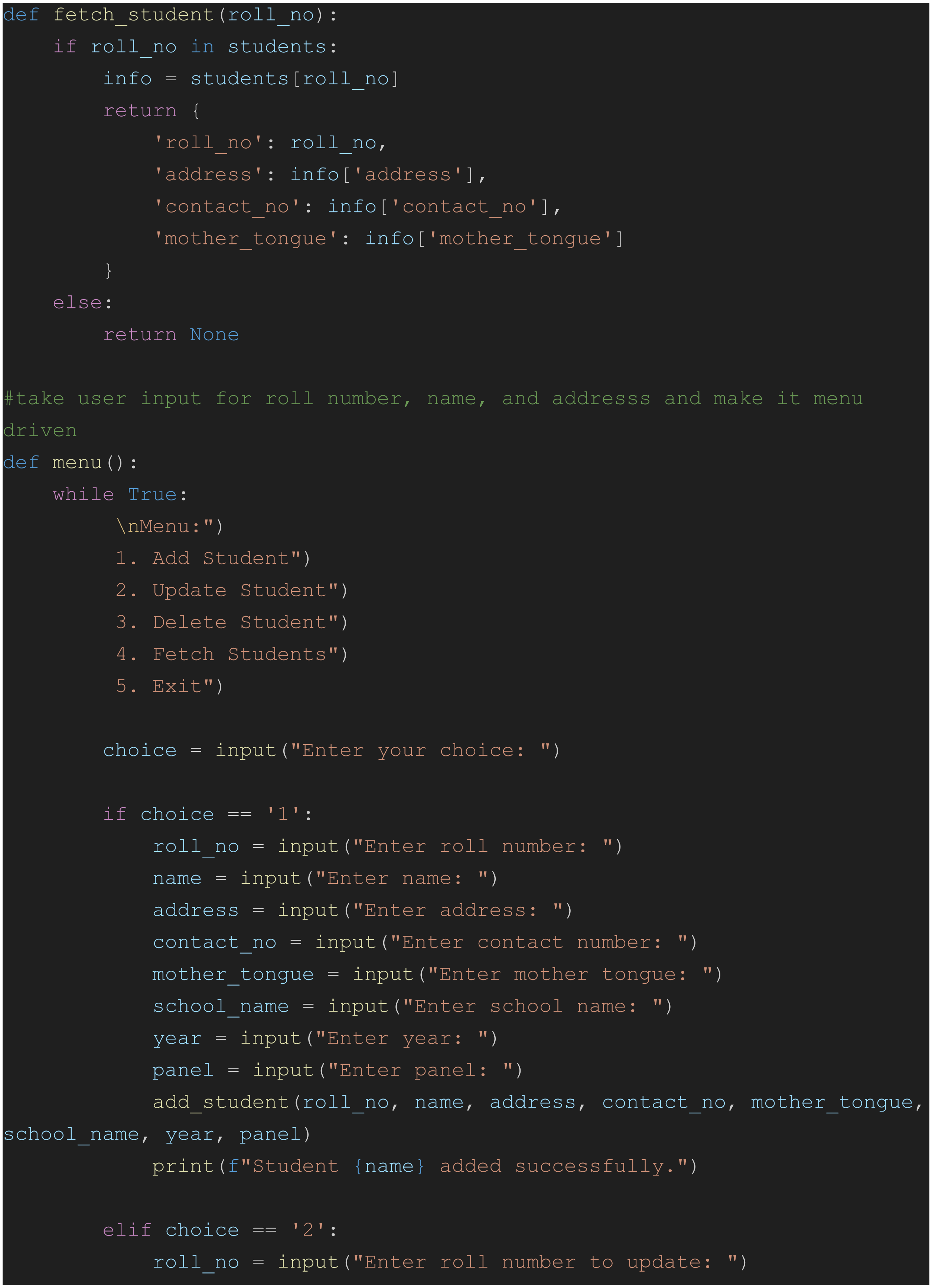
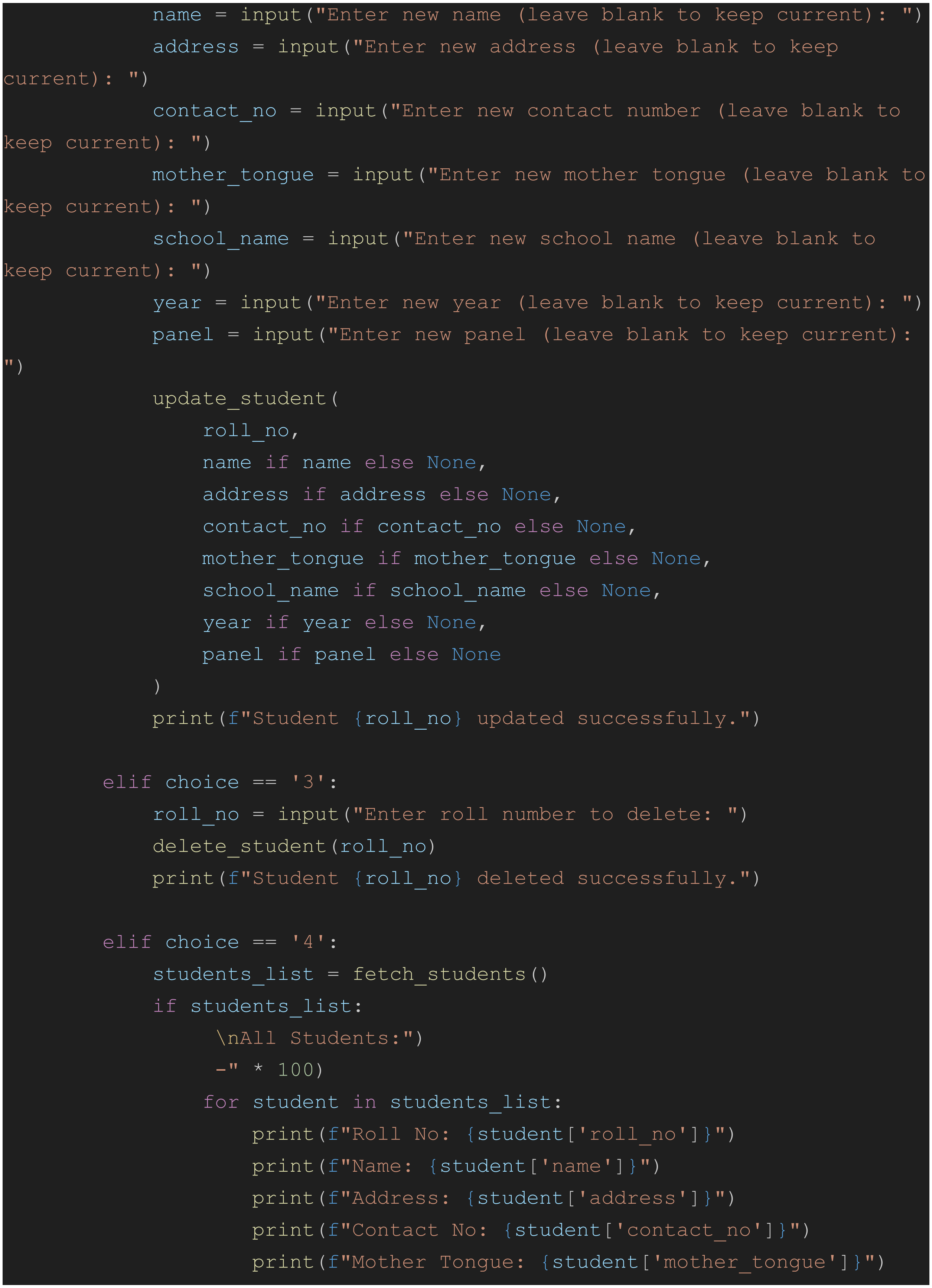
Student db:

```
students = {}
def add_student(roll_no, name, address, contact_no, mother_tongue,
school_name, year, panel):
    students[roll_no] = {
        'name': name,
        'address': address,
        'contact_no': contact_no,
        'mother_tongue': mother_tongue,
        'school_name': school_name,
        'year': year,
        'panel': panel
    }

def update_student(roll_no, name=None, address=None, contact_no=None,
mother_tongue=None, school_name=None, year=None, panel=None):
    if roll_no in students:
        if name:
            students[roll_no]['name'] = name
        if address:
            students[roll_no]['address'] = address
        if contact_no:
            students[roll_no]['contact_no'] = contact_no
        if mother_tongue:
            students[roll_no]['mother_tongue'] = mother_tongue
        if school_name:
            students[roll_no]['school_name'] = school_name
        if year:
            students[roll_no]['year'] = year
        if panel:
            students[roll_no]['panel'] = panel

def delete_student(roll_no):
    if roll_no in students:
        del students[roll_no]

def fetch_students():
    return [{ 'roll_no': roll_no, **info } for roll_no, info in
students.items()]
```



```
def fetch_student(roll_no):
    if roll_no in students:
        info = students[roll_no]
        return {
            'roll_no': roll_no,
            'address': info['address'],
            'contact_no': info['contact_no'],
            'mother_tongue': info['mother_tongue']
        }
    else:
        return None

#take user input for roll number, name, and addresss and make it menu
driven
def menu():
    while True:
         \nMenu:")
         1. Add Student")
         2. Update Student")
         3. Delete Student")
         4. Fetch Students")
         5. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            roll_no = input("Enter roll number: ")
            name = input("Enter name: ")
            address = input("Enter address: ")
            contact_no = input("Enter contact number: ")
            mother_tongue = input("Enter mother tongue: ")
            school_name = input("Enter school name: ")
            year = input("Enter year: ")
            panel = input("Enter panel: ")
            add_student(roll_no, name, address, contact_no, mother_tongue,
school_name, year, panel)
            print(f"Student {name} added successfully.")

        elif choice == '2':
            roll_no = input("Enter roll number to update: ")
```



```
            name = input("Enter new name (leave blank to keep current): ")
            address = input("Enter new address (leave blank to keep
current): ")
            contact_no = input("Enter new contact number (leave blank to
keep current): ")
            mother_tongue = input("Enter new mother tongue (leave blank to
keep current): ")
            school_name = input("Enter new school name (leave blank to
keep current): ")
            year = input("Enter new year (leave blank to keep current): ")
            panel = input("Enter new panel (leave blank to keep current):
")
            update_student(
                roll_no,
                name if name else None,
                address if address else None,
                contact_no if contact_no else None,
                mother_tongue if mother_tongue else None,
                school_name if school_name else None,
                year if year else None,
                panel if panel else None
            )
            print(f"Student {roll_no} updated successfully.")

        elif choice == '3':
            roll_no = input("Enter roll number to delete: ")
            delete_student(roll_no)
            print(f"Student {roll_no} deleted successfully.")

        elif choice == '4':
            students_list = fetch_students()
            if students_list:
                 \nAll Students:")
                 -" * 100)
                for student in students_list:
                    print(f"Roll No: {student['roll_no']}")
                    print(f"Name: {student['name']}")
                    print(f"Address: {student['address']}")
                    print(f"Contact No: {student['contact_no']}")
                    print(f"Mother Tongue: {student['mother_tongue']}")
                    print(f"School Name: {student['school_name']}")
                    print(f"Year: {student['year']}")
                    print(f"Panel: {student['panel']}")
                     -" * 100)
            else:
                 No students found.")

        elif choice == '5':
             Exiting the program.")
            break

        else:
             Invalid choice. Please try again.")
```

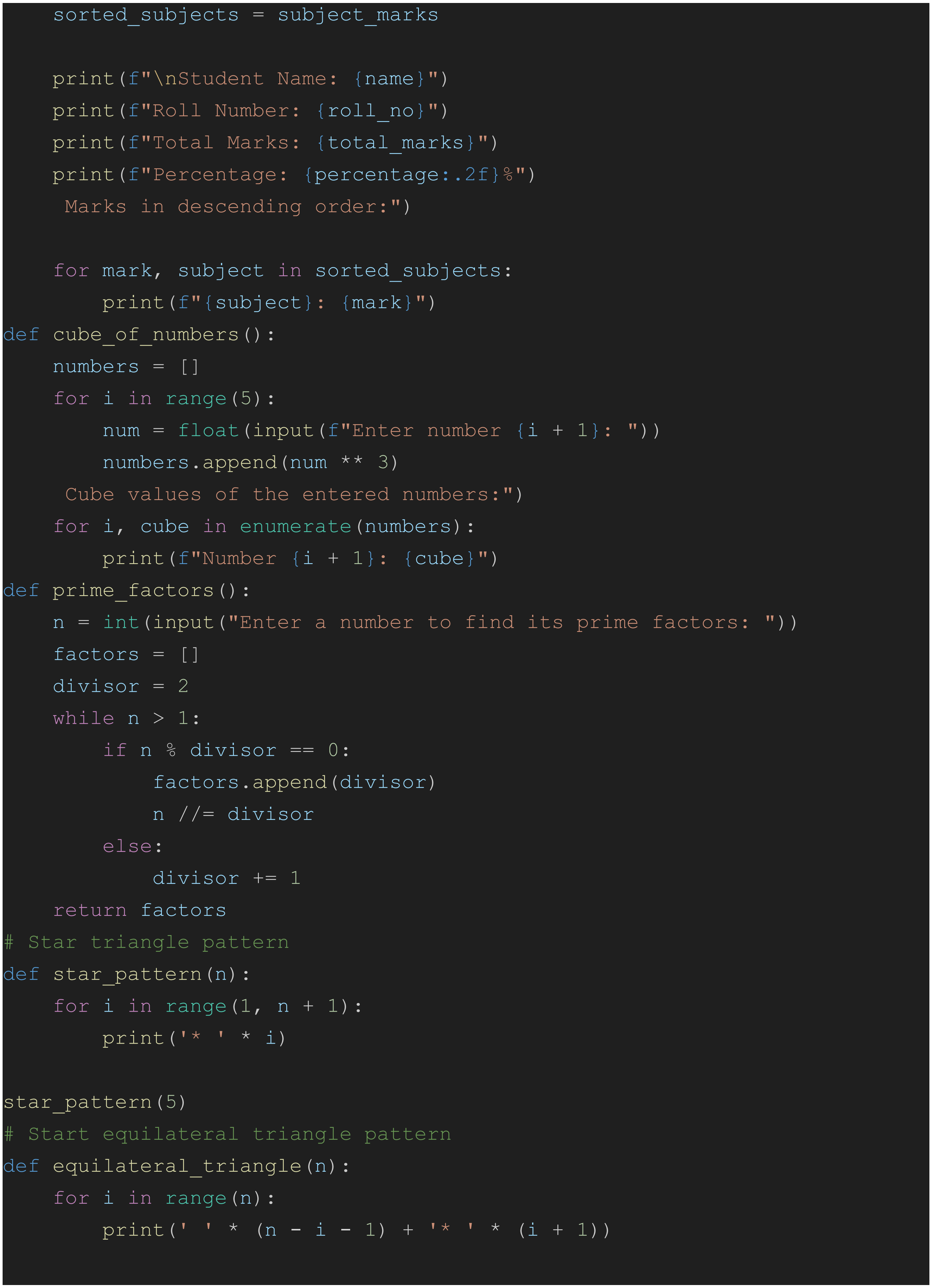
Output:

```
Menu:
1. Add Student
2. Update Student
3. Delete Student
4. Fetch Students
5. Exit
Student ak added successfully.

Menu:
1. Add Student
2. Update Student
3. Delete Student
4. Fetch Students
5. Exit

All Students:
----------------------------------------------------------------------------------------------------
Roll No: 1
Name: ak
Address: pune
Contact No: 1234567890
Mother Tongue: english
School Name: MIT
Year: 2025
Panel: b
----------------------------------------------------------------------------------------------------

Menu:
```

```
1. Add Student
2. Update Student
3. Delete Student
4. Fetch Students
5. Exit
Exiting the program.
```

Switch case for remaining exercises:

```
# Consolidated switch case statement for all functions

def switch_case():
    while True:
         \Switch Case Menu:")
         1. Largest of Three Numbers")
         2. Even or Odd")
         3. Student Marks")
         4. Cube of Numbers")
         5. Prime Factors of a Number")
         6. Star Pattern")
         7. Equilateral Triangle Pattern")
         8. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            a = float(input("Enter first number: "))
            b = float(input("Enter second number: "))
            c = float(input("Enter third number: "))
            print(f"Largest number is: {largest_of_three(a, b, c)}")

        elif choice == '2':
            print(f"The number is {even_or_odd()}.")

