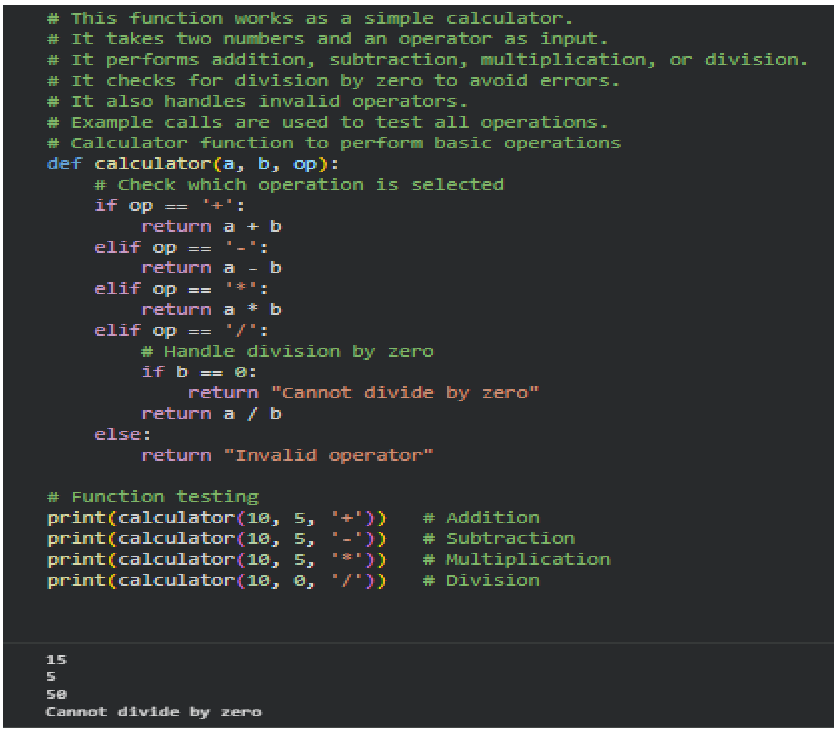
Assingment3.2

T.akanksh

2303A51078

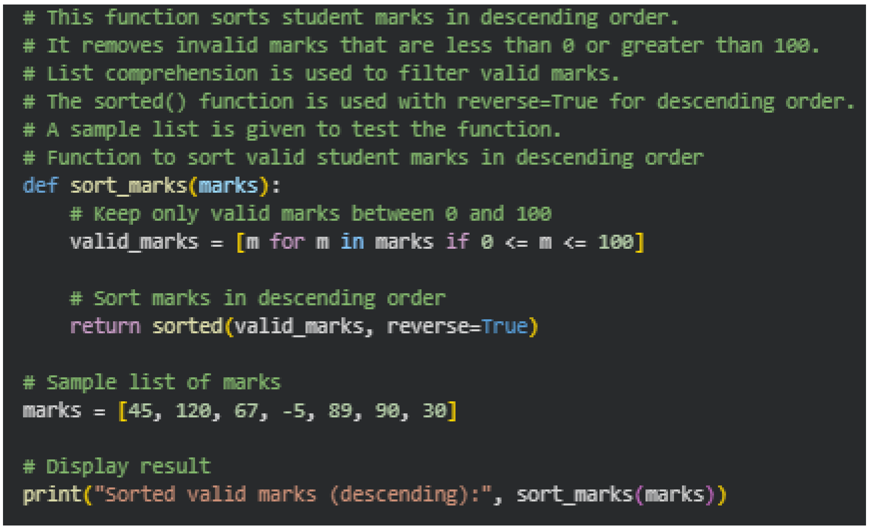
1.calculator design



```
# This function works as a simple calculator.
# It takes two numbers and an operator as input.
# It performs addition, subtraction, multiplication, or division.
# It checks for division by zero to avoid errors.
# It also handles invalid operators.
# Example calls are used to test all operations.
# Calculator function to perform basic operations
def calculator(a, b, op):
    # Check which operation is selected
    if op == '+':
        return a + b
    elif op == '-':
        return a - b
    elif op == '*':
        return a * b
    elif op == '/':
        # Handle division by zero
        if b == 0:
            return "Cannot divide by zero"
        return a / b
    else:
        return "Invalid operator"

# Function testing
print(calculator(10, 5, '+'))   # Addition
print(calculator(10, 5, '-'))   # Subtraction
print(calculator(10, 5, '*'))   # Multiplication
print(calculator(10, 0, '/'))   # Division
```

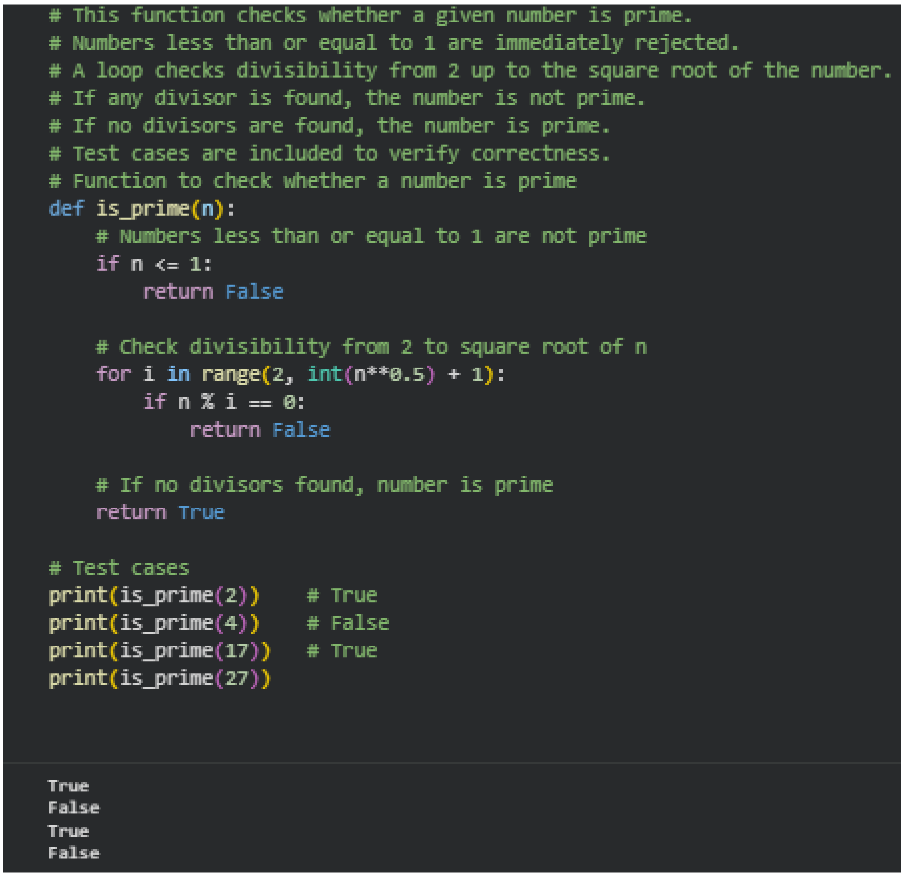
```
15
5
50
Cannot divide by zero
```

2.shorting logic



```
# This function sorts student marks in descending order.
# It removes invalid marks that are less than 0 or greater than 100.
# List comprehension is used to filter valid marks.
# The sorted() function is used with reverse=True for descending order.
# A sample list is given to test the function.
# Function to sort valid student marks in descending order
def sort_marks(marks):
    # Keep only valid marks between 0 and 100
    valid_marks = [m for m in marks if 0 <= m <= 100]

    # Sort marks in descending order
    return sorted(valid_marks, reverse=True)

# Sample list of marks
marks = [45, 120, 67, -5, 89, 90, 30]

# Display result
print("Sorted valid marks (descending):", sort_marks(marks))
```

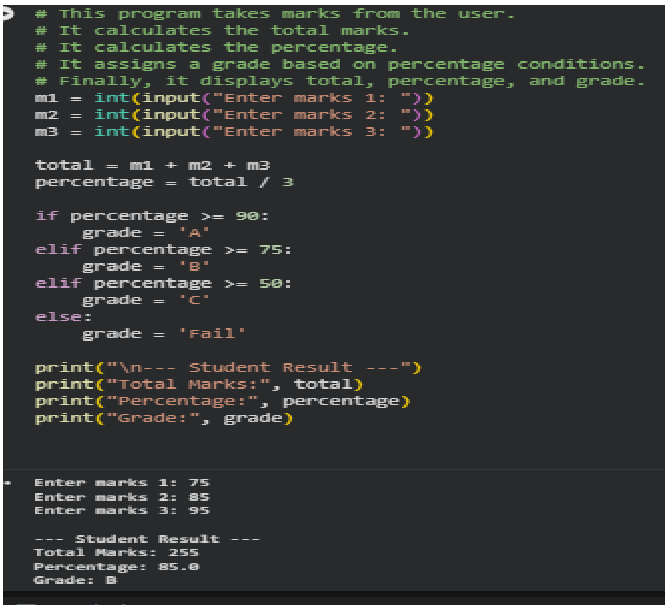
3.prime number validation



```
# This function checks whether a given number is prime.
# Numbers less than or equal to 1 are immediately rejected.
# A loop checks divisibility from 2 up to the square root of the number.
# If any divisor is found, the number is not prime.
# If no divisors are found, the number is prime.
# Test cases are included to verify correctness.
# Function to check whether a number is prime
def is_prime(n):
    # Numbers less than or equal to 1 are not prime
    if n <= 1:
        return False

    # Check divisibility from 2 to square root of n
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False

    # If no divisors found, number is prime
    return True

# Test cases
print(is_prime(2))    # True
print(is_prime(4))    # False
print(is_prime(17))   # True
print(is_prime(27))
```

```
True
False
True
False
```

