 1 2): 1 3
Invalid move. Try again.
Enter row and column (e.g., 1 2): 2 3
  | X | O
---------
  | X | X
---------
O |   |
---------
Computer's Move:
  | X | O
---------
  | X | X
---------
O | O |
---------
Enter row and column (e.g., 1 2): 1 1
X | X | O
---------
  | X | X
---------
O | O |
---------
Computer's Move:
```

```
X | X | O
---------
  | X | X
---------
O | O | O
---------
Computer wins! Better luck next time.
```

36. Write a program that generates Fibonacci numbers up to a specified term using iterative methods

In [64]:
```
n=int(input("Enter the value of n:"))
a=0
b=1
print("First",n,"fibonacci numbers are:")
print(a,b,end=" ")
for i in range(n-2):
    c=a+b
    a=b
    b=c
    print(c,end=" ")
```

```
Enter the value of n:15
First 15 fibonacci numbers are:
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

37. Develop a program that calculates the nth term of the Fibonacci sequence using memoization

In [70]:

```
def fibonacci(n, memo={}):
    if n in memo:
        return memo[n]
    if n <= 1:
        return n
    memo[n] = fibonacci(n - 1, memo) + fibonacci(n - 2, memo)
    return memo[n]


term = int(input("Enter the term (n) to calculate the Fibonacci number for:

if term < 0:
    print("Please enter a non-negative integer.")
else:

    result = fibonacci(term)

    print(f"The Fibonacci number at term {term} is: {result}")
```

```
Enter the term (n) to calculate the Fibonacci number for: 14
The Fibonacci number at term 14 is: 377
```

38. Create a program that generates a calendar for a given month and year using conditional statements

In [76]:
```
import calendar

def generate_calendar(year, month):
    cal = calendar.month(year, month)
    return cal


year = int(input("Enter the year: "))
month = int(input("Enter the month (1-12): "))

if month < 1 or month > 12:
    print("Invalid month. Please enter a number between 1 and 12.")
else:
    print(f"\nCalendar for {month}/{year}:\n")
    calendar_text = generate_calendar(year, month)
    print(calendar_text)
```

```
Enter the year: 2002
Enter the month (1-12): 7

Calendar for 7/2002:

     July 2002
Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

39. Build a program that simulates a basic text-based blackjack game against the computer

In [2]:
```
import random

card_categories = ['Hearts', 'Diamonds', 'Clubs', 'Spades']
cards_list = ['Ace', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'Jack',
deck = [(card, category) for category in card_categories for card in cards_

def card_value(card):
    if card[0] in ['Jack', 'Queen', 'King']:
        return 10
    elif card[0] == 'Ace':
        return 11
    else:
        return int(card[0])

random.shuffle(deck)
player_card = [deck.pop(), deck.pop()]
dealer_card = [deck.pop(), deck.pop()]

while True:
    player_score = sum(card_value(card) for card in player_card)
    dealer_score = sum(card_value(card) for card in dealer_card)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("\n")
    choice = input('What do you want? ["play" to request another card, "sto
    if choice == "play":
        new_card = deck.pop()
        player_card.append(new_card)
    elif choice == "stop":
        break
    else:
        print("Invalid choice. Please try again.")
        continue

    if player_score > 21:
        print("Cards Dealer Has:", dealer_card)
        print("Score Of The Dealer:", dealer_score)
        print("Cards Player Has:", player_card)
        print("Score Of The Player:", player_score)
        print("Dealer wins (Player Loss Because Player Score is exceeding 2
        break

while dealer_score < 17:
    new_card = deck.pop()
    dealer_card.append(new_card)
    dealer_score += card_value(new_card)

print("Cards Dealer Has:", dealer_card)
print("Score Of The Dealer:", dealer_score)
print("\n")

if dealer_score > 21:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("Player wins (Dealer Loss Because Dealer Score is exceeding 21)")
elif player_score > dealer_score:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
```

```
    print("Score Of The Player:", player_score)
    print("Player wins (Player Has High Score than Dealer)")
elif dealer_score > player_score:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("Dealer wins (Dealer Has High Score than Player)")
else:
    print("Cards Dealer Has:", dealer_card)
    print("Score Of The Dealer:", dealer_score)
    print("Cards Player Has:", player_card)
    print("Score Of The Player:", player_score)
    print("It's a tie.")
```

```
Cards Player Has: [('6', 'Hearts'), ('9', 'Diamonds')]
Score Of The Player: 15


What do you want? ["play" to request another card, "stop" to stop]: play
Cards Player Has: [('6', 'Hearts'), ('9', 'Diamonds'), ('7', 'Spades')]
Score Of The Player: 22


What do you want? ["play" to request another card, "stop" to stop]: play
Cards Dealer Has: [('8', 'Spades'), ('Jack', 'Spades')]
Score Of The Dealer: 18
Cards Player Has: [('6', 'Hearts'), ('9', 'Diamonds'), ('7', 'Spades'),
('5', 'Diamonds')]
Score Of The Player: 22
Dealer wins (Player Loss Because Player Score is exceeding 21)
Cards Dealer Has: [('8', 'Spades'), ('Jack', 'Spades')]
Score Of The Dealer: 18


Cards Dealer Has: [('8', 'Spades'), ('Jack', 'Spades')]
Score Of The Dealer: 18
Cards Player Has: [('6', 'Hearts'), ('9', 'Diamonds'), ('7', 'Spades'),
('5', 'Diamonds')]
Score Of The Player: 22
Player wins (Player Has High Score than Dealer)
```

40. Write a program that generates the prime factors of a given number using trial division

In [8]:
```
def prime_factors(n):
    factors = []
    divisor = 2
    while divisor <= n:
        if n % divisor == 0:
            factors.append(divisor)
            n = n // divisor  # Update n by dividing it by the divisor
        else:
            divisor += 1

    return factors

num = int(input("Enter a number to find its prime factors: "))

if num <= 1:
    print("Please enter a positive integer greater than 1.")
else:
    factors = prime_factors(num)

    if len(factors) == 0:
        print(f"{num} is a prime number, and it has no prime factors other
    else:
        print("The prime factors of", num, "are:", end=" ")
        for i in range(len(factors)):
            if i < len(factors) - 1:
                print(factors[i], end=", ")
            else:
                print(factors[i])
```

```
Enter a number to find its prime factors: 144
The prime factors of 144 are: 2, 2, 2, 2, 3, 3
```

In [ ]: