

ALAN GEORGE MATHEWS CS6A 11

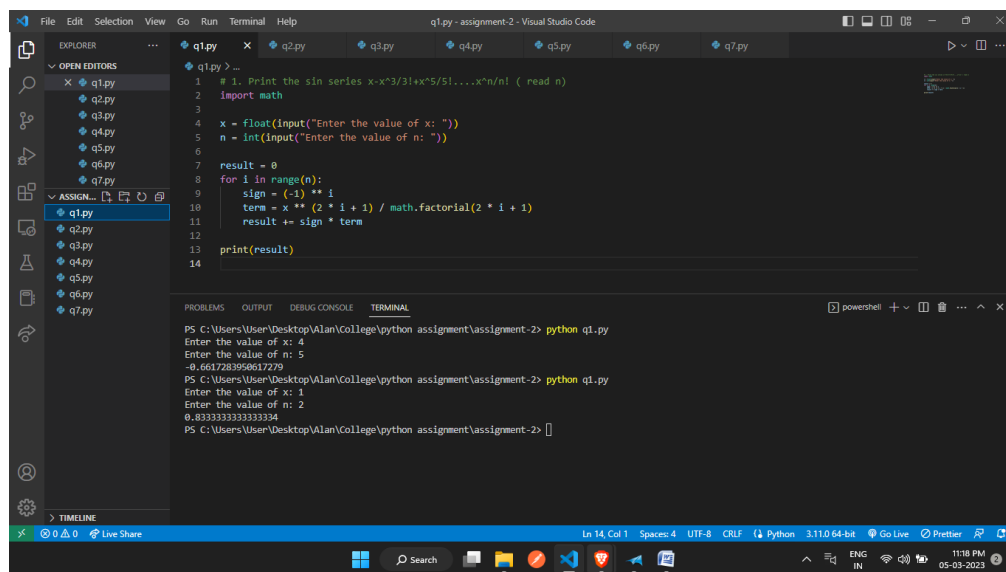
Programming in Python CST 362

Assignment 2

Learning outcome: Learn to use loops ,nested loops and strings

Date of submission:6-March-2023 before 12pm

1. Print the sin series $x - x^3/3! + x^5/5! - \dots x^n/n!$ (read n)

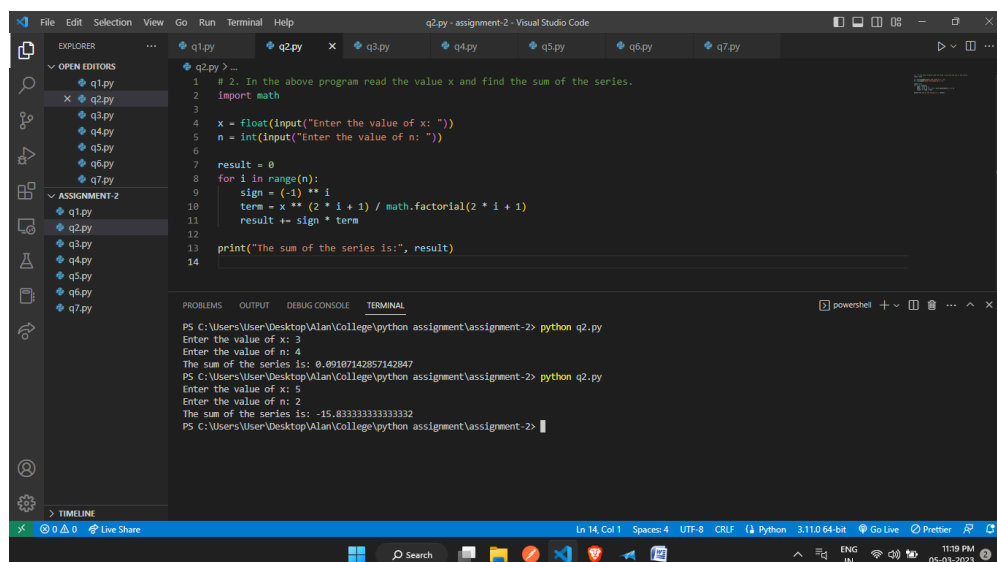


```
1 # 1. Print the sin series x-x^3/3!+x^5/5!....x^n/n! ( read n )
2 import math
3
4 x = float(input("Enter the value of x: "))
5 n = int(input("Enter the value of n: "))
6
7 result = 0
8 for i in range(n):
9     sign = (-1) ** i
10    term = x ** (2 * i + 1) / math.factorial(2 * i + 1)
11    result += sign * term
12
13 print(result)
14
```

Terminal output:

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q1.py
Enter the value of x: 4
Enter the value of n: 5
-0.6617283959617279
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q1.py
Enter the value of x: 1
Enter the value of n: 2
0.03333333333333334
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

2. In the above program read the value x and find the sum of the series.



```
1 # 2. In the above program read the value x and find the sum of the series.
2 import math
3
4 x = float(input("Enter the value of x: "))
5 n = int(input("Enter the value of n: "))
6
7 result = 0
8 for i in range(n):
9     sign = (-1) ** i
10    term = x ** (2 * i + 1) / math.factorial(2 * i + 1)
11    result += sign * term
12
13 print("The sum of the series is:", result)
14
```

Terminal output:

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q2.py
Enter the value of x: 3
Enter the value of n: 4
The sum of the series is: 0.89187142057142047
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q2.py
Enter the value of x: 5
Enter the value of n: 2
The sum of the series is: -15.833333333333332
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

3.

1

1 2

1 2 3

1 2 3 4

The pyramid is given for n=4 do this for any n

```

1 #1 2
2 #1 2 3
3 #1 2 3 4
4 #The pyramid is given for n=4 do this for any n
5 n = int(input("Enter the value of n: "))
6
7
8
9 for i in range(1, n+1):
10     for j in range(1, i+1):
11         print(j, end=" ")
12     print()
13

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q3.py
Enter the value of n: 4
1
1 2
1 2 3
1 2 3 4
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q3.py
Enter the value of n: 6
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

```

4. Reverse a number and also find the sum of digits

```

1 #4. Reverse a number and also find the sum of the digits
2 #Eg: 1/p: 546 o/p: reverse=645 sum=15
3 num = int(input("Enter a number: "))
4
5 reverse_num = 0
6 while num > 0:
7     remainder = num % 10
8     reverse_num = (reverse_num * 10) + remainder
9     num = num // 10
10
11 sum_of_digits = 0
12 for digit in str(reverse_num):
13     sum_of_digits += int(digit)
14
15 print("Reverse:", reverse_num)
16 print("Sum of digits:", sum_of_digits)
17

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q4.py
Enter a number: 768
Reverse: 867
Sum of digits: 21
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q4.py
Enter a number: 251
Reverse: 152
Sum of digits: 8
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

```

5. Positive integer is called an Armstrong number of order n if $abcd.... = a^n + b^n + c^n + d^n + \dots$ where n is the length of the number Eg: $153 = 1^1 + 5^5 + 3^3 = 1 + 125 + 27 = 153$ is an Armstrong Number. Eg: $1634 = 1^1 + 6^6 + 3^3 + 4^4 = 1 + 216 + 27 + 256 = 490$ is not an Armstrong Number.

```

1 # 5. Positive integer is called an Armstrong number of order n if
2 #abcd... = a^n + b^n + c^n + d^n + ... where n is the length of the number
3 #Eg: 153 = 1^1 + 5^5 + 3^3 // 153 is an Armstrong Number.
4 #Eg: 1634 = 1^1 + 6^6 + 3^3 + 4^4 = 1634 // 1634 is an Armstrong Number.
5 num = int(input("Enter a positive integer: "))
6 order = len(str(num))
7 sum = 0
8
9 temp = num
10 while temp > 0:
11     digit = temp % 10
12     sum += digit ** order
13     temp //= 10
14
15 if num == sum:
16     print(num, "is an Armstrong number!")
17 else:
18     print(num, "is not an Armstrong number.")

```

Terminal Output:

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q5.py
Enter a positive integer: 153
153 is an Armstrong number!
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q5.py
Enter a positive integer: 1634
1634 is an Armstrong number!
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q5.py
Enter a positive integer: 463
463 is not an Armstrong number.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

```

6. Find the square root of a number using Newton's method (refer the text book/blog for reference)

```

1 # 6. Find the square root of a number using Newton's method ( refer the text book/blog for reference)
2 n = float(input("Enter a number: "))
3
4 guess = n / 2
5 tolerance = 0.0001
6
7 while abs(guess ** 2 - n) > tolerance:
8     guess = (guess + n / guess) / 2
9
10 print("The square root of", n, "is", guess)

```

Terminal Output:

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q6.py
Enter a number: 16
The square root of 16.0 is 4.00000185845895
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q6.py
Enter a number: 56
The square root of 56.0 is 7.48331513811743
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q6.py
Enter a number: 64
The square root of 64.0 is 8.00000127385879
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

```

7. Write a program that computes an investment report. The inputs to this program are the following: An initial amount to be invested (a floating-point number), A period of years (an integer), An interest rate (a percentage expressed as an integer)

```

10 print("Year\tStarting Balance\tInterest\tEnding Balance")
11 print("-" * 50)
12
13 total_interest = 0
14
15 for year in range(1, years+1):
16     interest = round(initial_amount * interest_rate / 100, 2)
17     ending_balance = round(initial_amount + interest, 2)
18     print(f"{year}\t{initial_amount}\t\t{interest}\t\t{ending_balance}")
19     total_interest += interest
20     initial_amount = ending_balance
21
22 print("-" * 50)
23 print(f"Ending balance: {round(ending_balance, 2)}")
24 print(f"Total Interest Earned: {round(total_interest, 2)}")
25
26

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q7.py
Enter the initial amount to be invested: 10000
Enter the period of years: 5
Enter the interest rate: 5

Year	Starting Balance	Interest	Ending Balance
1	10000.0	500.0	10500.0
2	10500.0	525.0	11025.0
3	11025.0	551.25	11576.25
4	11576.25	578.81	12155.06
5	12155.06	607.75	12762.81

Ending balance: 12762.81
Total Interest Earned: 2762.81

8. Check whether the given number is a Krishnamurti number (.Use factorial () function from math) For example: $145 = 1! + 4! + 5! = 1 + 24 + 120 = 145$ is a Krishnamurthy Number