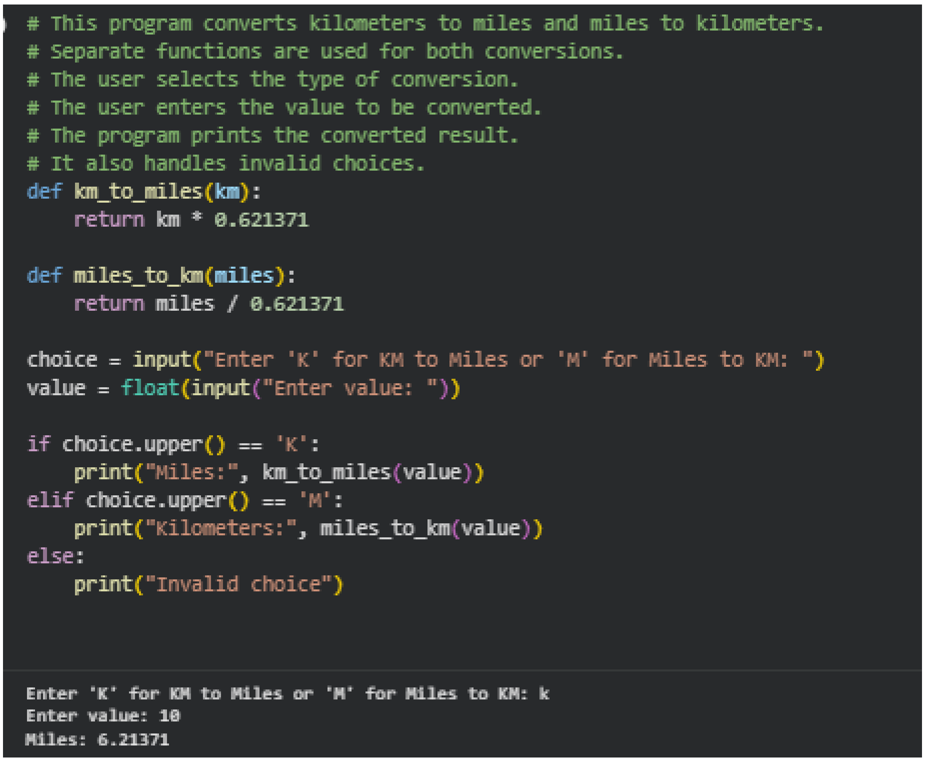
4.student grading system



```
# This program takes marks from the user.
# It calculates the total marks.
# It calculates the percentage.
# It assigns a grade based on percentage conditions.
# Finally, it displays total, percentage, and grade.
m1 = int(input("Enter marks 1: "))
m2 = int(input("Enter marks 2: "))
m3 = int(input("Enter marks 3: "))

total = m1 + m2 + m3
percentage = total / 3

if percentage >= 90:
    grade = 'A'
elif percentage >= 75:
    grade = 'B'
elif percentage >= 50:
    grade = 'C'
else:
    grade = 'Fail'

print("\n--- Student Result ---")
print("Total Marks:", total)
print("Percentage:", percentage)
print("Grade:", grade)
```

```
Enter marks 1: 75
Enter marks 2: 85
Enter marks 3: 95

--- Student Result ---
Total Marks: 255
Percentage: 85.0
Grade: B
```

5.kilometer to meter



```python
# This program converts kilometers to miles and miles to kilometers.
# Separate functions are used for both conversions.
# The user selects the type of conversion.
# The user enters the value to be converted.
# The program prints the converted result.
# It also handles invalid choices.
def km_to_miles(km):
    return km * 0.621371

def miles_to_km(miles):
    return miles / 0.621371

choice = input("Enter 'K' for KM to Miles or 'M' for Miles to KM: ")
value = float(input("Enter value: "))

if choice.upper() == 'K':
    print("Miles:", km_to_miles(value))
elif choice.upper() == 'M':
    print("Kilometers:", miles_to_km(value))
else:
    print("Invalid choice")
```

```
Enter 'K' for KM to Miles or 'M' for Miles to KM: k
Enter value: 10
Miles: 6.21371
```