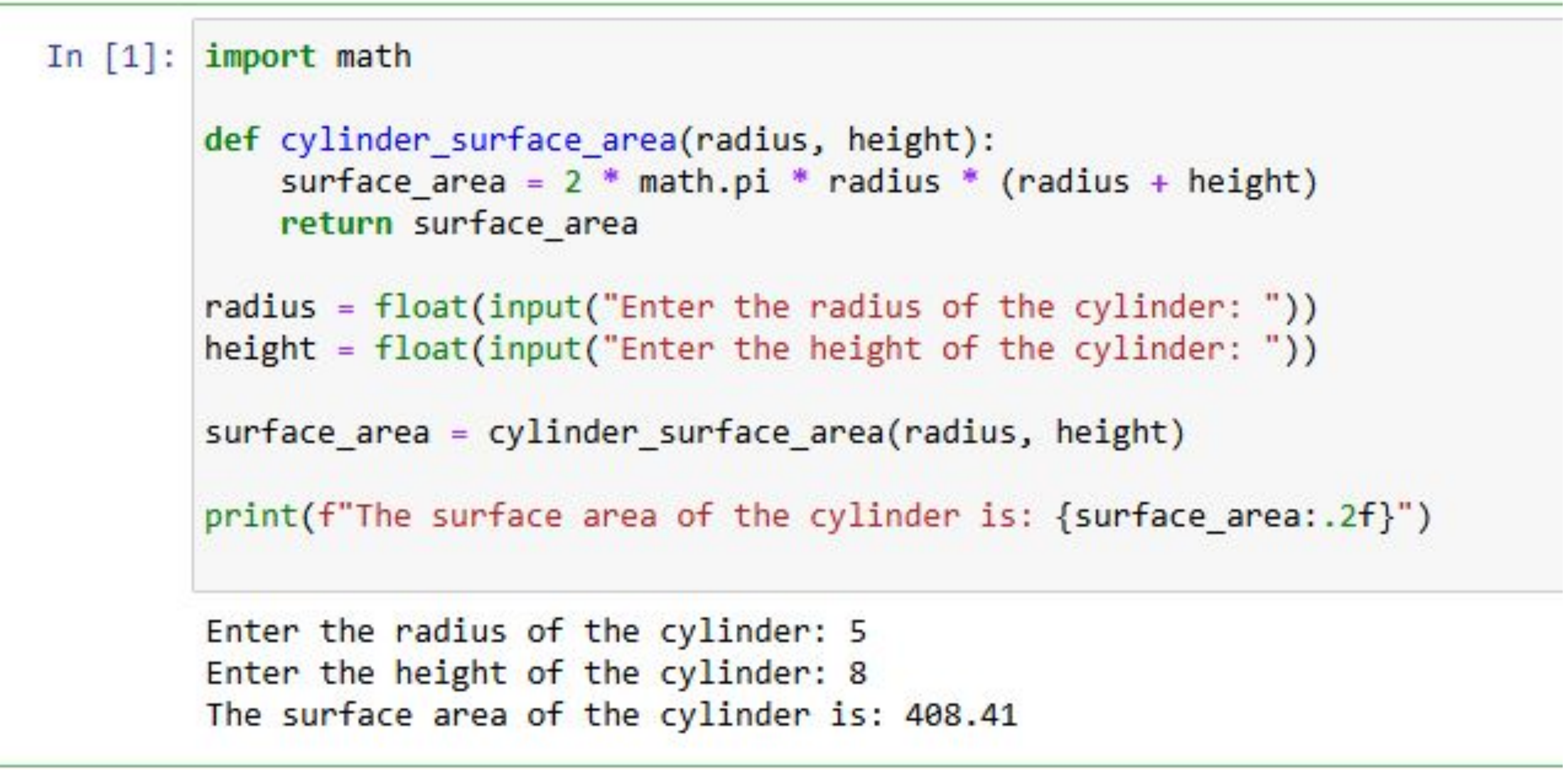
ANSWERS FOR PST 31130 COMPUTER LABORATORY 3-I

01)

a. Write a program in Python that facilitates users to enter radius (r) and height (h) of a cylinder. Calculates surface area of cylinder using following formula.

