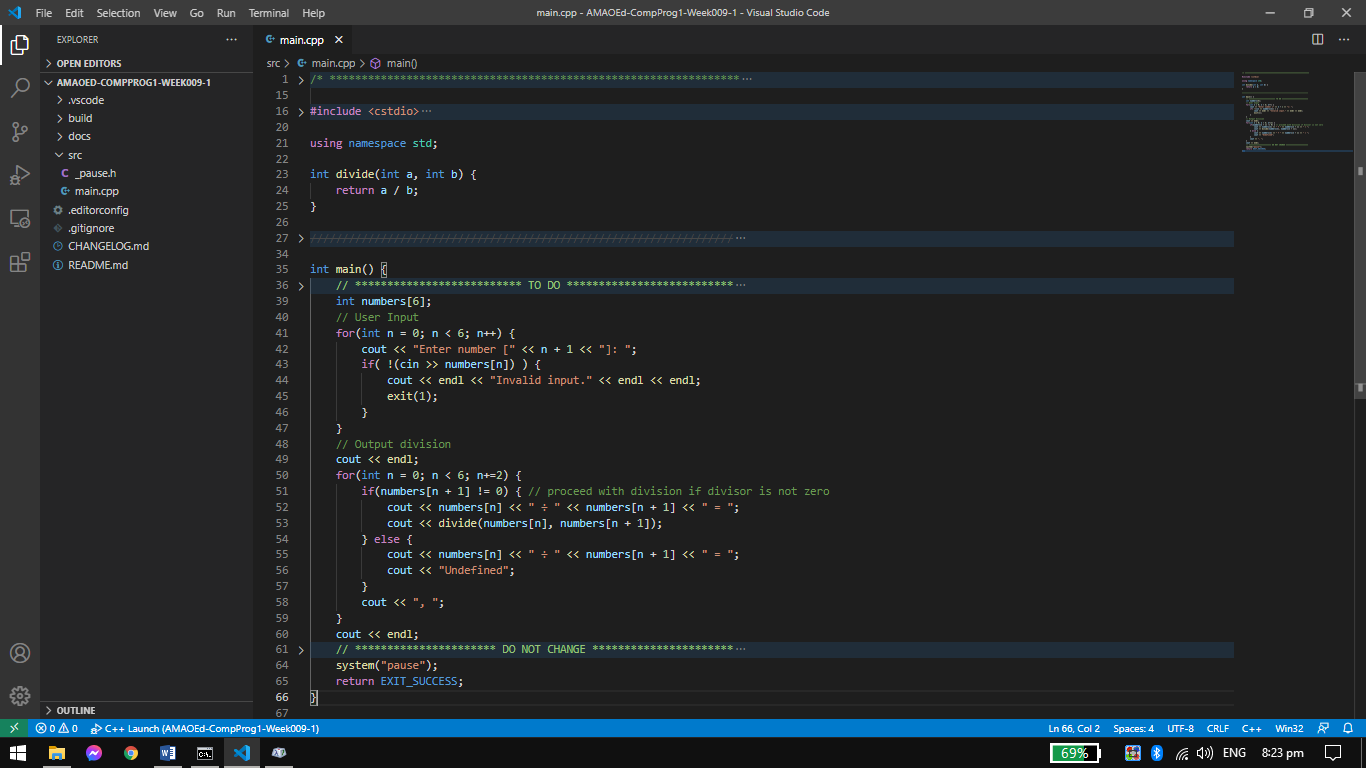
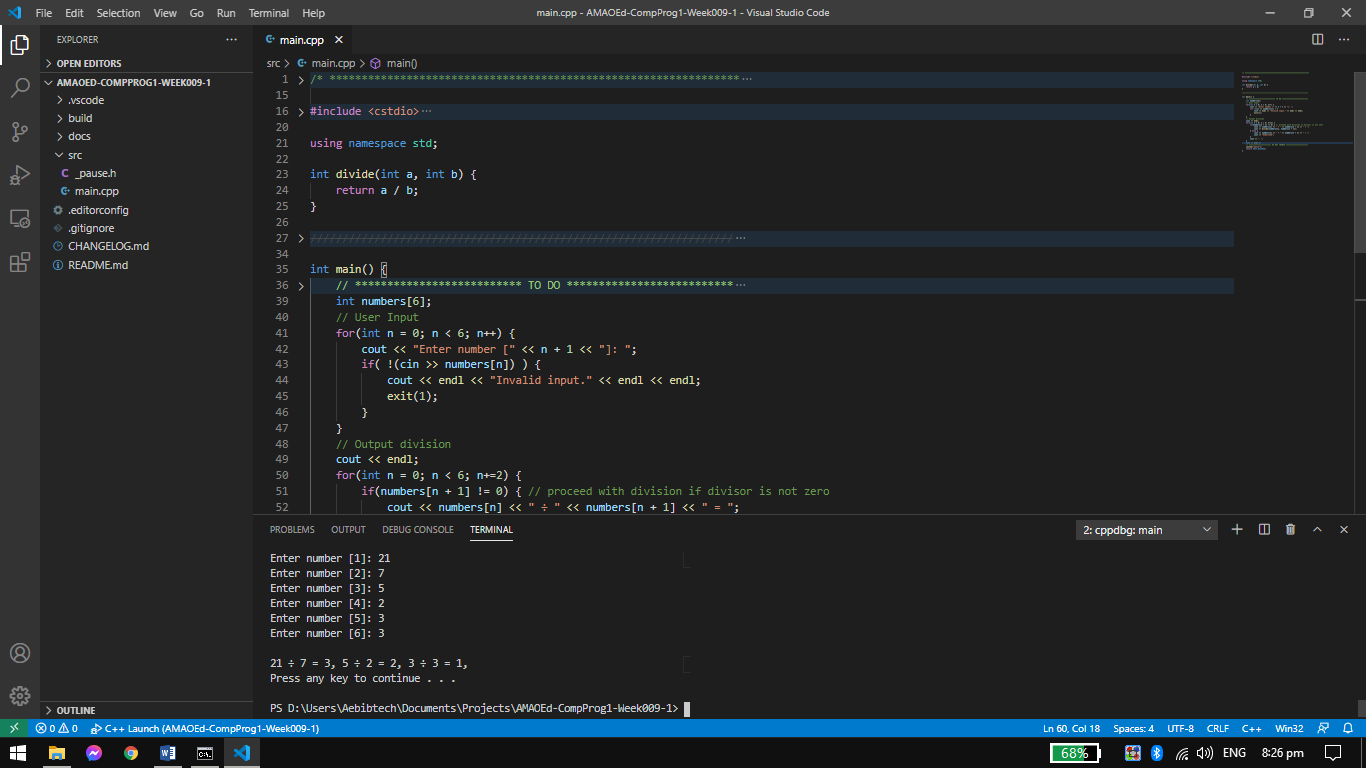
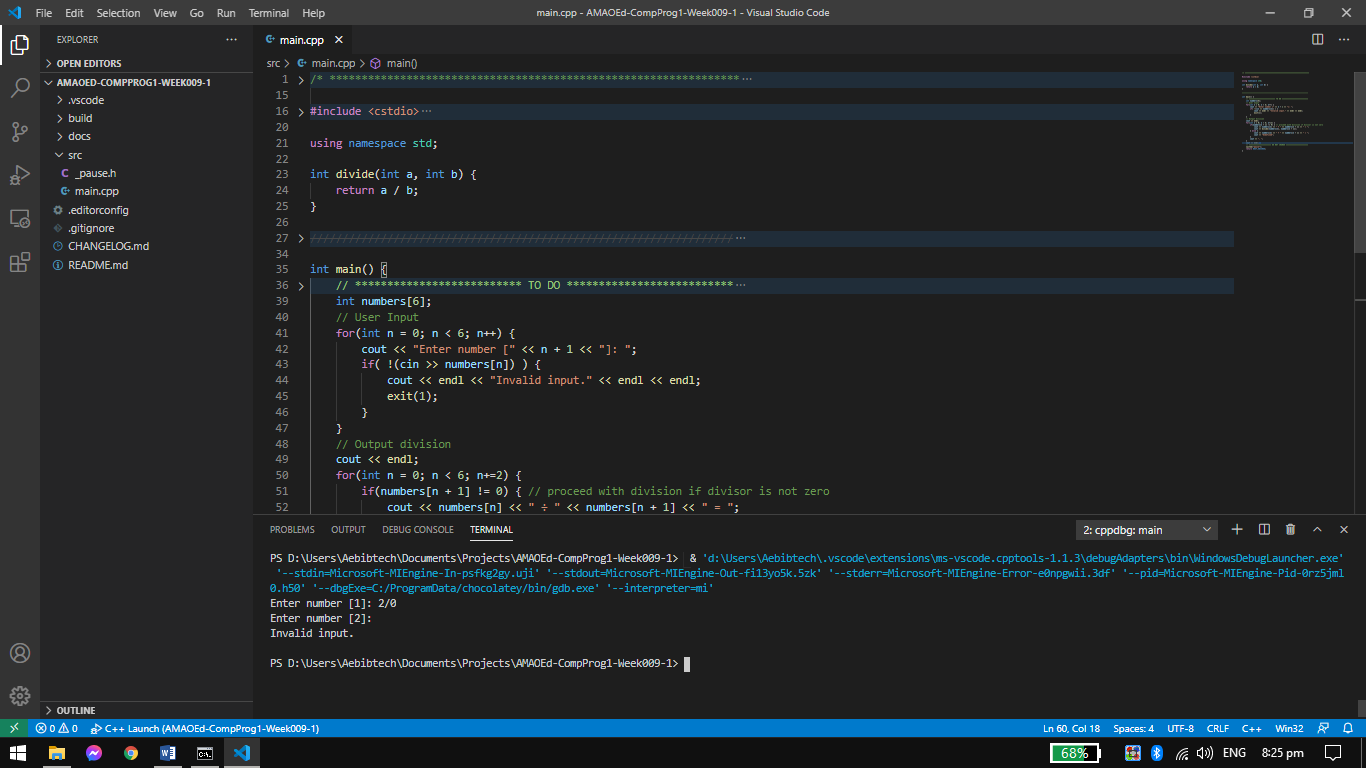
1. Write a program that can divide six non-zero integers (two integers per division) from the user and display the result to the user. Create a function that will perform the division operation. Display only the non-decimal part of the quotient.
   1. Code



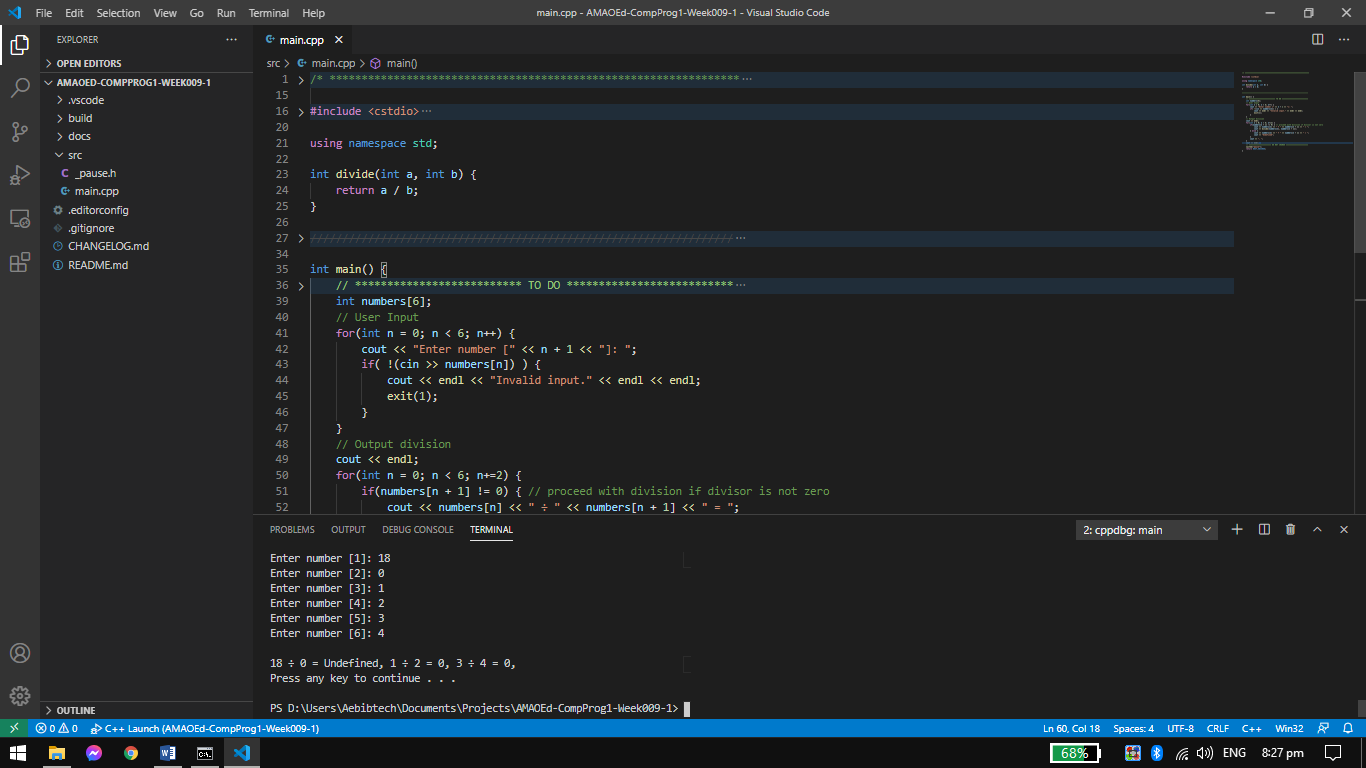
* 1. Sample Input 1



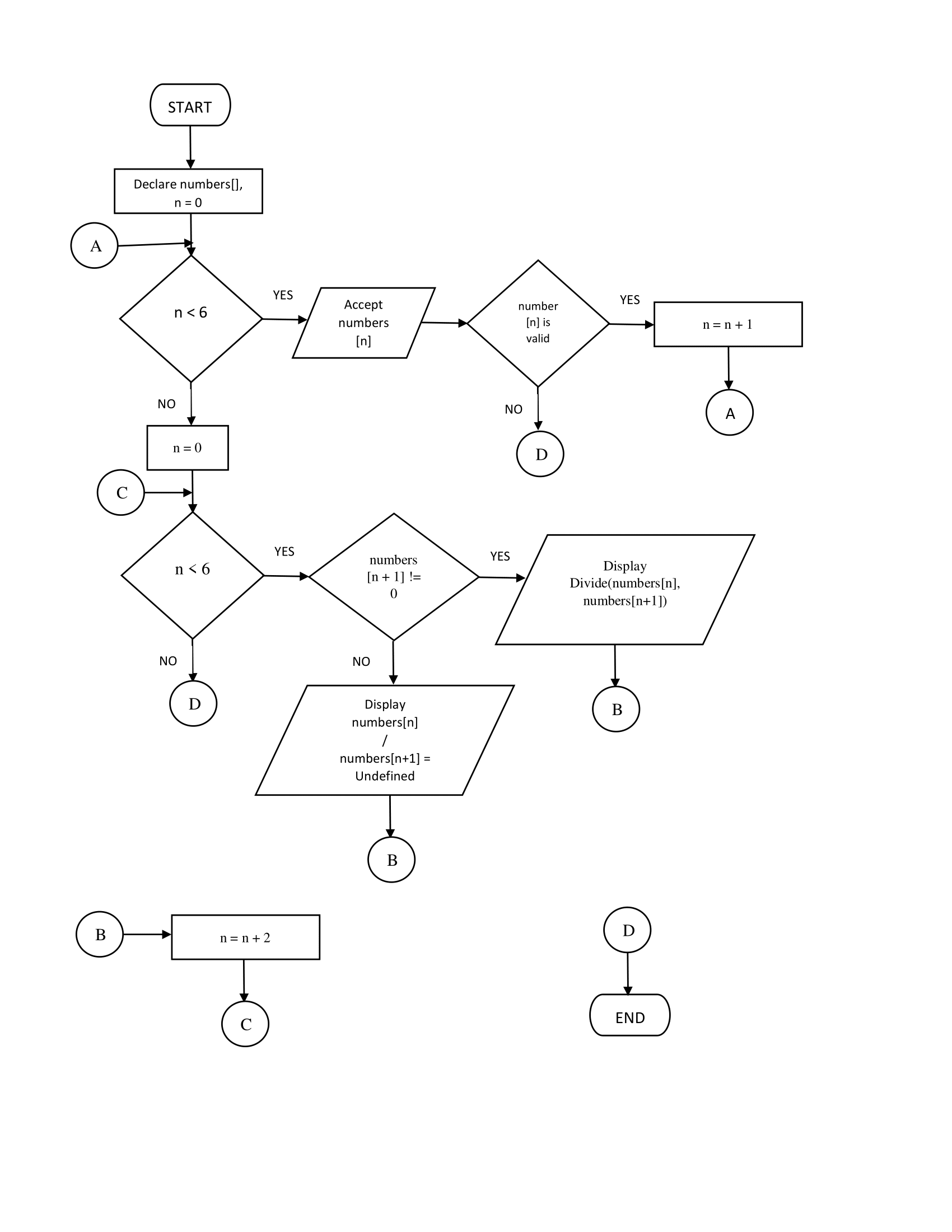
* 1. Sample Input 2



* 1. Sample Input 3



* 1. Flowchart



* 1. Pseudocode

DECLARE numbers[], n

SET n AS 0

WHILE n < 6 DO

INPUT numbers[n]

IF NotValid(numbers[n]) THEN

Call Exit()

END

SET n AS n + 1

LOOP

SET n AS 0

WHILE n < 6 DO

IF numbers[n+1] != 0 THEN

OUTPUT Divide(numbers[n], numbers[n+1])

ELSE

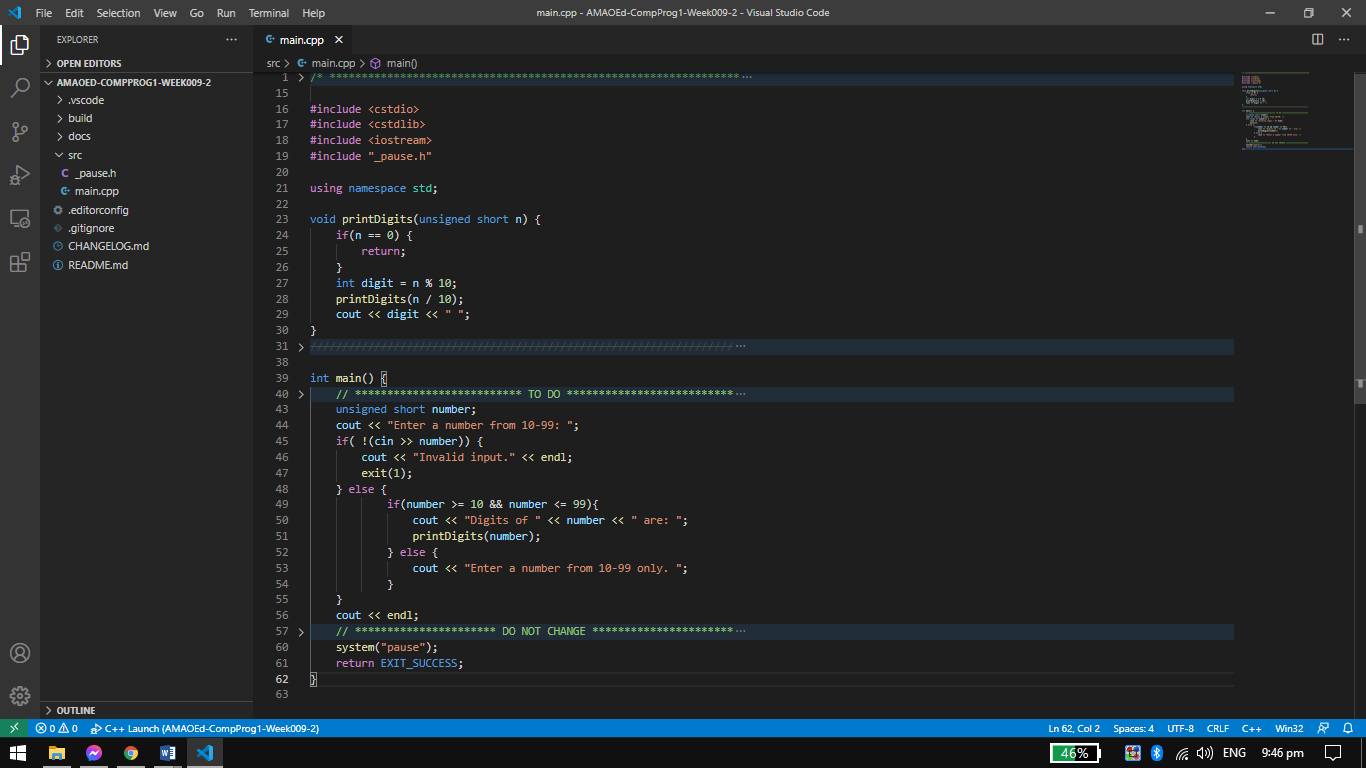
OUTPUT numbers[n] / numbers[n+1] = Undefined

