Challenge 1.1 ⋮ Exit

```python
num=int(input("please Enter the Number you wish:"))
if (num%4==0):
  if(num%100==0):
    if(num%400==0):
      print("%d is a leap year"%num)
    else:
      print("%d is not"%num)
  else:
    print("%d is a leap year"%num)
else:
   print("%d is not"%num)
```

Challenge 1.1 ⋮ Exit

Run 5s on 18:55:46, 10/24 ✓

```
please Enter the Number you wish:3
3 is not
```

```python
class BankAccount:
    def __init__(self, account_number, account_holder_name, initial_balance=0.0):
        self.__account_number = account_number
        self.__account_holder_name = account_holder_name
        self.__account_balance = initial_balance

    def deposit(self, amount):
        if amount > 0:
            self.__account_balance += amount
            print('Deposited {}. New balance: {}'.format(amount, self.__account_balance))
        else:
            print('Invalid deposit amount. Please deposit a positive amount.')

    def withdraw(self, amount):
        if amount > 0 and amount <= self.__account_balance:
            self.__account_balance -= amount
            print('Withdraw {}. New
```

Ln 1, Col 1 • Spaces: 2 History

Challenge 2.1 ⋮ Exit

```python
class BankAccount:
    def __init__(self, account_number, account_holder_name, initial_balance=0.0):
        self._account_number = account_number
        self.__account_holder_name = account_holder_name
        self.___account_balance = initial_balance

    def deposit(self, amount):
        if amount > 0:
            self.___account_balance += amount
            print("Deposited {}. New balance: {}".format(amount, self.___account_balance))
        else:
            print("Invalid deposit amount. Please deposit a positive amount.")

    def withdraw(self, amount):
        if amount > 0 and amount <= self.___account_balance:
            self.___account_balance -= amount  # Corrected this line
            print("Withdrew {}. New
```

Ln 1, Col 1 • Spaces: 2 History

Challenge 3.1 ⋮ Exit

```python
def linearSearchProduct(productlist, targetProduct):
  indices = []

  for index, product in enumerate(productlist):
    if product == targetProduct:
      indices.append(index)


  return indices


# Example usage:
products = ["shoes", "boot", "loafes", "shoes", "sandal", "shoes"]
target = "shoes"
result = linearSearchProduct(products, target)
print(result)
```

Ln 1, Col 1 • Spaces: 2 History

Challenge 3.2 ⋮ Exit

Run 49ms on 18:52:11, 10/24 ✓

```
Name: Charlie, Roll Number: A003, CGPA:
3.9
Name: Alice, Roll Number: A001, CGPA: 3.
8
Name: David, Roll Number: A004, CGPA: 3.
7
Name: Bob, Roll Number: A002, CGPA: 3.5
```

Run 45ms on 18:52:27, 10/24 ✓

```
Name: Charlie, Roll Number: A003, CGPA:
3.9
Name: Alice, Roll Number: A001, CGPA: 3.
8
Name: David, Roll Number: A004, CGPA: 3.
7
Name: Bob, Roll Number: A002, CGPA: 3.5
```

Challenge 3.2 ⋮ Exit

```python
class Student:
    def __init__(self, name, roll_number, cgpa):
        self.name = name
        self.roll_number = roll_number
        self.cgpa = cgpa

def sort_students(student_list):
    sorted_students = sorted(student_list, key=lambda student: student.cgpa, reverse=True)
    return sorted_students

# Example usage
students = [
    Student("Alice", "A001", 3.8),
    Student("Bob", "A002", 3.5),
    Student("Charlie", "A003", 3.9),
    Student("David", "A004", 3.7)
]

# Sort students based on CGPA in descending order
sorted_students = sort_students(students)

# Print the sorted list of students
for student in sorted_students:
    print(f"Name: {student.name},
```

Ln 1, Col 1 • Spaces: 2 History

Challenge 3.1

Exit

Run

155ms on 18:53:00, 10/24 ✓

```
[0, 3, 5]
```

Console

Challenge 2.2 ⋮ Exit

Run 229ms on 18:53:28, 10/24 ✓

```
The batsman is batting.
The bowler is bowling.
```

Challenge 2.2 ⋮ Exit

```python
class Player:
    def play(self):
        print("The player is playing cricket.")

class Batsman(Player):
    def play(self):
        print("The batsman is batting.")

class Bowler(Player):
    def play(self):
        print("The bowler is bowling.")

# Create objects of Batsman and Bowler classes
batsman = Batsman()
bowler = Bowler()

# Call the play() method for each object
batsman.play()
bowler.play()
```

Ln 1, Col 1 • Spaces: 2 History

Challenge 2.1 ⋮ Exit

Run 227ms on 18:53:54, 10/24 ✓

```
Account balance for Hari Prabu (Account
#123456789): 5000.0
Deposited 500.0. New balance: 5500.0
Withdrew 200.0. New balance: 5300.0
Account balance for Hari Prabu (Account
#123456789): 5300.0
```

Challenge 1.2 ⋮ Exit

Run 48ms on 18:54:21, 10/24 ✓

```
Deposited 500. New balance: 1500.0
Withdraw 200. New balance: 1300.0
```