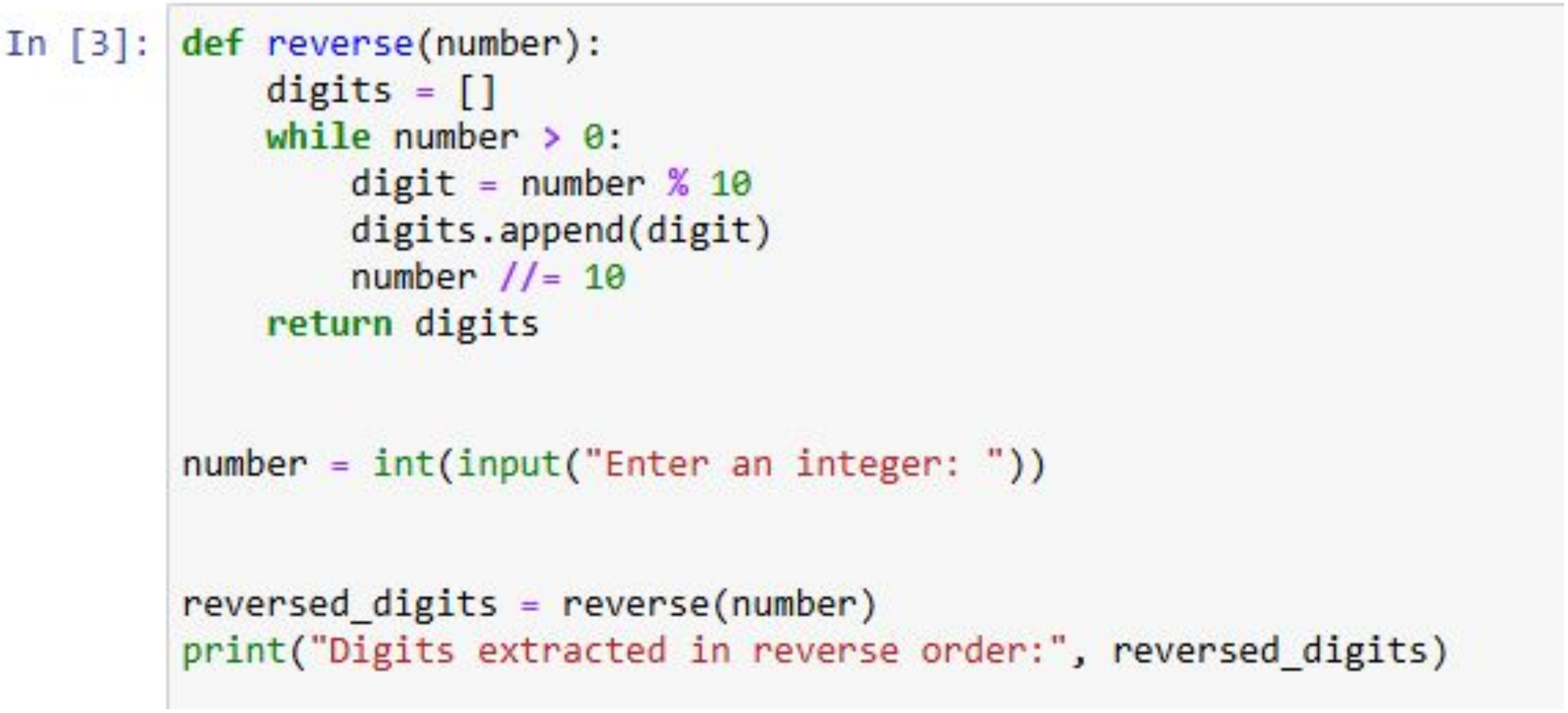
Surface area = 2 * Pi * r * (r + h)


In [1]:
```python
import math

def cylinder_surface_area(radius, height):
    surface_area = 2 * math.pi * radius * (radius + height)
    return surface_area

radius = float(input("Enter the radius of the cylinder: "))
height = float(input("Enter the height of the cylinder: "))

surface_area = cylinder_surface_area(radius, height)

print(f"The surface area of the cylinder is: {surface_area:.2f}")
```

```
Enter the radius of the cylinder: 5
Enter the height of the cylinder: 8
The surface area of the cylinder is: 408.41
```

b. Write a Python program that takes an integer as input and extracts each digit from the integer in reverse order. Then, print the extracted digits.

Example:
Input: 12345
Digits extracted in reverse order: [5, 4, 3, 2, 1]



```
In [3]: def reverse(number):
            digits = []
            while number > 0:
                digit = number % 10
                digits.append(digit)
                number //= 10
            return digits


        number = int(input("Enter an integer: "))


        reversed_digits = reverse(number)
        print("Digits extracted in reverse order:", reversed_digits)
```

02).

Develop a Python program that computes the maturity amount of a bank deposit using user-provided inputs. The program offers a user-friendly menu interface allowing users to select between calculating maturity amounts for both Term Deposit and Recurring Deposit. The program iteratively prompts users for input until they option to exit.

The user is given the following options:

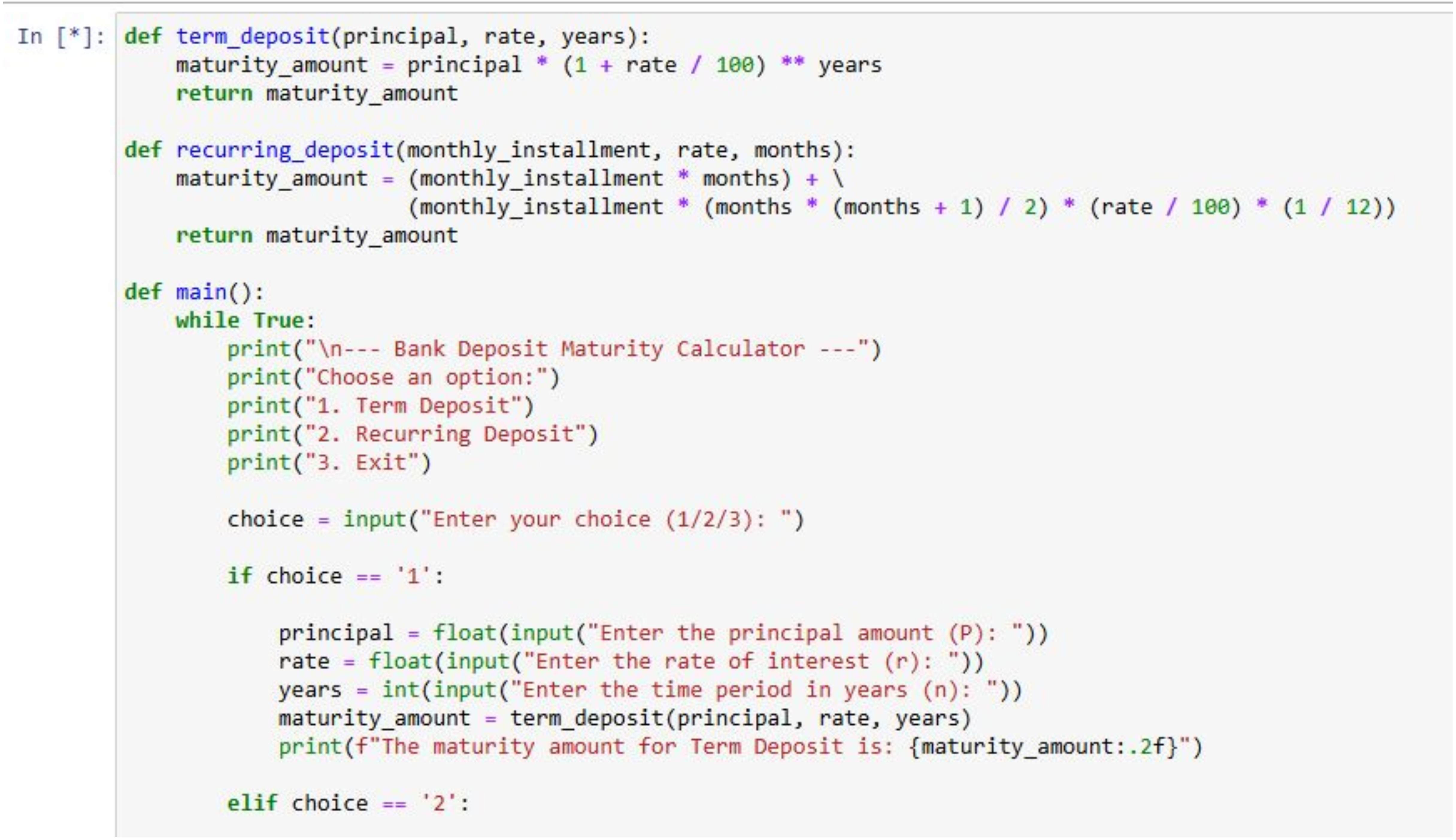
(i) Term Deposit

(ii) Recurring Deposit

For **option (i)** accept principal(P), rare of interest(r) and time period in years(n). Calculate and output the maturity amount(A) receivable using the formula,

For **option (ii)** accept Monthly Installment (P), rate of interest(r) and time period in months (n). Calculate and output the maturity amount(A) receivable using the formula,

$$\boxed{A = P \times n + P \times \frac{n(n+1)}{2} \times \frac{r}{100} \times \frac{1}{12}}$$



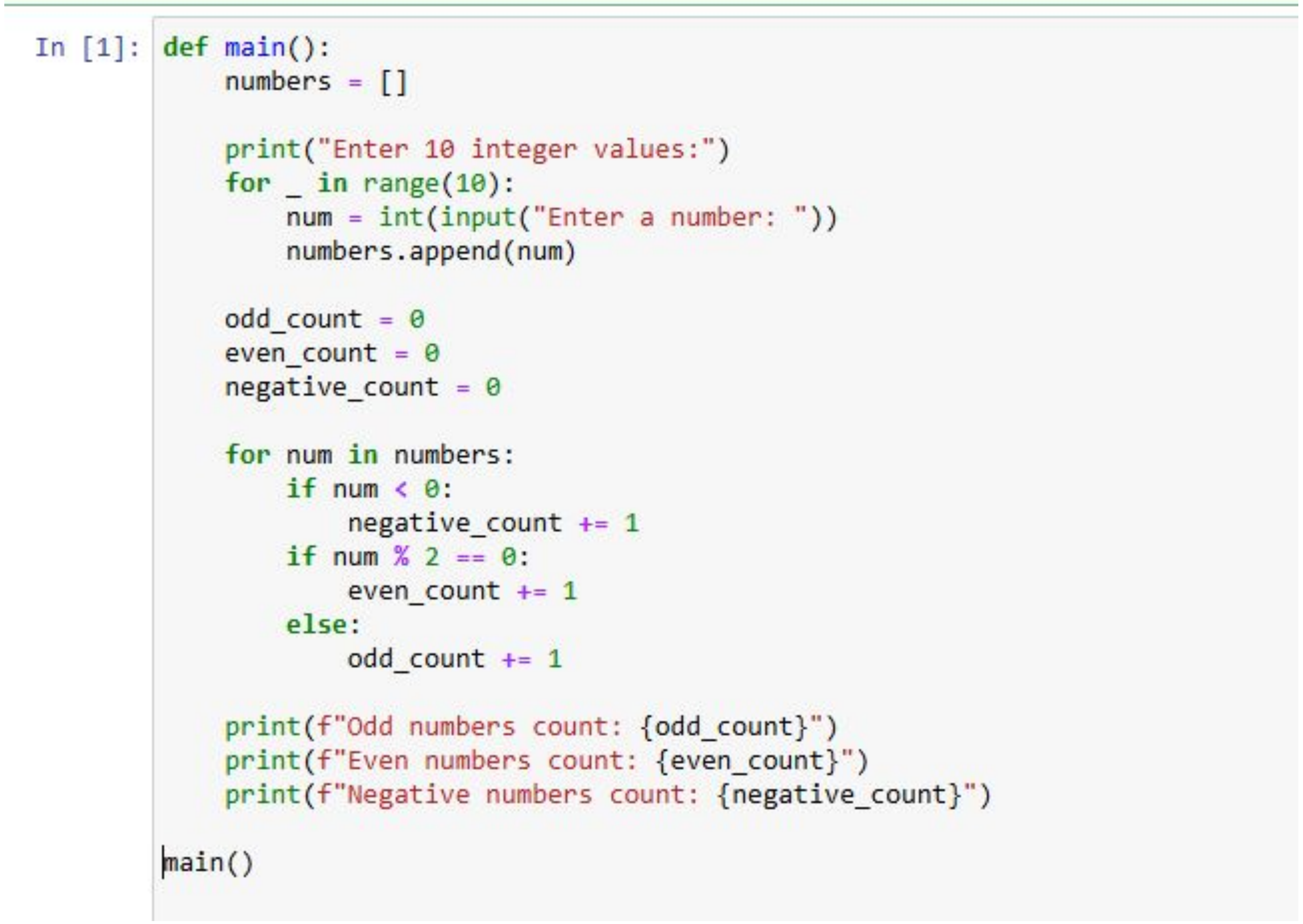
In [*]:

```python
def term_deposit(principal, rate, years):
    maturity_amount = principal * (1 + rate / 100) ** years
    return maturity_amount

def recurring_deposit(monthly_installment, rate, months):
    maturity_amount = (monthly_installment * months) + \
                      (monthly_installment * (months * (months + 1) / 2) * (rate / 100) * (1 / 12))
    return maturity_amount

def main():
    while True:
        print("\n--- Bank Deposit Maturity Calculator ---")
        print("Choose an option:")
        print("1. Term Deposit")
        print("2. Recurring Deposit")
        print("3. Exit")

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

        if choice == '1':

            principal = float(input("Enter the principal amount (P): "))
            rate = float(input("Enter the rate of interest (r): "))
            years = int(input("Enter the time period in years (n): "))
            maturity_amount = term_deposit(principal, rate, years)
            print(f"The maturity amount for Term Deposit is: {maturity_amount:.2f}")

        elif choice == '2':
```

```python
        if choice == '1':

            principal = float(input("Enter the principal amount (P): "))
            rate = float(input("Enter the rate of interest (r): "))
            years = int(input("Enter the time period in years (n): "))
            maturity_amount = term_deposit(principal, rate, years)
            print(f"The maturity amount for Term Deposit is: {maturity_amount:.2f}")

        elif choice == '2':

            monthly_installment = float(input("Enter the monthly installment (P): "))
            rate = float(input("Enter the rate of interest (r): "))
            months = int(input("Enter the time period in months (n): "))
            maturity_amount = recurring_deposit(monthly_installment, rate, months)
            print(f"The maturity amount for Recurring Deposit is: {maturity_amount:.2f}")

        elif choice == '3':
            print("Exiting the program. Goodbye!")
            break
        else:
            print("Invalid choice. Please try again.")


main()
```