        elif choice == '3':
            student_marks()

        elif choice == '4':
            cube_of_numbers()

        elif choice == '5':
            factors = prime_factors()
            print(f"Prime factors are: {factors}")

        elif choice == '6':
            n = int(input("Enter the number of rows for the star pattern:
"))
            star_pattern(n)
        elif choice == '7':
            n = int(input("Enter the number of rows for the equilateral
triangle pattern: "))
            equilateral_triangle(n)

        elif choice == '8':
             Exiting the program.")
            break

        else:
             Invalid choice. Please try again.")

def largest_of_three(a, b, c):
    return max(a, b, c)
def even_or_odd():
    num = int(input("Enter a number: "))
    return "Even" if num % 2 == 0 else "Odd"
def student_marks():
    roll_no = input("Enter roll number: ")
    name = input("Enter student name: ")
    marks = []

    for i in range(1, 4):
        mark = float(input(f"Enter marks for subject {i}: "))
        marks.append(mark)

    total_marks = sum(marks)
    percentage = (total_marks / 300) * 100
    # Create a list of (mark, subject) pairs manually
    subjects = [f"Subject {i}" for i in range(1, 4)]
    subject_marks = []
    for i in range(3):
        subject_marks.append((marks[i], subjects[i]))
    # Sort the list in descending order by marks
    for i in range(len(subject_marks)):
        for j in range(i + 1, len(subject_marks)):
            if subject_marks[j][0] > subject_marks[i][0]:
                subject_marks[i], subject_marks[j] = subject_marks[j],
subject_marks[i]
```



```
    sorted_subjects = subject_marks

    print(f"\nStudent Name: {name}")
    print(f"Roll Number: {roll_no}")
    print(f"Total Marks: {total_marks}")
    print(f"Percentage: {percentage:.2f}%")
     Marks in descending order:")

    for mark, subject in sorted_subjects:
        print(f"{subject}: {mark}")
def cube_of_numbers():
    numbers = []
    for i in range(5):
        num = float(input(f"Enter number {i + 1}: "))
        numbers.append(num ** 3)
     Cube values of the entered numbers:")
    for i, cube in enumerate(numbers):
        print(f"Number {i + 1}: {cube}")
def prime_factors():
    n = int(input("Enter a number to find its prime factors: "))
    factors = []
    divisor = 2
    while n > 1:
        if n % divisor == 0:
            factors.append(divisor)
            n //= divisor
        else:
            divisor += 1
    return factors
# Star triangle pattern
def star_pattern(n):
    for i in range(1, n + 1):
        print('* ' * i)

star_pattern(5)
# Start equilateral triangle pattern
def equilateral_triangle(n):
    for i in range(n):
        print(' ' * (n - i - 1) + '* ' * (i + 1))
```

```
equilateral_triangle(5)
```

Outputs

1. Largest of Three Numbers:

```
Enter your choice: 1
Enter first number: 15
Enter second number: 8
Enter third number: 12
Largest number is: 15.0
```

2. Even or Odd:

```
Enter your choice: 2
Enter a number: 7
The number is Odd.
```

3. Student Marks

```
Enter your choice: 3
Enter roll number: 123
Enter student name: John Smith
Enter marks for subject 1: 85
Enter marks for subject 2: 92
Enter marks for subject 3: 78

Student Name: John Smith
Roll Number: 123
Total Marks: 255.0
Percentage: 85.00%
Marks in descending order:
Subject 2: 92.0
Subject 1: 85.0
Subject 3: 78.0
```

4. Cube of Numbers

```
Enter your choice: 4
Enter number 1: 2
Enter number 2: 3
Enter number 3: 4
Enter number 4: 5
Enter number 5: 6
Cube values of the entered numbers:
Number 1: 8.0
Number 2: 27.0
Number 3: 64.0
Number 4: 125.0
Number 5: 216.0

5. Prime Factors of a Number
Enter your choice: 5
Enter a number to find its prime factors: 60
Prime factors are: [2, 2, 3, 5]

6. Star Pattern
Enter your choice: 6
Enter the number of rows for the star pattern: 4
*
* *
* * *
* * * *

7. Equilateral Triangle Pattern
Enter your choice: 7
Enter the number of rows for the equilateral triangle pattern: 4
   *
  * *
 * * *
* * * *

8. Exit
Enter your choice: 8
Exiting the program.
```