// Computes SPI (grade list, credits, grade to pointer)

// Input: grade list, credits, grade to pointer
(no negative values) dictionary

// Output: SPI value.

SPI = 0, credits_total = 0
LOOP over the grade list:

SPI += grade_to_pointer [grade]
× credit

~~SPI /= length of cre~~
credit_total += credit

SPI /= credit_total

return SPI

★ Test Cases: ~~① SPI ( [2,~~

① SPI ( ["AA", "BC", "FF"] , [3, 2, 3] ,
{ "AA": 10, "BC": 7, "FF": 0,
"AB": 9, "BB": 8 } )

SPI = 5.5

② SPI ( ["GG", "AA", "BC"], [1, 2, 5],
{ "AA": 10, "BC": 7, "FF": 0, "AB": 9 }

SPI = error in input grades

③ SPI( ["AA", "BC", "BB"], [-1, 0, 3],
{"AA": 10, "BB": 8, "AB": 9, "BC": 7})

error
SPI = credits cant be negative

④ SPI = ( ["AA", "BC"], [5, 3], {"BB": 8, "BC": 7})

SPI = error grade not in the converter dictionary

⑤ SPI = ( ["BB", "BC", "BC"], [1, 1, 1],
{"BB": 8, "AA": 10, "BC": 7})

SPI = 7.33

// Computes CPI (SPI) using the provided formula
// Input: SPI array containing all SPI ($0 \le SPI \le 10$)
// Output: CPI value
CPI = 0
LOOP over SPI array:
CPI += SPI

return CPI / length of SPI array.

* Test Cases:

① CPI([5, 7, 8, 8.5])

CPI = 7.125

② CPI([-1, 0, 5, 8])

CPI = error SPI cant be negative

③ CPI([5, 8, 9, 11])

CPI = error SPI cant be greater than 10

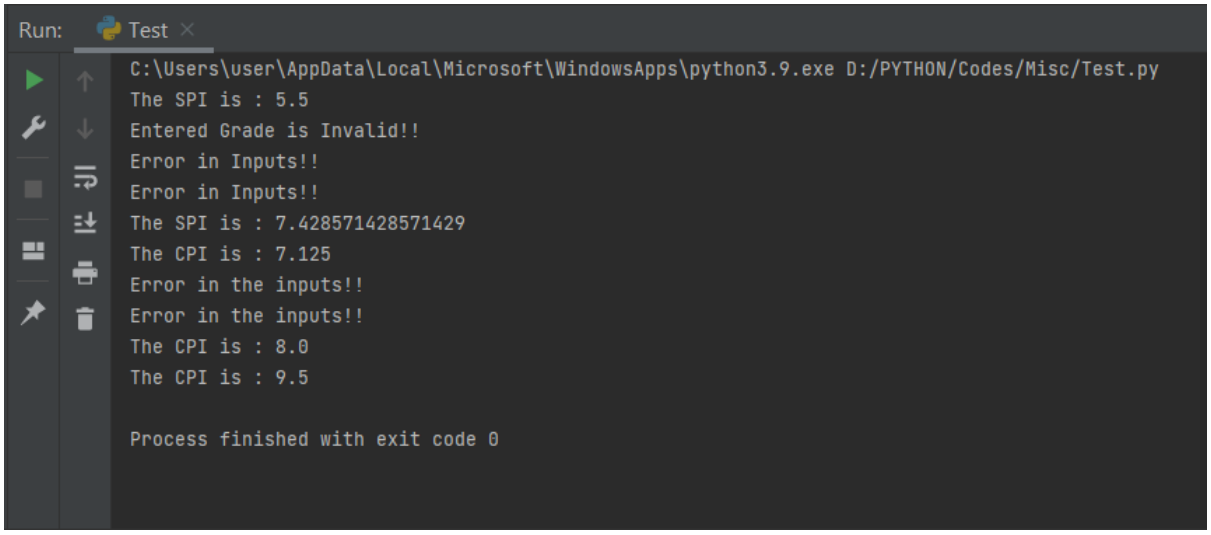
④ CPI([8, 8, 8, 8, 8])
CPI = 8.0

⑤ CPI([9, 10, 9, 10])
CPI = 9.5

Code:

```
def SPI(grades, credits, convert):
    pointer = 0
    total_credits = 0
    if len(grades) != len(credits):
        return "Error in Inputs!!"
    for i in credits:
        if i < 0:
            return "Error in Inputs!!"
    for i in range(len(grades)):
        if grades[i] in convert:
            pointer += (convert[grades[i]] * credits[i])
            total_credits += credits[i]
        else:
            return "Entered Grade is Invalid!!"

    return "The SPI is : " + str(pointer/total_credits)


def CPI(pointers):
    ans = 0
    for i in pointers:
        if i > 10 or i < 0:
            return "Error in the inputs!!"
        ans += i

    return "The CPI is : " + str(ans / len(pointers))


convert = {"AA": 10, "AB": 9, "BB": 8, "BC": 7, "CC": 6, "CD": 5, "DD": 4,
"FF": 0, "RR": 0}
print(SPI(["AA", "BC", "FF"], [3, 2, 3], convert))
print(SPI(["GG", "BC", "FF"], [1, 2, 5], convert))
print(SPI(["AA", "DD", "CD"], [-1, 2, 5], convert))
print(SPI(["GG", "BB"], [1, 2, 5], convert))
print(SPI(["AA", "AA", "DD"], [2, 2, 3], convert))

print(CPI([5, 7, 8, 8.5]))
print(CPI([-1, 0, 5, 8]))
print(CPI([5, 8, 9, 11]))
print(CPI([8, 8, 8, 8, 8, 8]))
print(CPI([9, 10, 9, 10]))
```

Output:




Conclusion:

The code implemented in this lab assignment effectively calculates the Semester Performance Index (SPI) and Cumulative Performance Index (CPI) using student grades and credit data. It includes error handling for invalid inputs, ensuring robustness and reliability.