END

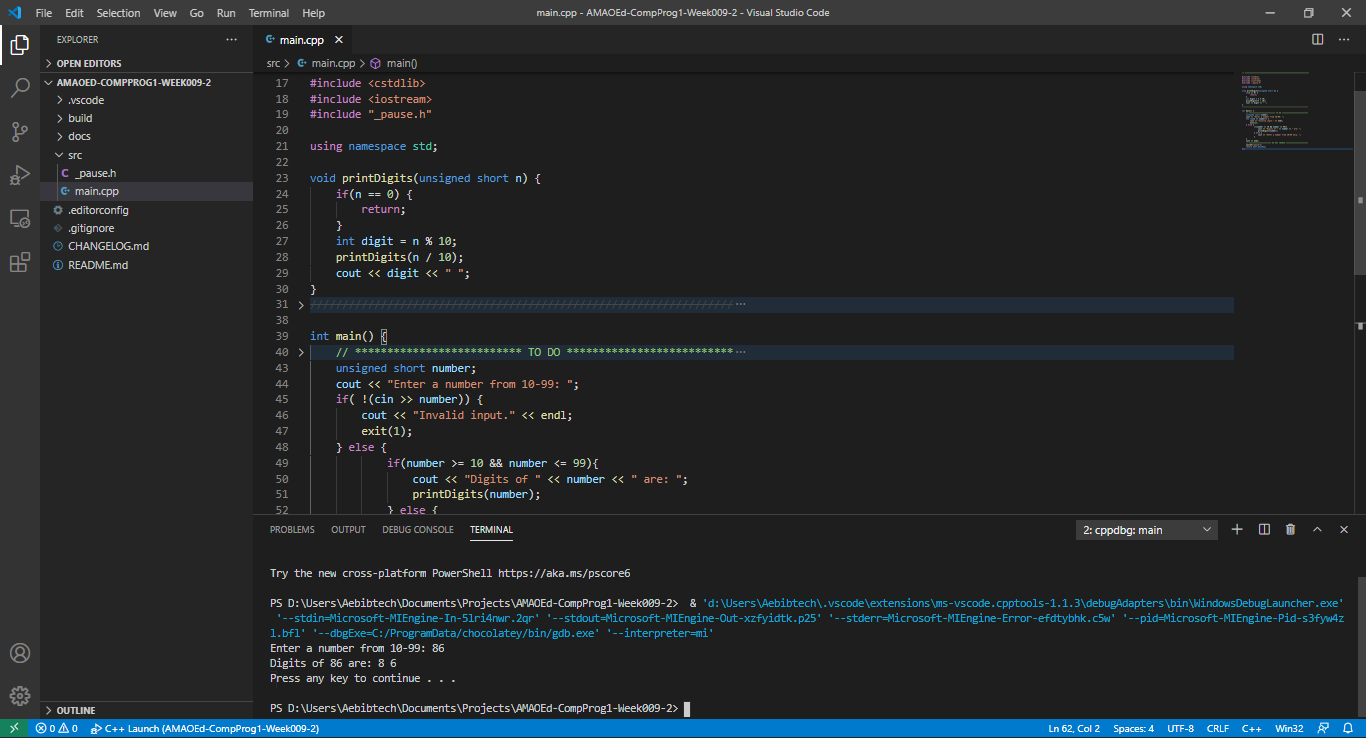
SET n AS n + 2

LOOP

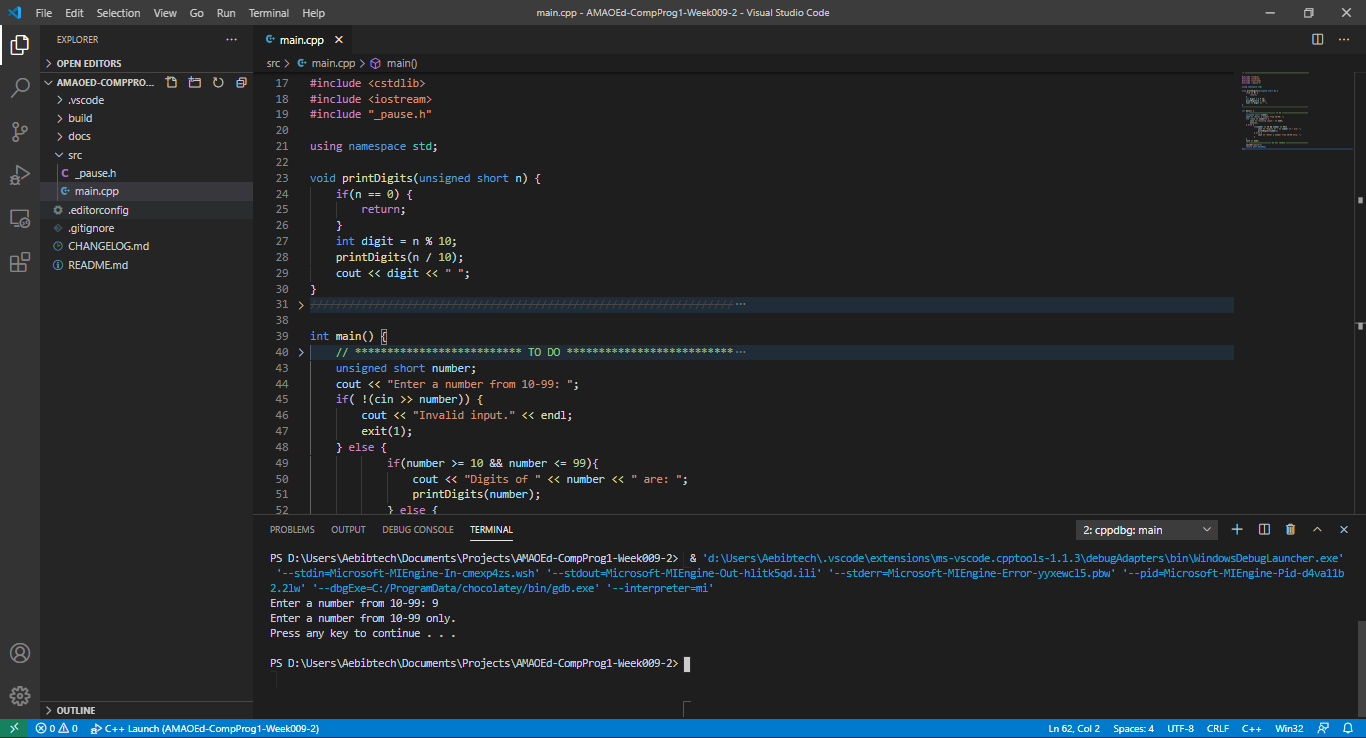
1. Write a program that will accept a short value from 10 to 99 and display them per digit (separated by a space).
   1. Code