```

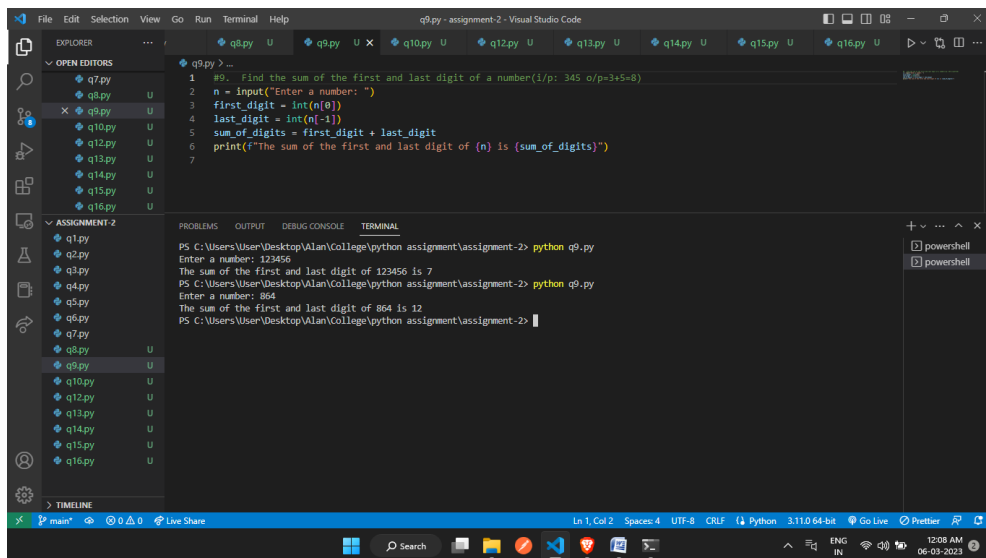
1 #8. Check whether the given number is a Krishnamurti number(.Use factorial() function from math)
2 > #For example: 145 = 1! + 4! + 5! = 1 + 24 + 120 = 145 is a Krishnamurthy Number...
3
4
5 n = int(input("Enter a number: "))
6 digits = str(n)
7 fact_sum = sum([math.factorial(int(digit)) for digit in digits])
8 if fact_sum == n:
9     print(f"{n} is a Krishnamurti number")
10 else:
11     print(f"{n} is not a Krishnamurti number")

```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q8.py
Enter a number: 145
145 is a Krishnamurti number

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q8.py
Enter a number: 456
456 is not a Krishnamurti number

9. Find the sum of the first and last digit of a number(i/p:345 o/p=3+5=8)

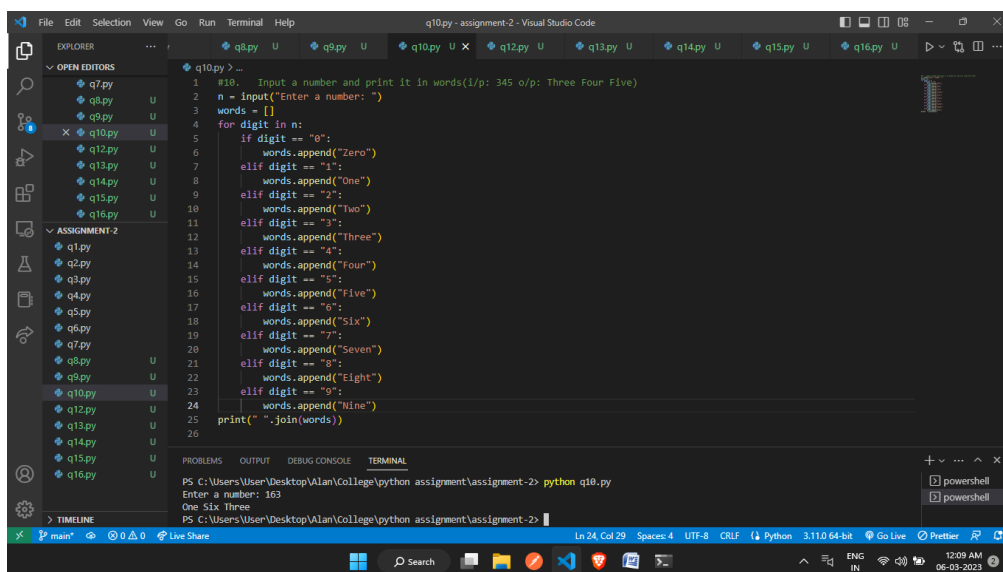


```
1 #9. Find the sum of the first and last digit of a number(i/p: 345 o/p=3+5=8)
2 n = input("Enter a number: ")
3 first_digit = int(n[0])
4 last_digit = int(n[-1])
5 sum_of_digits = first_digit + last_digit
6 print(f"The sum of the first and last digit of (n) is (sum_of_digits)")
7
```

Terminal output:

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q9.py
Enter a number: 123456
The sum of the first and last digit of 123456 is 7
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q9.py
Enter a number: 864
The sum of the first and last digit of 864 is 12
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

10. Input a number and print it in words (i/p:345 o/p: Three Four Five)



```
1 #10. Input a number and print it in words(i/p: 345 o/p: Three Four Five)
2 n = input("Enter a number: ")
3 words = []
4 for digit in n:
5     if digit == "0":
6         words.append("Zero")
7     elif digit == "1":
8         words.append("One")
9     elif digit == "2":
10        words.append("Two")
11    elif digit == "3":
12        words.append("Three")
13    elif digit == "4":
14        words.append("Four")
15    elif digit == "5":
16        words.append("Five")
17    elif digit == "6":
18        words.append("Six")
19    elif digit == "7":
20        words.append("Seven")
21    elif digit == "8":
22        words.append("Eight")
23    elif digit == "9":
24        words.append("Nine")
25 print(" ".join(words))
26
```

Terminal output:

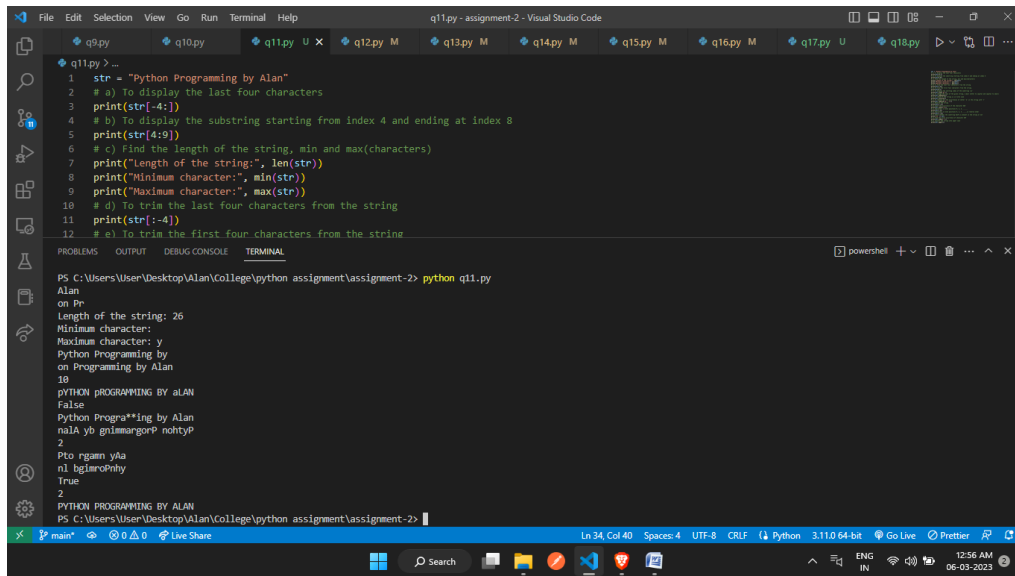
```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q10.py
Enter a number: 163
One Six Three
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

Strings

Outcome: Learn String Indexing and slicing, programming with strings

Consider the string `str="Python Programming by Yourname"` (Replace Yourname with your first name) Write statements in python to implement the following

- a) To display the last four characters.
- b) To display the substring starting from index 4 and ending at index 8.
- c) Find the length of the string,min and max(characters)
- d) To trim the last four characters from the string.
- e) To trim the first four characters from the string.
- f) To display the starting index of the substring 'gr'.
- g) To change the case of the given string.(small letter to capital and capital to small)
- h) To check if the string is in title case.
- i) To replace all the occurrences of letter 'm' in the string with '**'
- j)reverse the string
- k)count the occurrence of the character 'm'
- l)characters in even positions 0,2,4,....
- m) characters in even positions 0,2,4,.....in reverse order
- n)check whether the substring 'on' is present in the string or not
- o)Find the first occurrence of character 't'
- p)convert the string into upper case



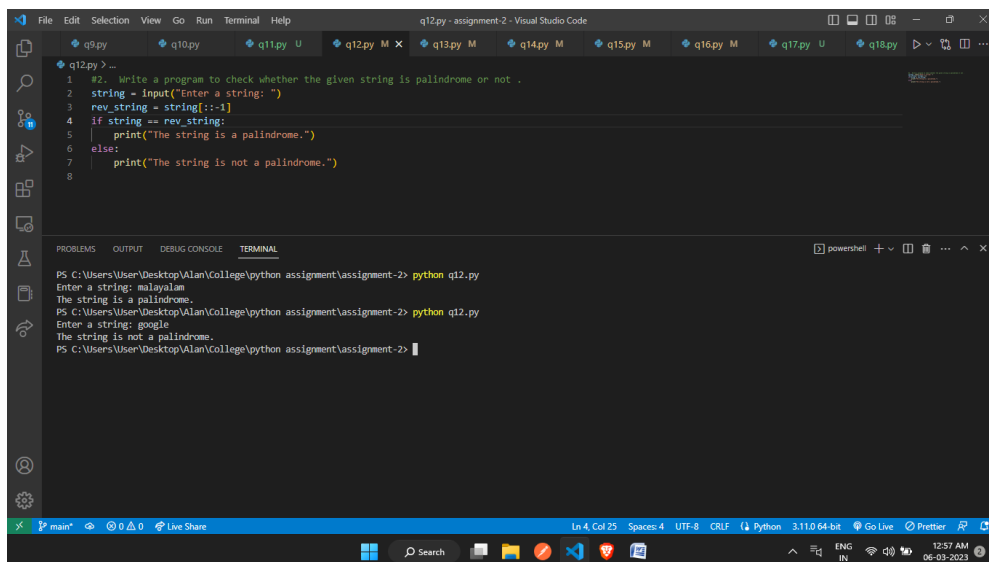
The screenshot shows a Visual Studio Code window with a Python file named `q11.py`. The code defines a string `str = "Python Programming by Alan"` and performs several operations: printing the last four characters, a substring from index 4 to 8, the length of the string and its minimum and maximum characters, and trimming the last four characters. The terminal output shows the execution of `python q11.py`, displaying the results of these operations.

```
1 str = "Python Programming by Alan"
2 # a) To display the last four characters
3 print(str[-4:])
4 # b) To display the substring starting from index 4 and ending at index 8
5 print(str[4:9])
6 # c) Find the length of the string, min and max(characters)
7 print("Length of the string:", len(str))
8 print("Minimum character:", min(str))
9 print("Maximum character:", max(str))
10 # d) To trim the last four characters from the string
11 print(str[:-4])
12 # e) To trim the first four characters from the string
```

Terminal Output:

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q11.py
Alan
on Pr
Length of the string: 26
Minimum character:
Maximum character: y
Python Programming by
on Programming by Alan
18
PYTHON PROGRAMMING BY aLAN
False
Python Progra**ing by Alan
naIA yb gnimmargonP nohtyP
2
Pto rgam yaa
nl lgarschvity
True
2
PYTHON PROGRAMMING BY ALAN
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

2. Write a program to check whether the given string is palindrome or not.



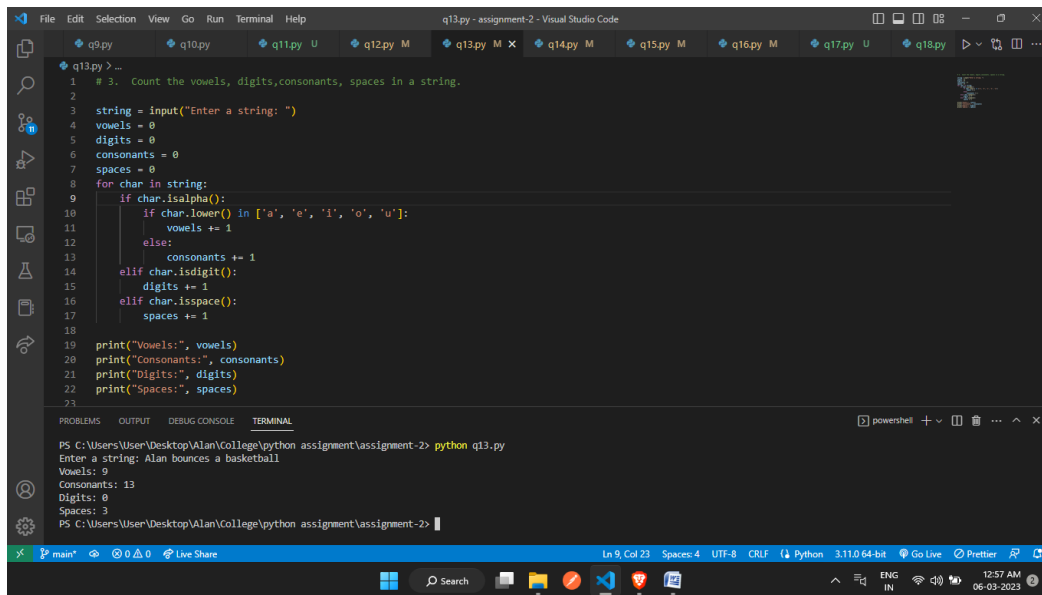
The screenshot shows a Visual Studio Code window with a Python file named `q12.py`. The code prompts the user to enter a string, reverses it, and checks if it is a palindrome. The terminal output shows the execution of `python q12.py` with two test cases: "malayalam" (which is a palindrome) and "google" (which is not a palindrome).

```
1 #2. Write a program to check whether the given string is palindrome or not .
2 string = input("Enter a string: ")
3 rev_string = string[::-1]
4 if string == rev_string:
5     print("The string is a palindrome.")
6 else:
7     print("The string is not a palindrome.")
8
```

Terminal Output:

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q12.py
Enter a string: malayalam
The string is a palindrome.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q12.py
Enter a string: google
The string is not a palindrome.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

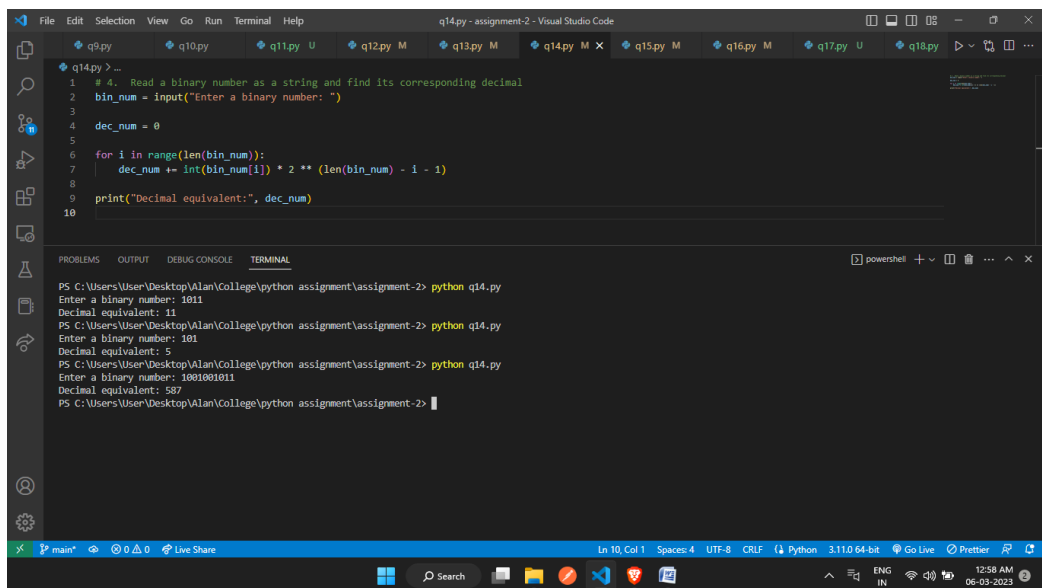
3. Count the vowels, digits, consonants, spaces in a string.



```
1 # 3. Count the vowels, digits, consonants, spaces in a string.
2
3 string = input("Enter a string: ")
4 vowels = 0
5 digits = 0
6 consonants = 0
7 spaces = 0
8 for char in string:
9     if char.isalpha():
10         if char.lower() in ['a', 'e', 'i', 'o', 'u']:
11             vowels += 1
12         else:
13             consonants += 1
14     elif char.isdigit():
15         digits += 1
16     elif char.isspace():
17         spaces += 1
18
19 print("Vowels:", vowels)
20 print("Consonants:", consonants)
21 print("Digits:", digits)
22 print("Spaces:", spaces)
23
```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q13.py
Enter a string: Alan bounces a basketball
Vowels: 9
Consonants: 13
Digits: 0
Spaces: 3
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

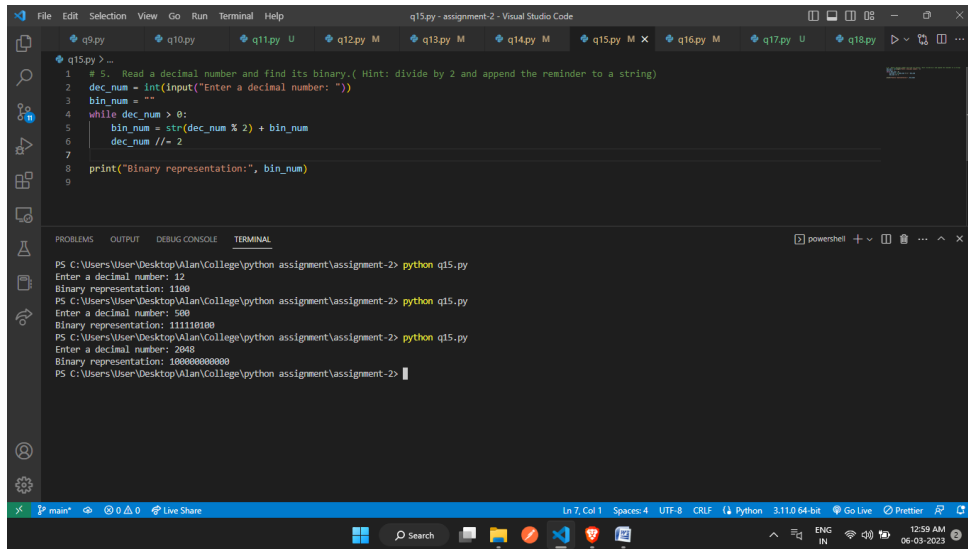
4. Read a binary number as a string and find its corresponding decimal



```
1 # 4. Read a binary number as a string and find its corresponding decimal
2 bin_num = input("Enter a binary number: ")
3
4 dec_num = 0
5
6 for i in range(len(bin_num)):
7     dec_num += int(bin_num[i]) * 2 ** (len(bin_num) - i - 1)
8
9 print("Decimal equivalent:", dec_num)
10
```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q14.py
Enter a binary number: 1011
Decimal equivalent: 11
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q14.py
Enter a binary number: 101
Decimal equivalent: 5
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q14.py
Enter a binary number: 1001001011
Decimal equivalent: 587
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>

5. Read a decimal number and find its binary.(Hint: divide by 2 and append the remainder to a string)

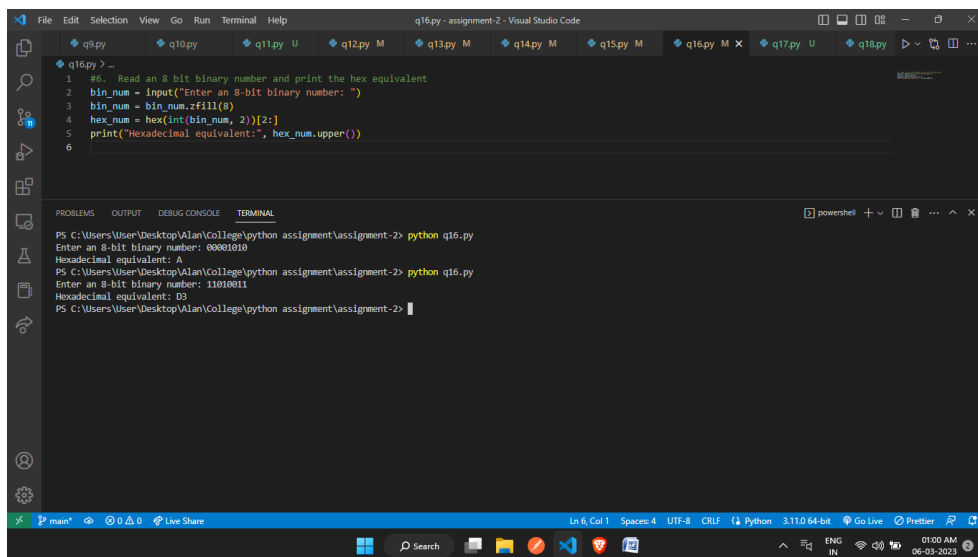


```
q15.py > ...
1 # 5. Read a decimal number and find its binary.( Hint: divide by 2 and append the remainder to a string)
2 dec_num = int(input("Enter a decimal number: "))
3 bin_num = ""
4 while dec_num > 0:
5     bin_num = str(dec_num % 2) + bin_num
6     dec_num //= 2
7
8 print("Binary representation:", bin_num)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q15.py
Enter a decimal number: 1100
Binary representation: 1100
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q15.py
Enter a decimal number: 500
Binary representation: 111110100
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q15.py
Enter a decimal number: 2048
Binary representation: 100000000000
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

6. Read an 8 bit binary number and print the hex equivalent

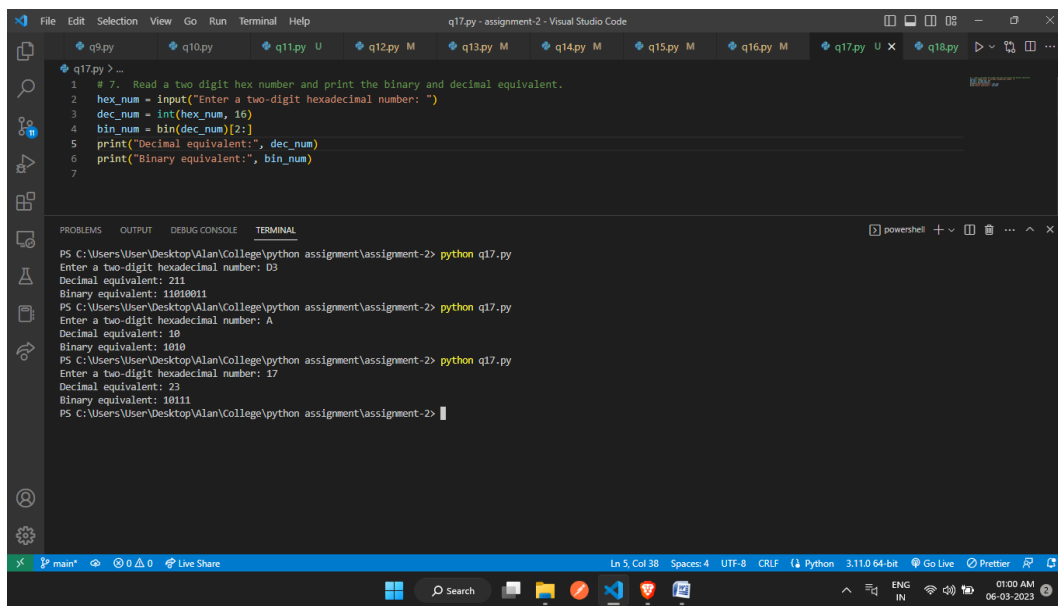