03).

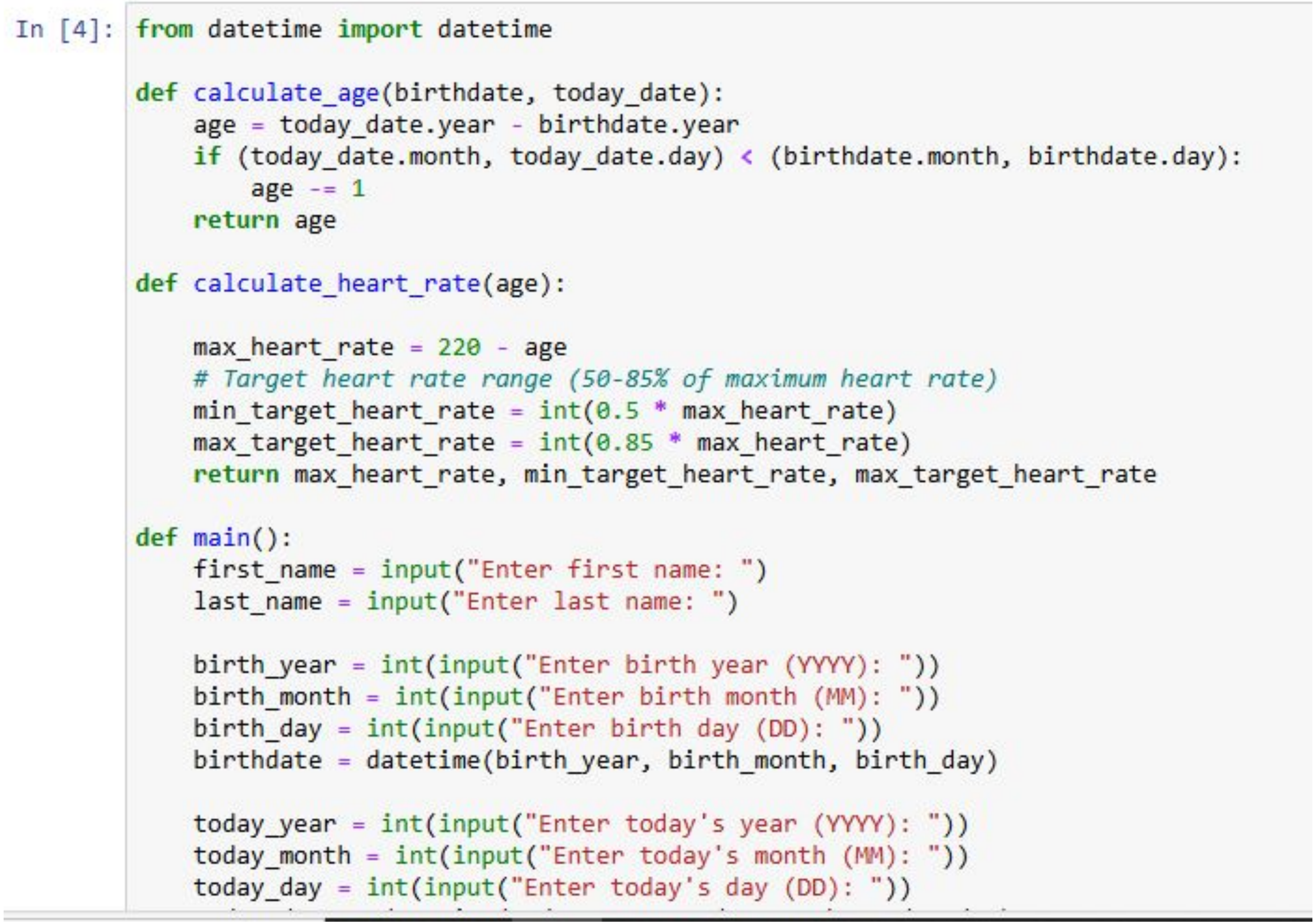
a. Write a Python program that prompts the user to input 10 integer values and stores them in an **array.** The program should then determine and display the counts of odd, even, and negative numbers among these inputs.