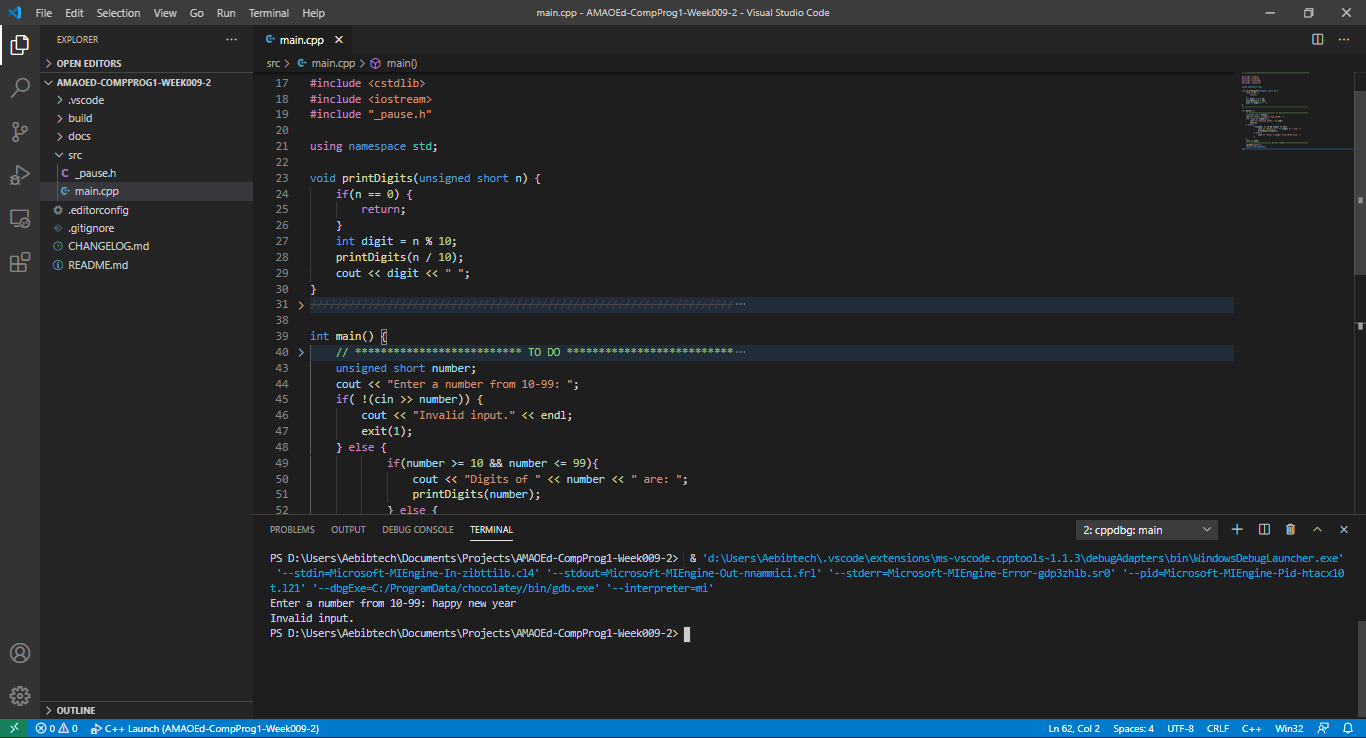
* 1. Sample Input 1



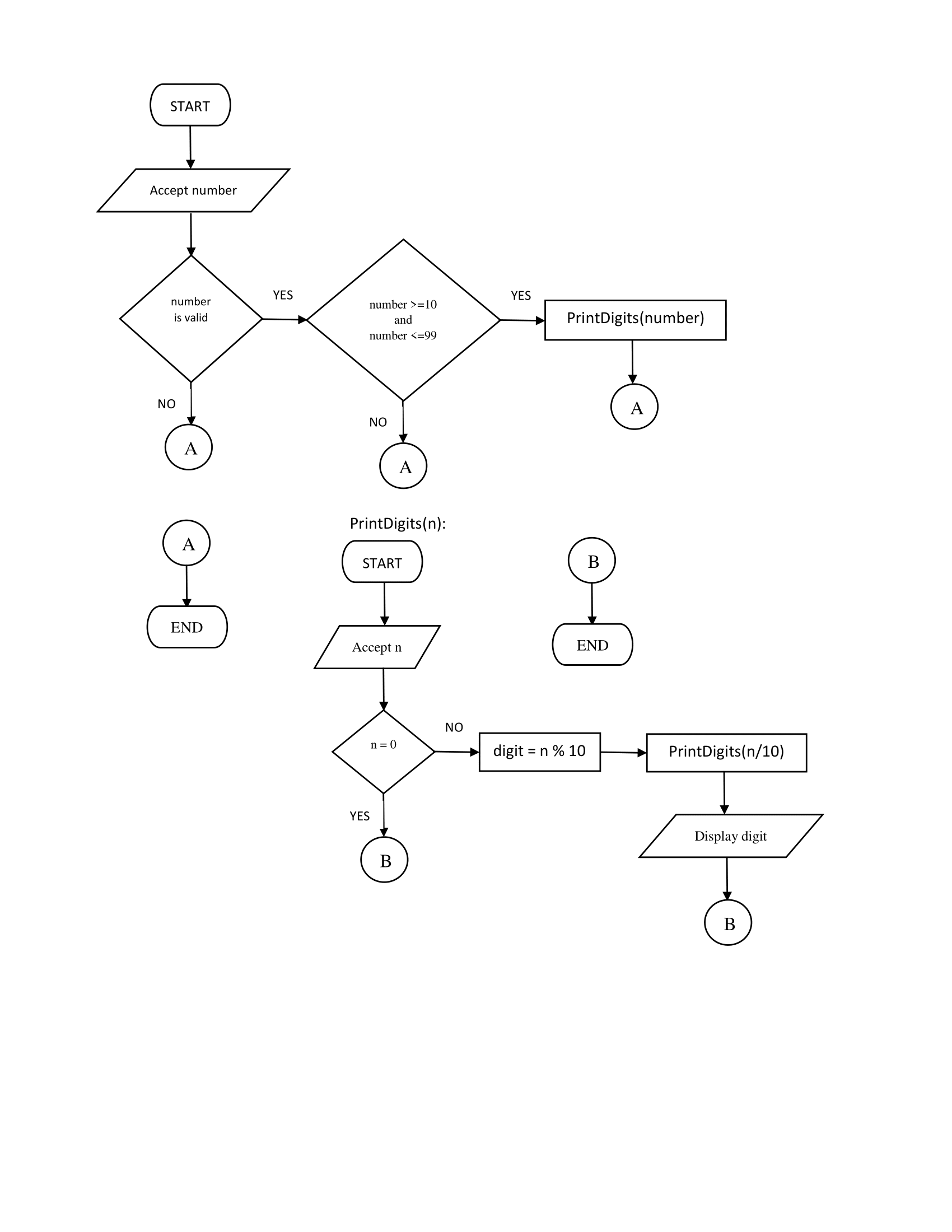
* 1. Sample Input 2



* 1. Sample Input 3



* 1. Flowchart



* 1. Pseudocode

DECLARE number

INPUT number

