**Workshop 1**

**Subject**: PRF192

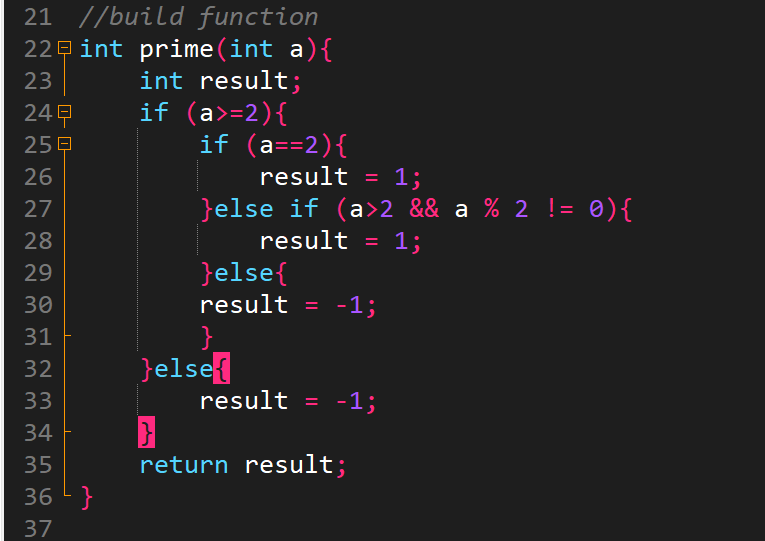
**Name**: Hoàng Thủy Nguyên

**ID**: DE191056

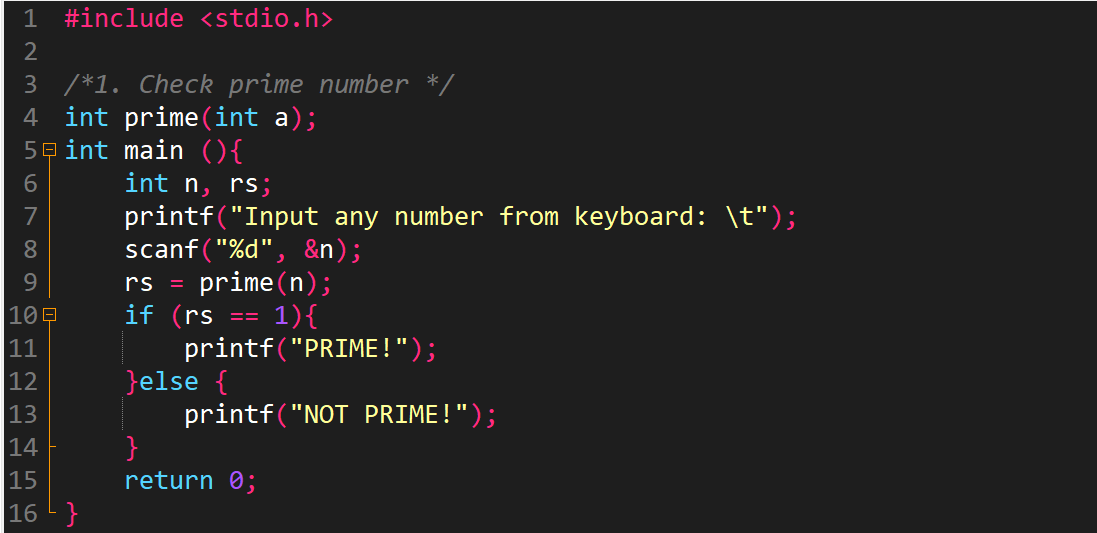
1. Write a function to check whether a number is prime. Then using this function to support printing out on screen the sequence of the primes from 2 to n where n is inputted from keyboard.

* **Write a function to check whether a number is prime**

1. The function is:

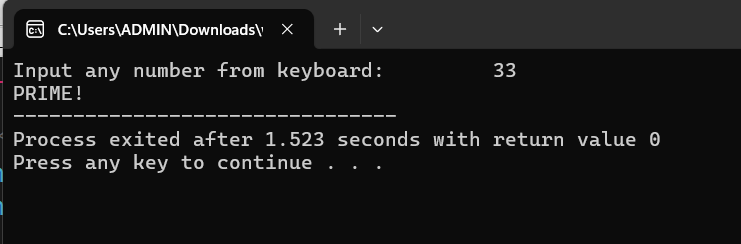


1. Call the function in main function:



1. Test:

* The first case:



Walkthrough:

Line 8: enter 33 🡪 n = 33

Line 9: function “prime” is called 🡪pass n to function “prime”

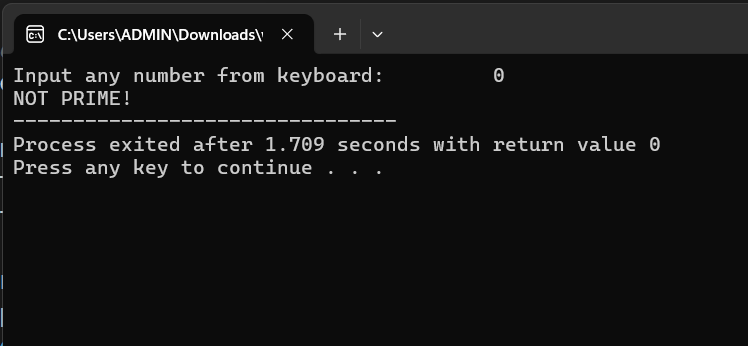
Line 18: a = 33

Line 24: result = 1 because 33 is prime

Line 9: result returned by “prime” function is assigned to “rs” variable

Line 11: because rs = 1 🡪print out PRIME

* The second case:



Walkthrough:

Line 8: enter 0 🡪 n = 0

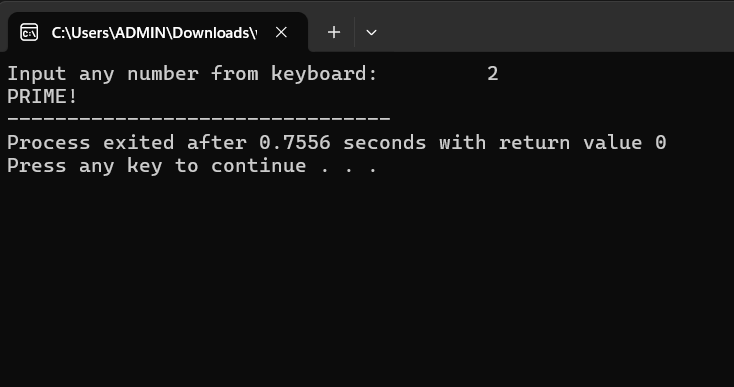
Line 9: function “prime” is called 🡪pass n to function “prime”

Line 18: a = 0

Line 24: result = -1 because 0 is NOT prime

Line 9: result returned by “prime” function is assigned to “rs” variable

Line 11: because rs = -1 🡪print out NOT PRIME

* The third case: 

Walkthrough:

Line 8: enter 2 🡪 n = 2

Line 9: function “prime” is called 🡪pass n to function “prime”