```
q16.py > ...
1 #6. Read an 8 bit binary number and print the hex equivalent
2 bin_num = input("Enter an 8-bit binary number: ")
3 bin_num = bin_num.zfill(8)
4 hex_num = hex(int(bin_num, 2))[2:]
5 print("Hexadecimal equivalent:", hex_num.upper())
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q16.py
Enter an 8-bit binary number: 00001010
Hexadecimal equivalent: A
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q16.py
Enter an 8-bit binary number: 11010011
Hexadecimal equivalent: D3
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

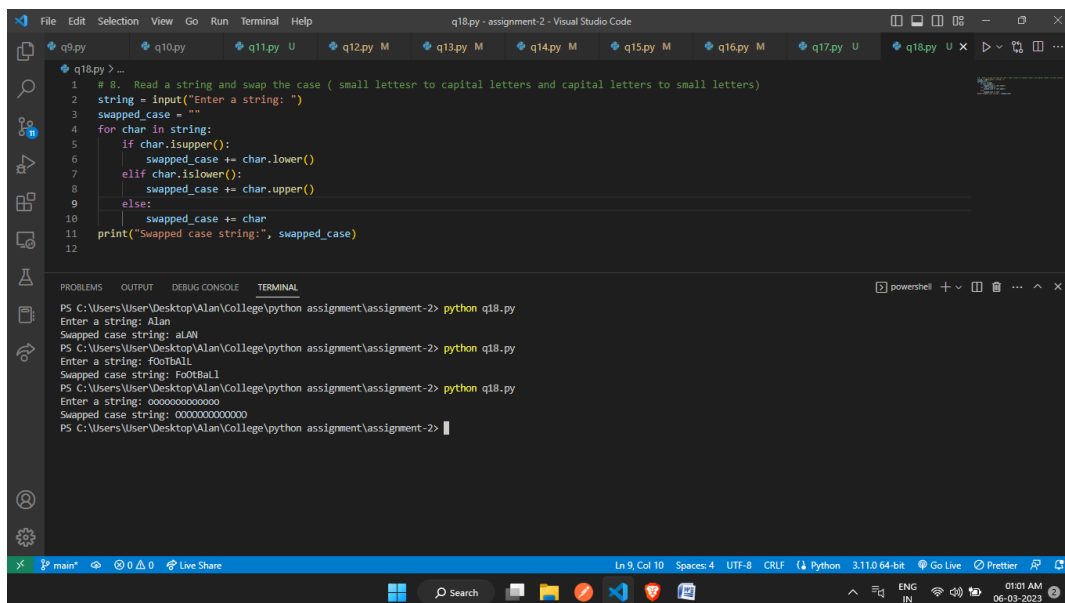
7. Read a two digit hex number and print the binary and decimal equivalent.



```
1 # 7. Read a two digit hex number and print the binary and decimal equivalent.
2 hex_num = input("Enter a two-digit hexadecimal number: ")
3 dec_num = int(hex_num, 16)
4 bin_num = bin(dec_num)[2:]
5 print("Decimal equivalent:", dec_num)
6 print("Binary equivalent:", bin_num)
7
```

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q17.py
Enter a two-digit hexadecimal number: 03
Decimal equivalent: 3
Binary equivalent: 11
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q17.py
Enter a two-digit hexadecimal number: A
Decimal equivalent: 10
Binary equivalent: 1010
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q17.py
Enter a two-digit hexadecimal number: 17
Decimal equivalent: 23
Binary equivalent: 10111
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