IF NotValid(number) THEN

OUTPUT “Invalid input.”

ELSE

IF number >=10 and number<=99 THEN

CALL PrintDigits(number)

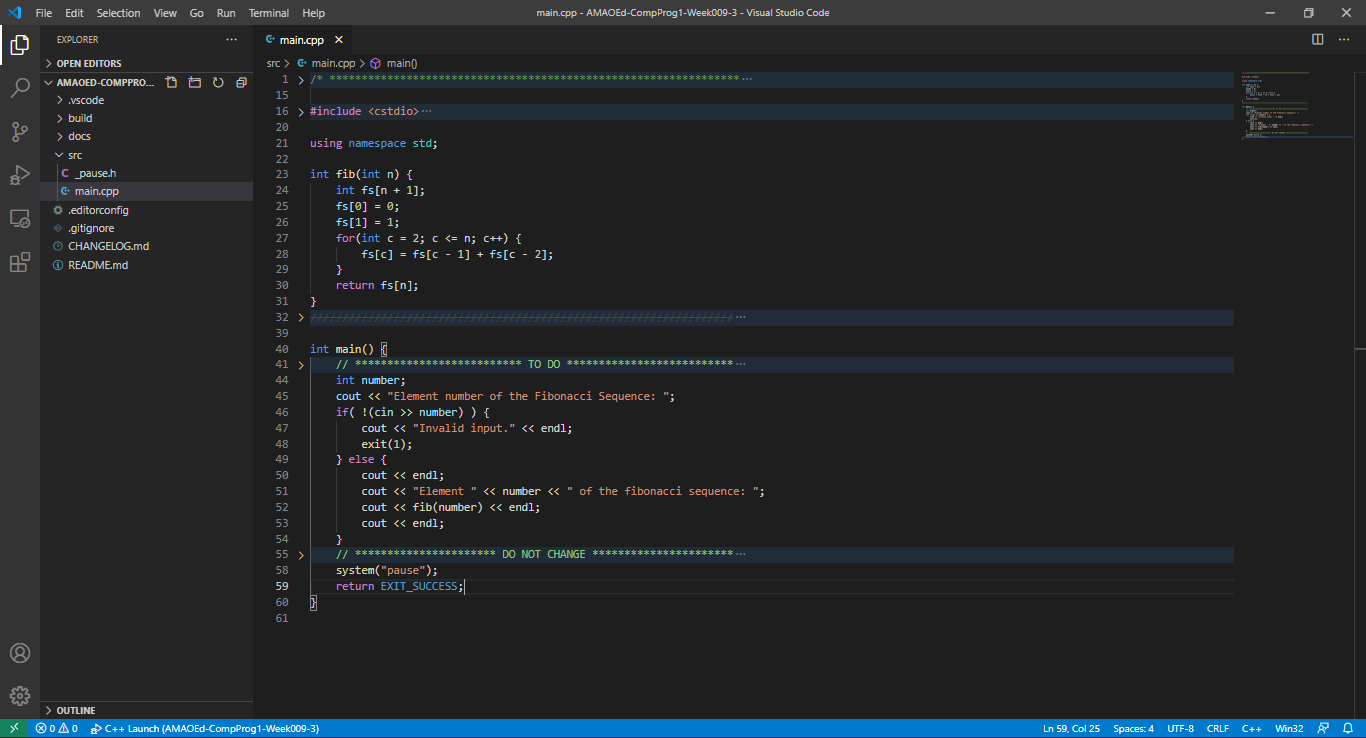
ELSE

OUTPUT “Enter a number from 10-99 only.”