Line 18: a = 2

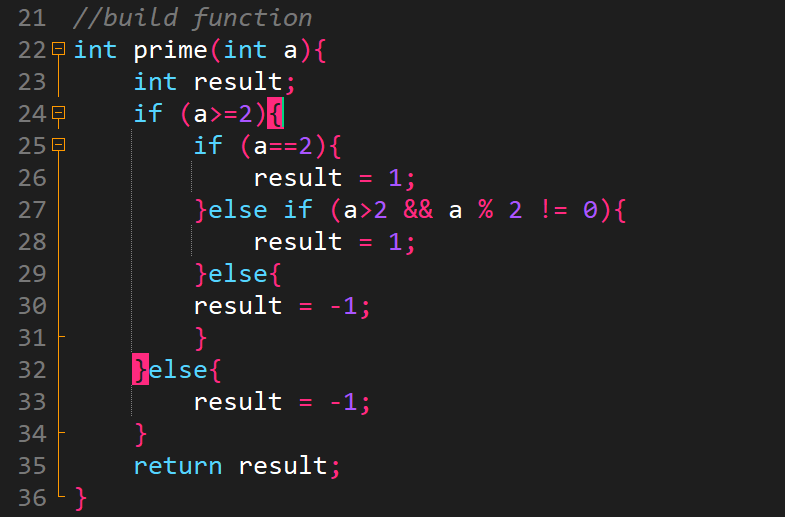
Line 24: result = 1 because 0 is prime

Line 9: result returned by “prime” function is assigned to “rs” variable

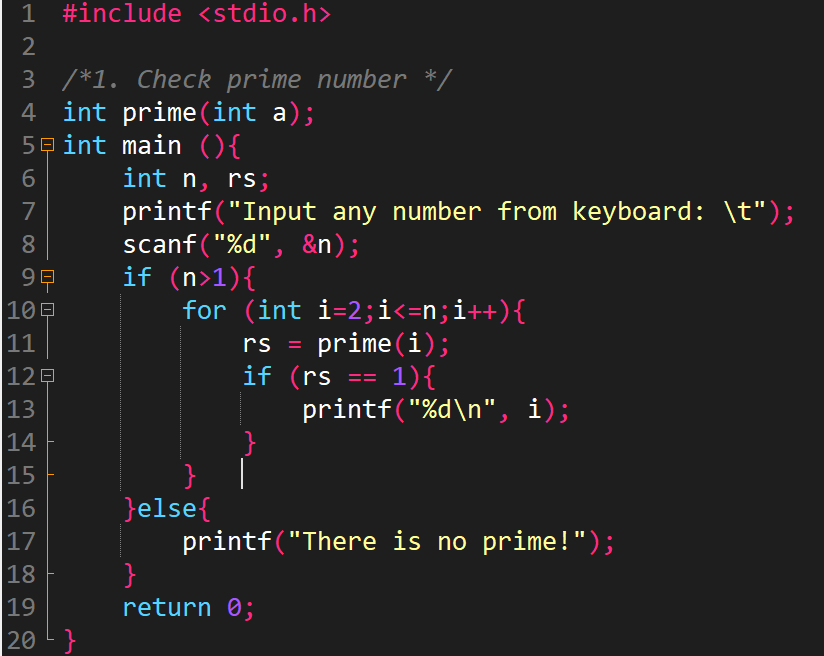
Line 11: because rs = 1 🡪print out PRIME

* **Then using this function to support printing out on screen the sequence of the primes from 2 to n where n is inputted from keyboard.**

1. The function is:

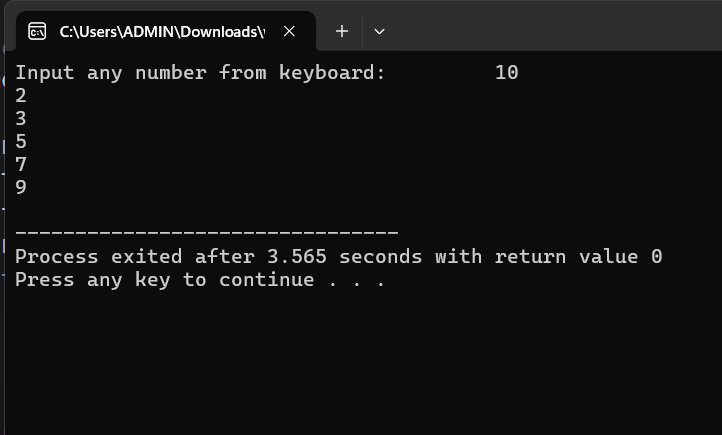


1. Call the function in main function:



1. Test:

* The first case:



Walkthrough:

Line 8: enter 10 🡪 n=10

Line 9: because n=10>1 is true 🡪execute condition command

Line 10: start with i=2( because 0 and 1 is not prime), pass i to “prime” function

Line 22: a = 2

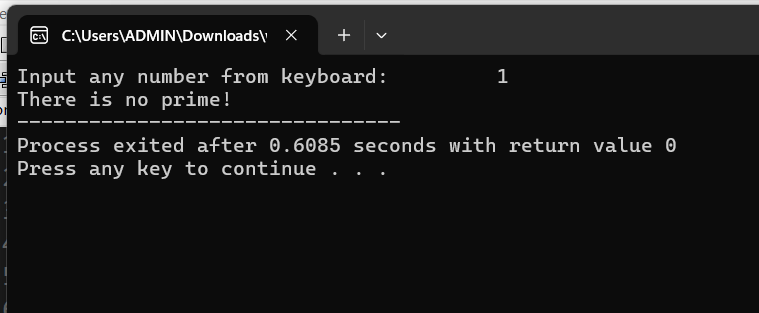
Line 26: result = 1(because 2 is prime)

Line 11: result returned by “prime” function is assigned to “rs” variable

Line 13: print out 2 on screen because 2 is prime and increase i by one unit (i++)

Line 10: now i=3 and loop until i=n

* The second case:



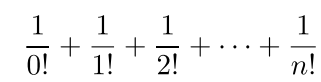
Walkthrough:

Line 8: enter 1 🡪 n=1

Line 9: because n=1>1 is false 🡪execute else command

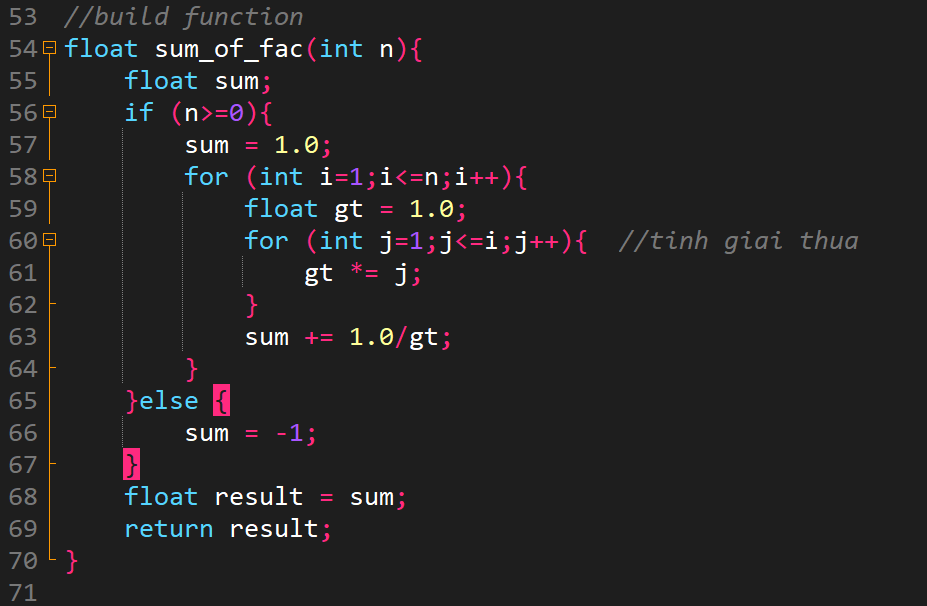
Line 17: print out “There is no prime”

1. Write a function to calculate

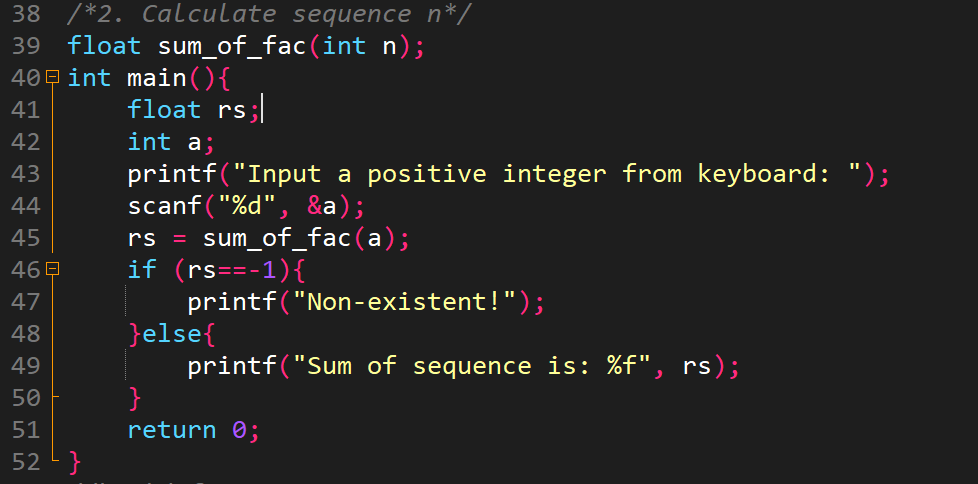


where n is inputted from keyboard.