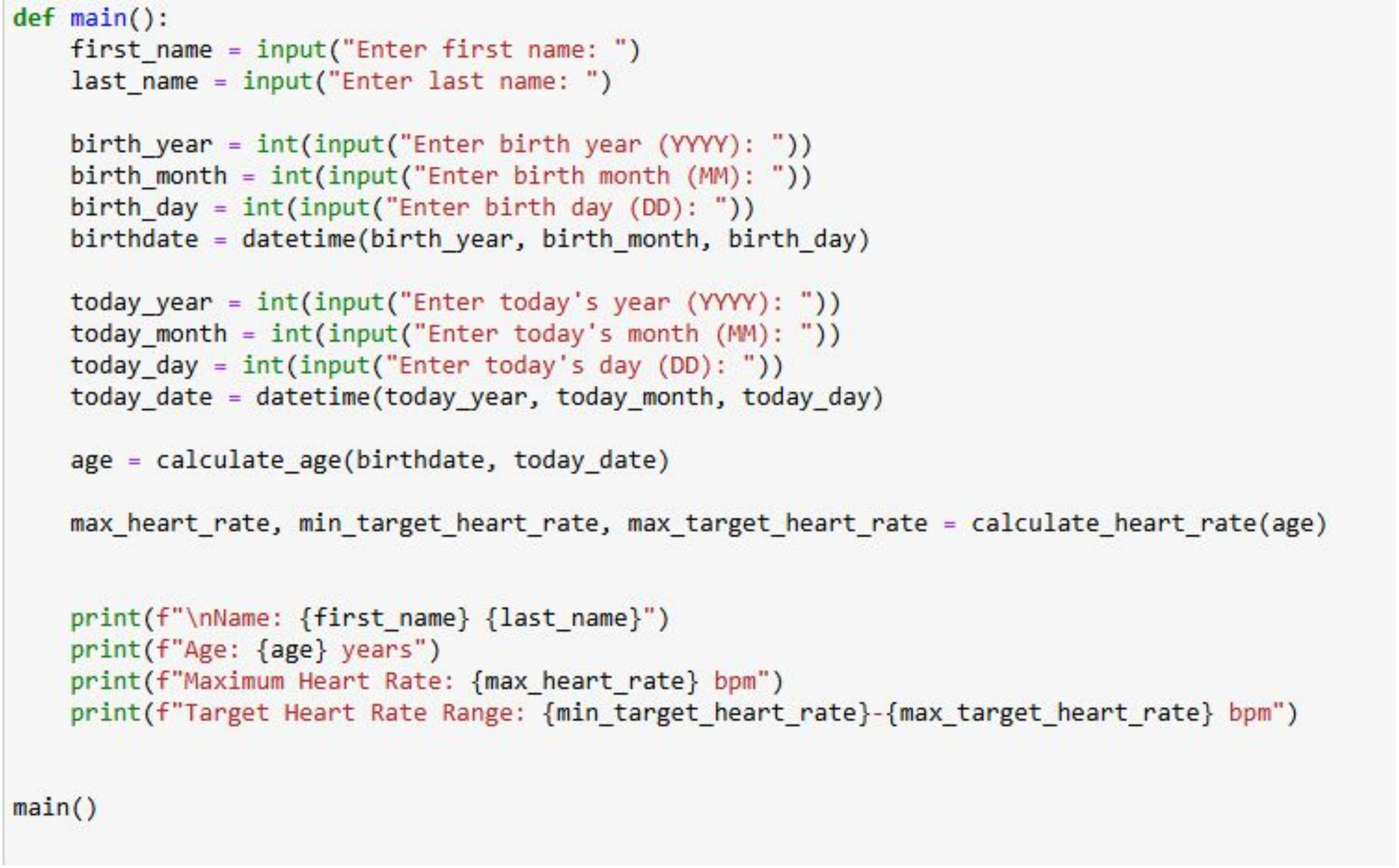

In [1]:
```
def main():
    numbers = []

    print("Enter 10 integer values:")
    for _ in range(10):
        num = int(input("Enter a number: "))
        numbers.append(num)

    odd_count = 0
    even_count = 0
    negative_count = 0

    for num in numbers:
        if num < 0:
            negative_count += 1
        if num % 2 == 0:
            even_count += 1
        else:
            odd_count += 1

    print(f"Odd numbers count: {odd_count}")
    print(f"Even numbers count: {even_count}")
    print(f"Negative numbers count: {negative_count}")

main()
```

b) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA), the formula for calculating your maximum heart rate in beats per minute is **220 minus your age in years**.
Your target heart rate is a range that's 50– 85% of your maximum heart rate.