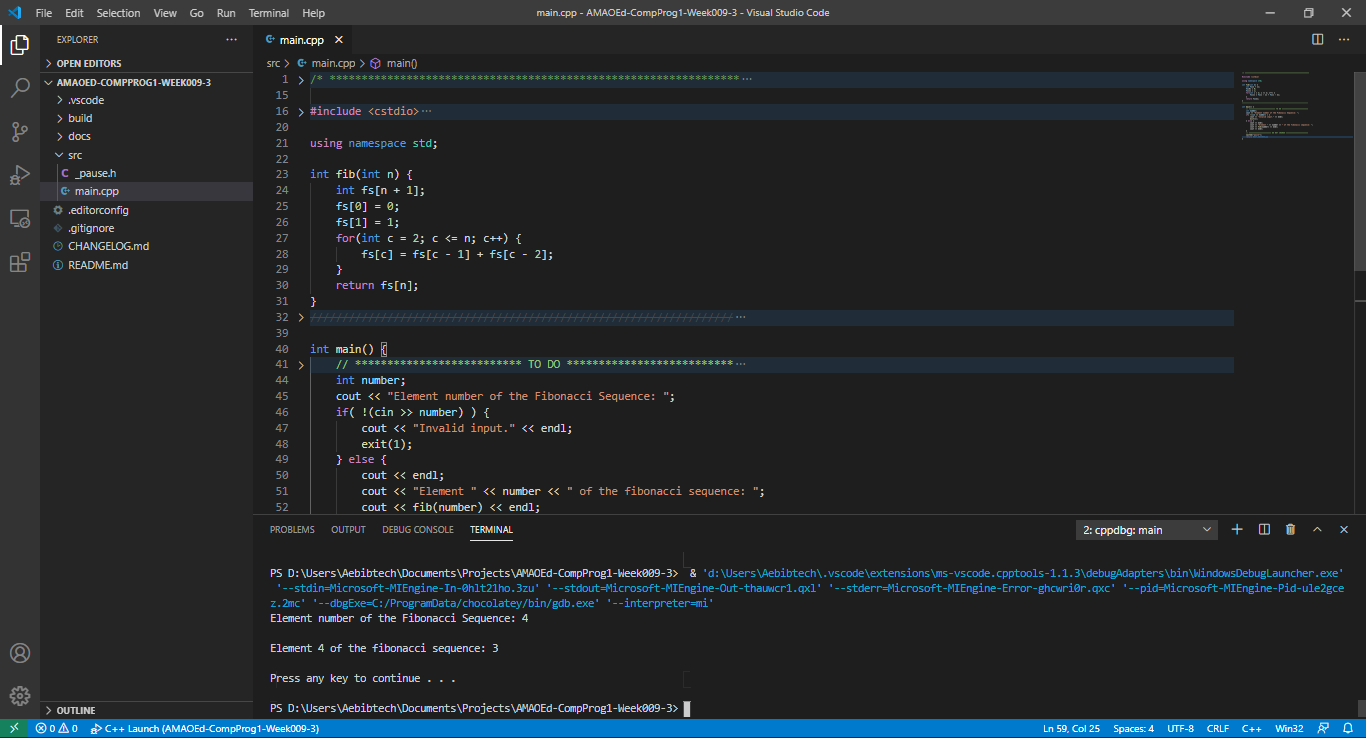
END

END

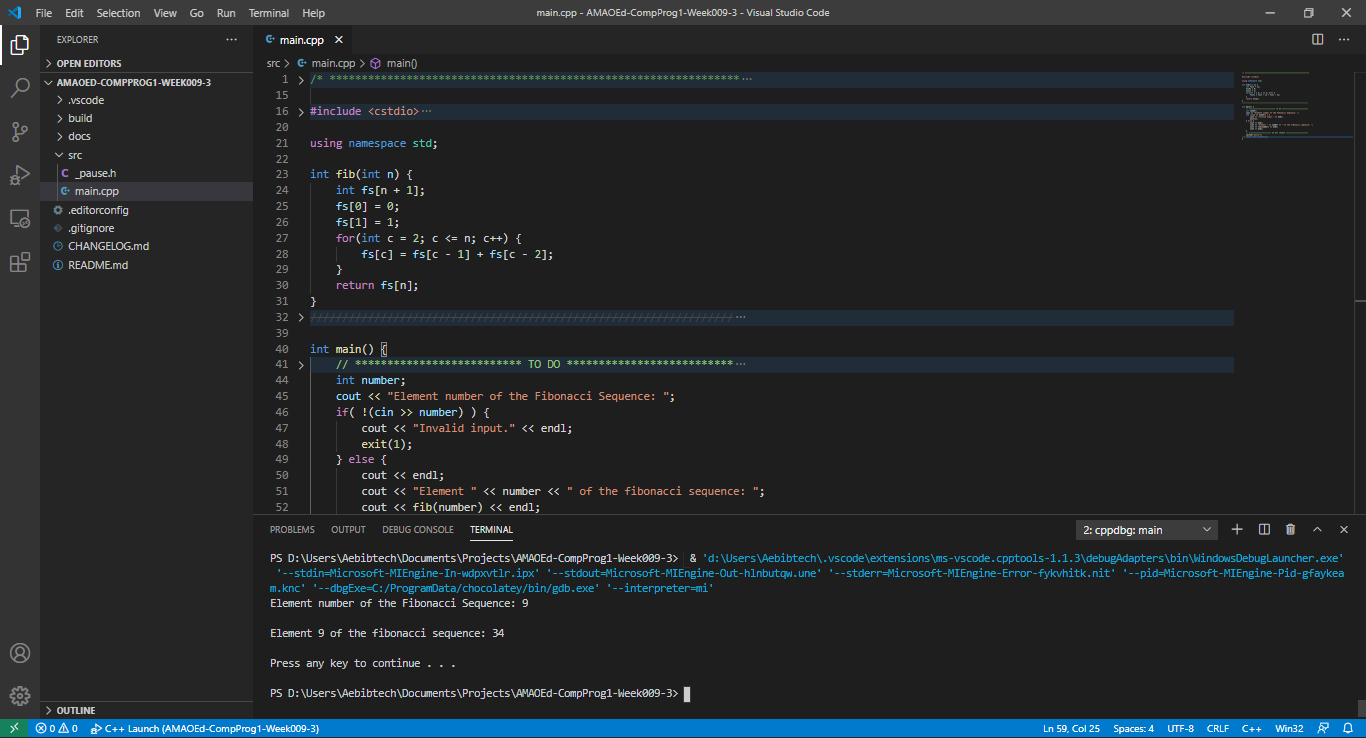
1. Write a program that will display the nth Fibonacci number. Create a function that will generate the nth Fibonacci number. Fibonacci numbers are numbers from the Fibonacci sequence which follows the pattern of 1, 1, 2, 3, 5, 8, 13, 21, 33, 54…
   1. Code