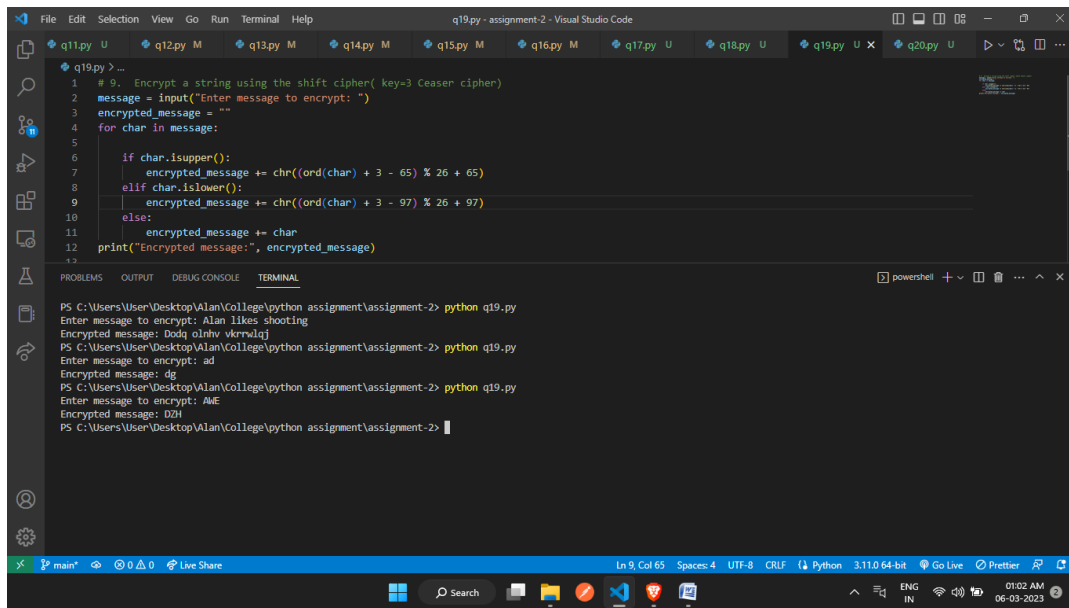
8. Read a string and swap the case (small letters to capital letters and capital letters to small letters)



```
1 # 8. Read a string and swap the case ( small letters to capital letters and capital letters to small letters)
2 string = input("Enter a string: ")
3 swapped_case = ""
4 for char in string:
5     if char.isupper():
6         swapped_case += char.lower()
7     elif char.islower():
8         swapped_case += char.upper()
9     else:
10        swapped_case += char
11 print("Swapped case string:", swapped_case)
12
```

```
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q18.py
Enter a string: Alan
Swapped case string: aLAN
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q18.py
Enter a string: FoOtBa1l
Swapped case string: FoOtBa1l
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q18.py
Enter a string: 0000000000000000
Swapped case string: 0000000000000000
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2>
```

9. Encrypt a string using the shift cipher(key=3 Ceaser cipher)



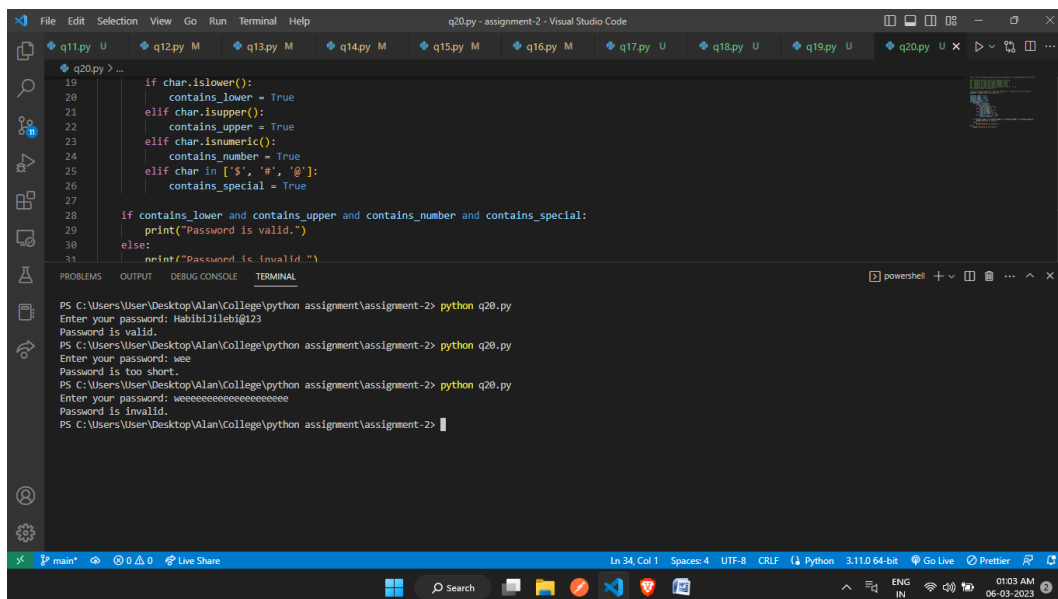
```
1 # 9. Encrypt a string using the shift cipher( key=3 Ceaser cipher)
2 message = input("Enter message to encrypt: ")
3 encrypted_message = ""
4 for char in message:
5
6     if char.isupper():
7         encrypted_message += chr((ord(char) + 3 - 65) % 26 + 65)
8     elif char.islower():
9         encrypted_message += chr((ord(char) + 3 - 97) % 26 + 97)
10    else:
11        encrypted_message += char
12    print("Encrypted message:", encrypted_message)
```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q19.py
Enter message to encrypt: Alan likes shooting
Encrypted message: Dohq olhvw vkrndoj
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q19.py
Enter message to encrypt: ad
Encrypted message: dg
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q19.py
Enter message to encrypt: AME
Encrypted message: DQI
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> |

10. Write a Python program to check the validity of a password given by the user.

The Password should satisfy the following criteria:

1. Contains at least one letter between a and z
2. Contains at least one number between 0 and 9
3. Contains at least one letter between A and Z
4. Contains at least one special character from \$, #, @
5. Minimum length of password: 8



```
19 if char.islower():
20     contains_lower = True
21 elif char.isupper():
22     contains_upper = True
23 elif char.isnumeric():
24     contains_number = True
25 elif char in ['$','#','@']:
26     contains_special = True
27
28 if contains_lower and contains_upper and contains_number and contains_special:
29     print("Password is valid.")
30 else:
31     print("Password is invalid.")
```

PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q20.py
Enter your password: Habibi11lebig123
Password is valid.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q20.py
Enter your password: wee
Password is too short.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> python q20.py
Enter your password: weeeeeeeeeeeeeeeee
Password is invalid.
PS C:\Users\User\Desktop\Alan\College\python assignment\assignment-2> |