1. The function is:

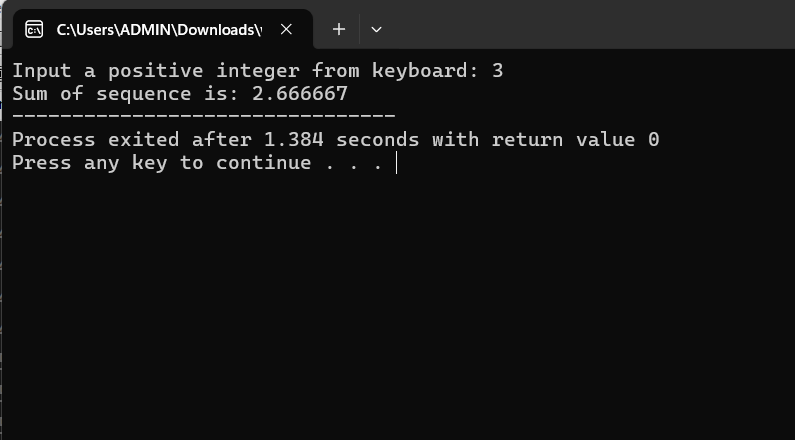


1. Call the function in main function:



1. Test:

* The first case:



Walkthrough:

Line 44: enter 3 from keyboard 🡪 a=3

Line 54: pass a to n of “sum\_of\_fac” function🡪n=3

Line 56: because n=3>0 is true🡪 condition command is executed

Line 57: assign 1 to ‘sum’

Line 58: initiate loop to calculate “sum”

Line 60: initiate loop to calculate “gt”(factorial).

</>Explain loop in loop: First, i =1 🡪 j=1 🡪 gt = 1 = 1!

i=2 🡪 j=1 🡪 gt=1

j=2 🡪 gt =1x2 = 2!

i=3 🡪 j=1 🡪 gt =1

j=2 🡪 gt=1x2

j=3 🡪 gt=1x2x3 =3!

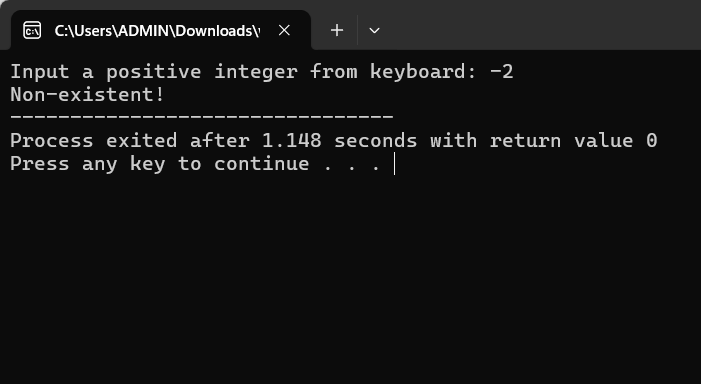
…

Line 63: after calculate “gt”🡪 calculate “sum”

Line 69: : result returned by “sum\_of\_fac” function is assigned to “rs” variable

Line 49: print out “rs” on screen

* The second case:



Walkthrough:

Line 44: enter -2 from keyboard 🡪a=-2

Line 54: pass a to n of “sum\_of\_fac” function🡪n=-2

Line 56: n=-2 >=0 is false 🡪Line 65

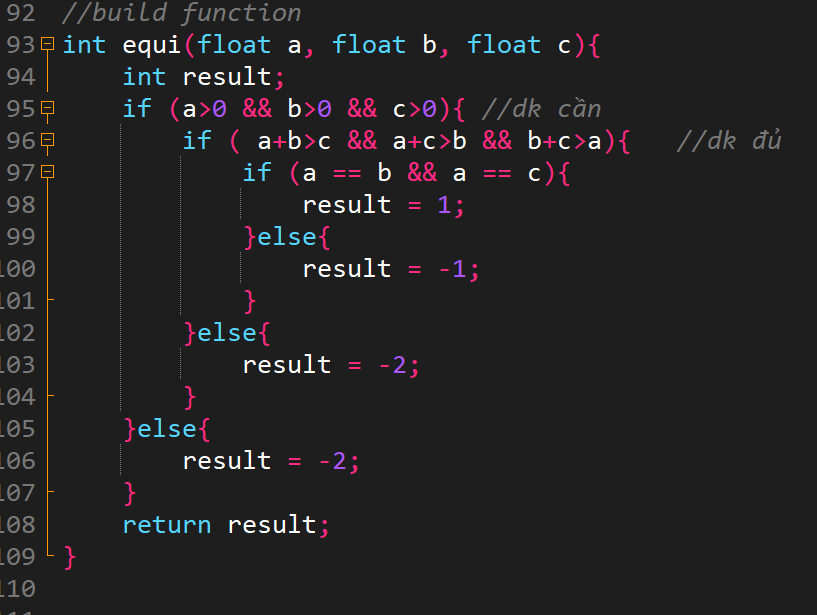
Line 69: : result returned by “sum\_of\_fac” function is assigned to “rs” variable

Line 46: because “rs = -1”🡪 condition command is executed🡪print out screen “Non-existent”

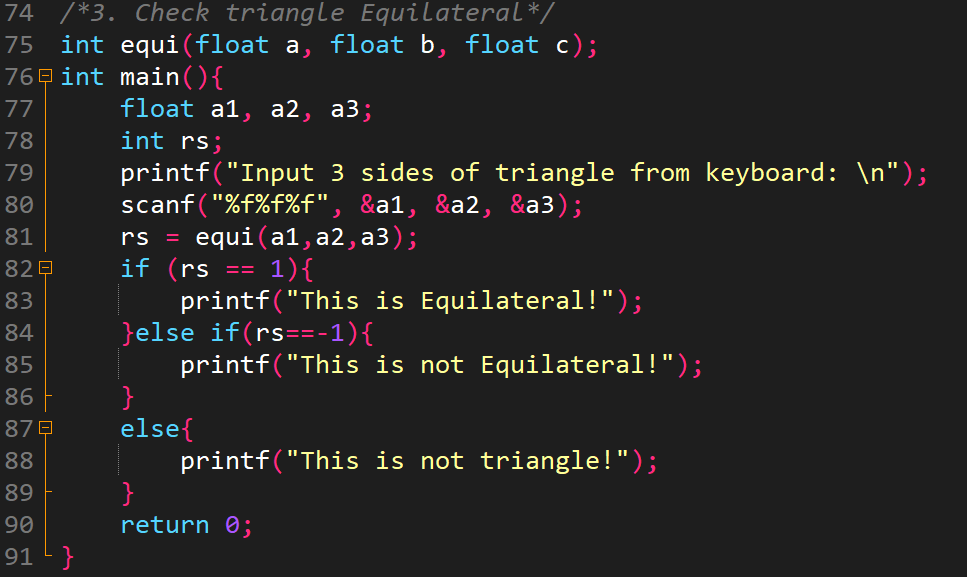
1. Write a function to check whether a triangle is Equilateral.

Please test your program where walkthrough should be done along the lines of code.

1. The function is:

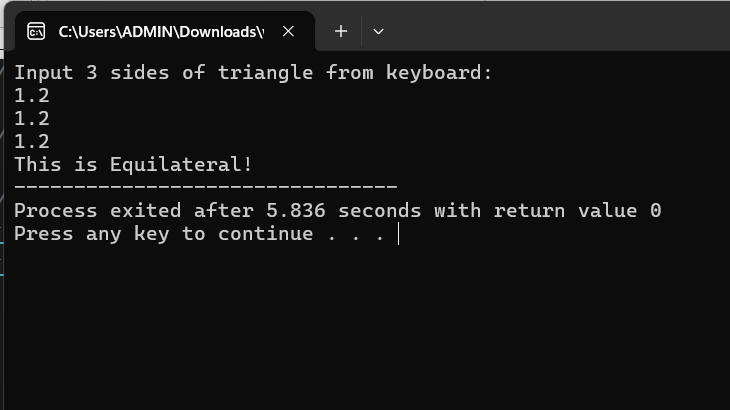


1. Call the function in main function:



1. Test:

* The firtst case:



Walkthrough:

Line 80: enter 1.2 1.2 1.2 from keyboard 🡪 a1=1.2 a2=1.2 a3=1.2

Line 93: pass a1, a2, a3 to a, b, c of “equi” function🡪a=1.2 b=1.2 c=1.2

Line 95: a, b, c =1.2>0 is true🡪condition command is executed

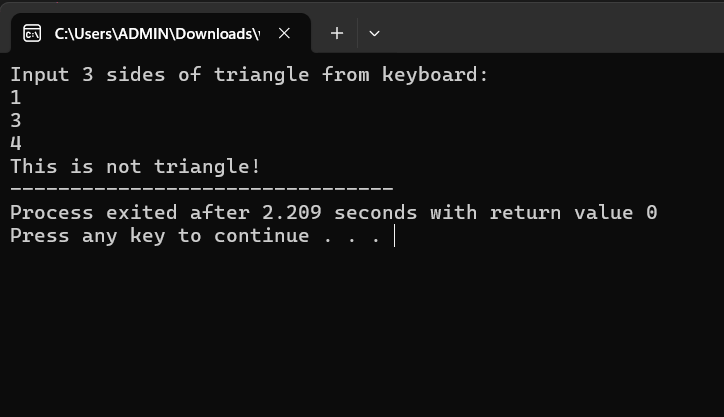
Line 96: this condition is true 🡪condition command is executed

Line 97: this condition is true🡪 condition command is executed 🡪assign 1 to “result” variable

Line 108: result returned by “equi” function is assigned to “rs” variable

Line 82: because rs=1🡪 print out on screen “This is Equilateral”

* The second case:



Walkthrough:

Line 80: enter 1 3 4 from keyboard 🡪 a1=1 a2=3 a3=4

Line 93: pass a1, a2, a3 to a, b, c of “equi” function🡪a=1 b=3 c=4

Line 95: a, b, c >0(1>0, 3>0, 4>0) is true🡪condition command is executed

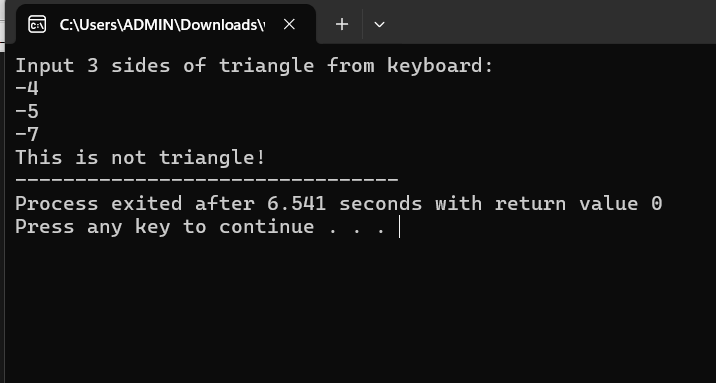
Line 96: this condition is false (1+3>4 is false) 🡪line 102

Line 102: assign -2 to “result” variable

Line 108: result returned by “equi” function is assigned to “rs” variable

Line 84: because rs= -2 🡪 print out on screen “This is not triangle”

* The third case:



Walkthrough:

Line 80: enter -4 -5 -7 from keyboard 🡪 a1=-4 a2=-5 a3=-7

Line 93: pass a1, a2, a3 to a, b, c of “equi” function🡪a=-4 b=-5 c=-7

Line 95: this condition is false (-4>0 is false)🡪Line 105

Line 105: assign -2 to “result” variable

Line 108: result returned by “equi” function is assigned to “rs” variable

Line 87: because rs=-2🡪 print out on screen “This is not triangle”