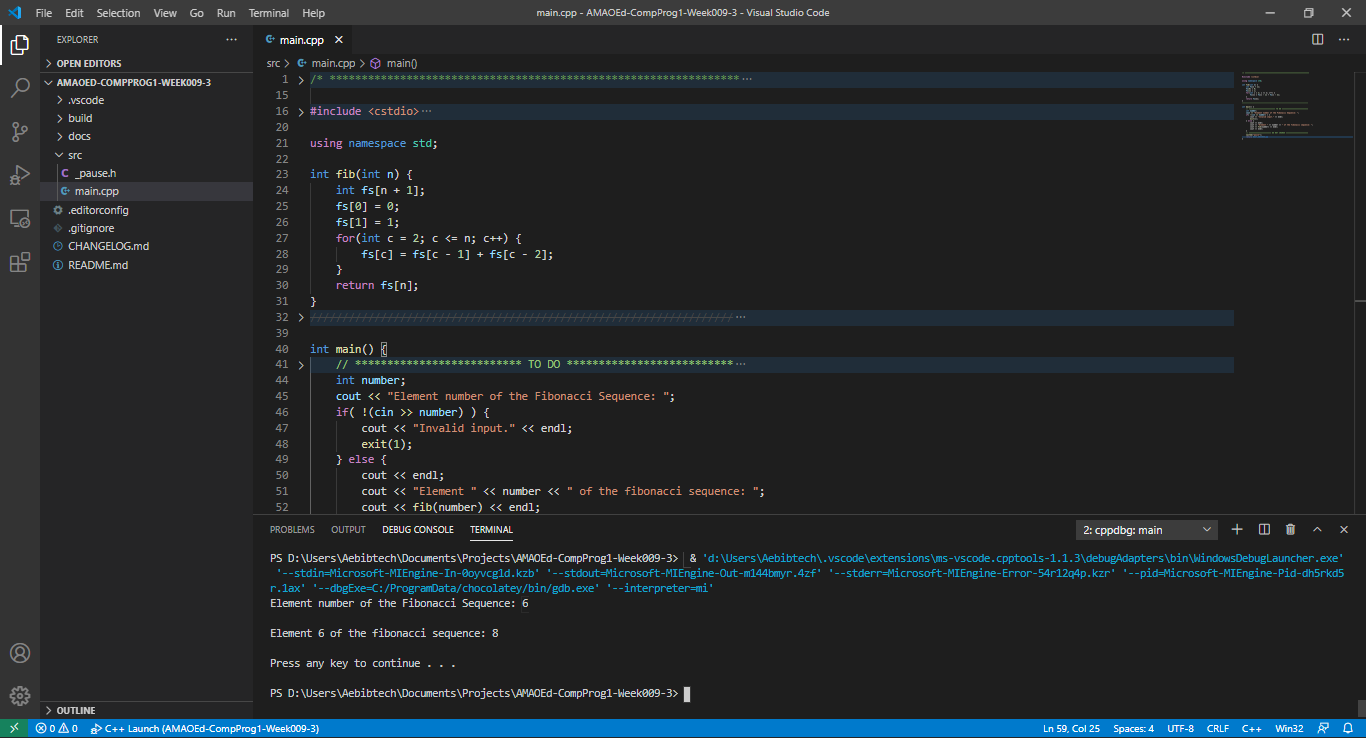
* 1. Sample Input 1



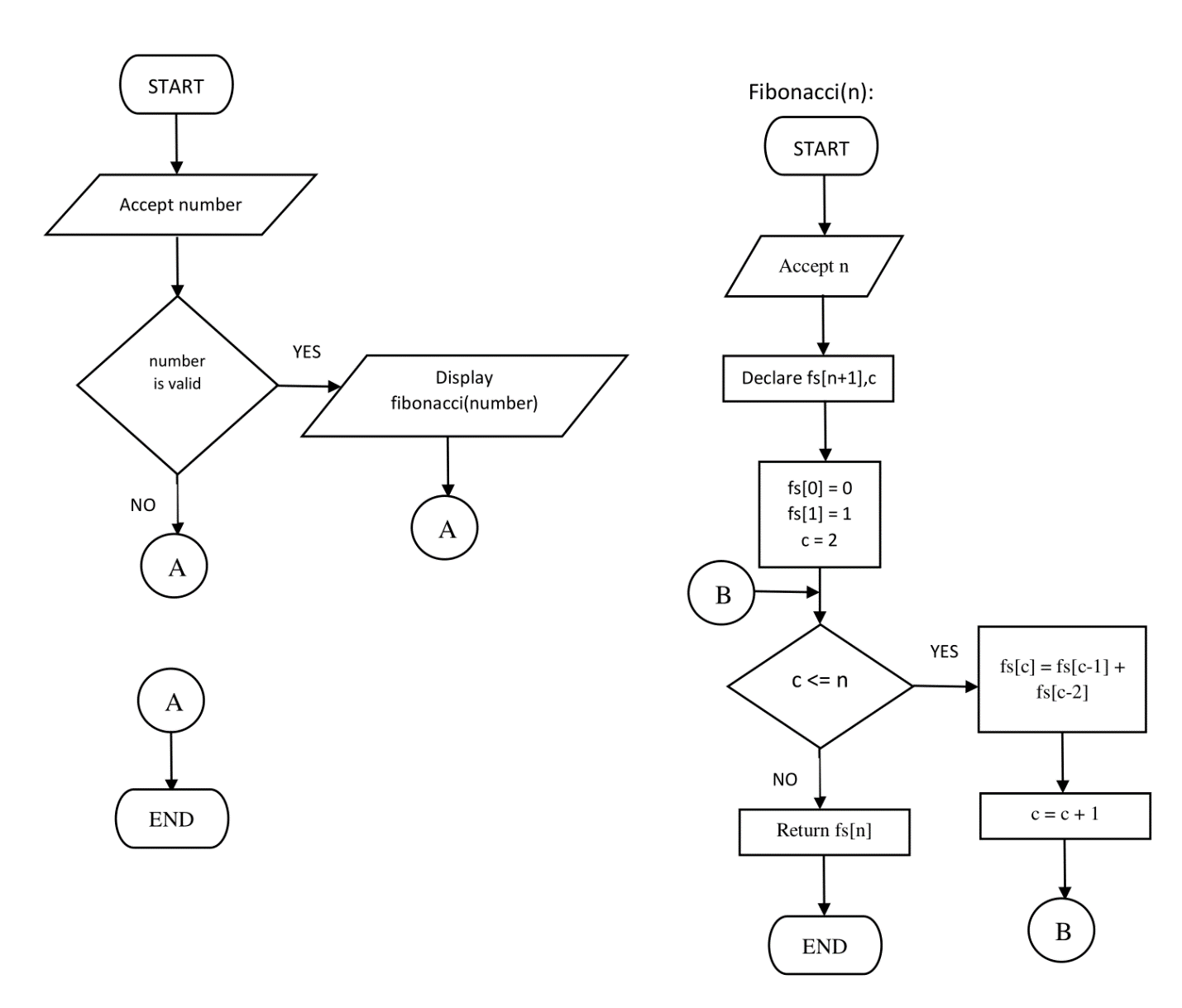
* 1. Sample Input 2



* 1. Sample Input 3



* 1. Flowchart



* 1. Pseudocode

DECLARE number

INPUT number

IF NotValid(number) THEN

OUTPUT “Invalid input.”

ELSE

OUTPUT fib(number)

END

1. What can you conclude from this activity?

I can conclude from this activity that functions are useful for organizing code blocks based on the code’s purpose and promotes cleaner coding. Also, some problems can be solved by recursive functions. Recursive functions are functions that call itself.