Age	Target Heart Rate 50-85% beats per minute (bpm)
20 years	100-170 bpm
30 years	95-162 bpm
35 years	93-157 bpm
40 years	90-153 bpm
45 years	88-149 bpm
50 years	85-145 bpm
55 years	83-140 bpm
60 years	80-136 bpm
65 years	78-132 bpm
70 years	75-128 bpm

- Write a program in Python to prompt for and take the input of a person's name (first and last), date of birth (year, month, and day), and the today date (year, month, and day). Calculates and prints the person's age in (years), maximum heart rate and target heart rate range according to the above table.


In [4]:
```
from datetime import datetime

def calculate_age(birthdate, today_date):
    age = today_date.year - birthdate.year
    if (today_date.month, today_date.day) < (birthdate.month, birthdate.day):
        age -= 1
    return age

def calculate_heart_rate(age):

    max_heart_rate = 220 - age
    # Target heart rate range (50-85% of maximum heart rate)
    min_target_heart_rate = int(0.5 * max_heart_rate)
    max_target_heart_rate = int(0.85 * max_heart_rate)
    return max_heart_rate, min_target_heart_rate, max_target_heart_rate

def main():
    first_name = input("Enter first name: ")
    last_name = input("Enter last name: ")

    birth_year = int(input("Enter birth year (YYYY): "))
    birth_month = int(input("Enter birth month (MM): "))
    birth_day = int(input("Enter birth day (DD): "))
    birthdate = datetime(birth_year, birth_month, birth_day)

    today_year = int(input("Enter today's year (YYYY): "))
    today_month = int(input("Enter today's month (MM): "))
    today_day = int(input("Enter today's day (DD): "))
```



```python
def main():
    first_name = input("Enter first name: ")
    last_name = input("Enter last name: ")

    birth_year = int(input("Enter birth year (YYYY): "))
    birth_month = int(input("Enter birth month (MM): "))
    birth_day = int(input("Enter birth day (DD): "))
    birthdate = datetime(birth_year, birth_month, birth_day)

    today_year = int(input("Enter today's year (YYYY): "))
    today_month = int(input("Enter today's month (MM): "))
    today_day = int(input("Enter today's day (DD): "))
    today_date = datetime(today_year, today_month, today_day)

    age = calculate_age(birthdate, today_date)

    max_heart_rate, min_target_heart_rate, max_target_heart_rate = calculate_heart_rate(age)


    print(f"\nName: {first_name} {last_name}")
    print(f"Age: {age} years")
    print(f"Maximum Heart Rate: {max_heart_rate} bpm")
    print(f"Target Heart Rate Range: {min_target_heart_rate}-{max_target_heart_rate} bpm")


main()
```

04)

You have a dataset containing information about students, including their names, ages, grades, and favorite subjects. Here are the lists representing the data:

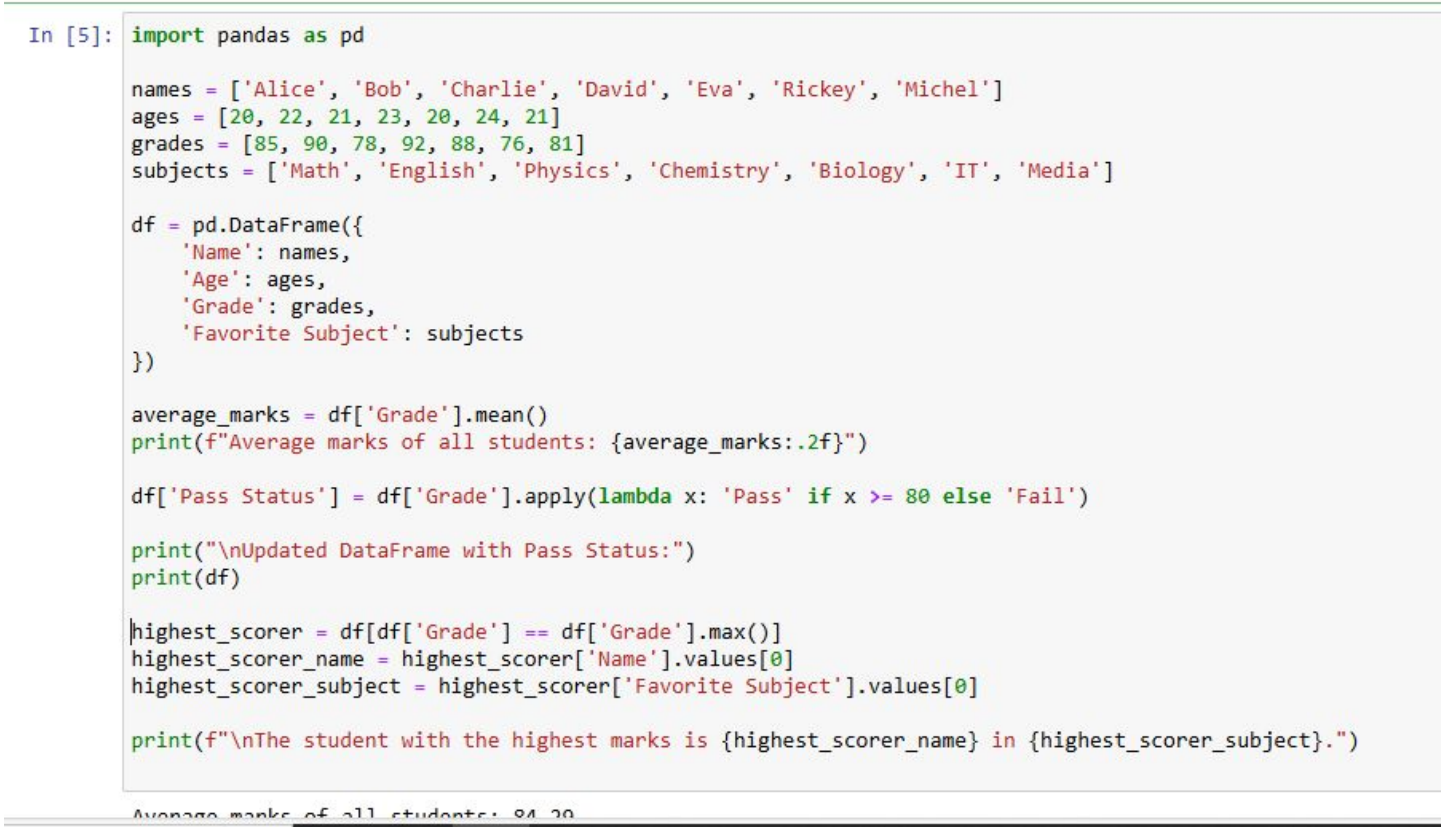
[‘Alice’ , ‘Bob ‘ , ‘Charlie’ , ‘David’ , ‘Eva’, ‘Rickey’, ‘Michel’]

[20,22,21,23,20,24,21]

[85,90,78,92,88,76,81]

[‘Math’, ’English’, ’Physics’ , ‘Chemistry’ ,’Biology’, ‘IT’, ‘Media’]

a) Using the provided lists, create a pandas **DataFrame** named **df** that combines the data into columns with appropriate names.
b) Calculate and display the average marks of all the students in the DataFrame.
c) Create a new column named Pass Status in the DataFrame. This column should indicate whether a student passed or failed based on a passing grade of **80**.
d) Display the updated DataFrame with the new column.
e) Find the student who scored the highest marks and display their name and favorite subject.



```
In [5]: import pandas as pd

        names = ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Rickey', 'Michel']
        ages = [20, 22, 21, 23, 20, 24, 21]
        grades = [85, 90, 78, 92, 88, 76, 81]
        subjects = ['Math', 'English', 'Physics', 'Chemistry', 'Biology', 'IT', 'Media']

        df = pd.DataFrame({
            'Name': names,
            'Age': ages,
            'Grade': grades,
            'Favorite Subject': subjects
        })

        average_marks = df['Grade'].mean()
        print(f"Average marks of all students: {average_marks:.2f}")

        df['Pass Status'] = df['Grade'].apply(lambda x: 'Pass' if x >= 80 else 'Fail')

        print("\nUpdated DataFrame with Pass Status:")
        print(df)

        highest_scorer = df[df['Grade'] == df['Grade'].max()]
        highest_scorer_name = highest_scorer['Name'].values[0]
        highest_scorer_subject = highest_scorer['Favorite Subject'].values[0]

        print(f"\nThe student with the highest marks is {highest_scorer_name} in {highest_scorer_subject}.")
```

```
        Average marks of all students: 84.29
```