

Student name : Viraj Patel  
Student ID ; 201351707

## Short report

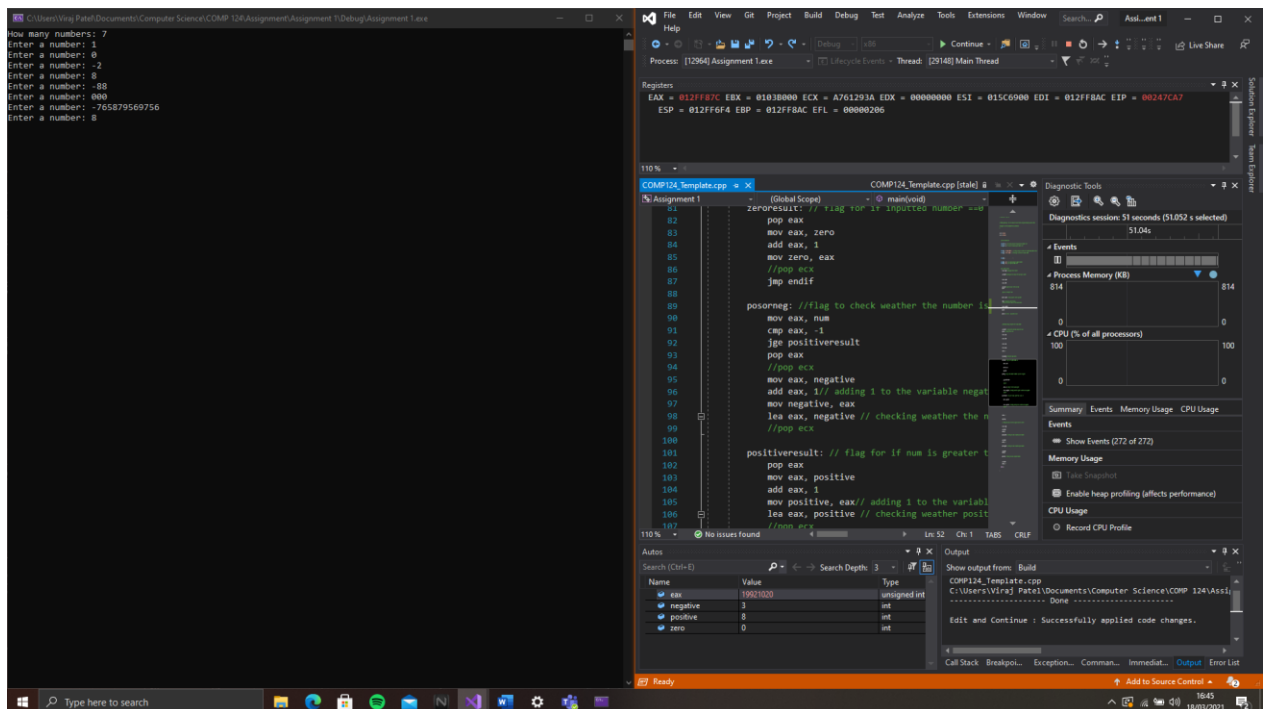
I created 5 integer variables to store the Number of **Positive** , **Negative**, and **Zero** inputs from the user. The **count** variable was created to store the number of inputs the user decided to enter. This **count** variable was used as a counter for the loop. A variety of String variables were also created, this was used for outputting the calculated values.

The call **printf** function was used to print the message asking the use to input a number, this number was used to determine amount of times the loop was iterated through. The inputted number was then saved as **count**, and then stored on the **ecx** register.

I had encountered significant issues when attempting to get the loop to terminate. I have placed the **pop ecx** function in its relevant place in order for ECX to decrement buy 1 after each iteration , however this did not work , and the loop does not terminate. Inside the loop the user number input is taken and stored as num. to check weather the input was a Zero , on line 74 the **jnz** (jump if not zero 0 was used, if the value was 0 the jump would not occur and the Zero count would be appended buy 1.

If the **num** value was anything other than 0 then the code would jump to the **posorneg** flag. This is where I checked to see if **num** was a positive or negative. The **cmp** (compare) function was used, along with the **jge** (jump if greater than) , if **num** was greater than -1 then the code would jump to the flag where the Positive variable will be appended by 1. If **num** was less then or equals to -1 than the Negative variable would be appended buy 1.

As I could not get my loop to terminate, the code where I output the calculated results are not displayed, however they do work , and after **line 111** it is visible that I have outputted the requested numbers. The commented out **pop ecx** function I have implemented is evidence that I have attempted to amend the bug in my code.



```
117: How many numbers: 7
118: Enter a number: 1
119: Enter a number: 0
120: Enter a number: -2
121: Enter a number: 8
122: Enter a number: -88
123: Enter a number: 888
124: Enter a number: -765879569756
125: Enter a number: 8
```

```
82: zerosresult: //flag top if inputted NUMBER == 0
83: pop eax
84: mov eax, zero
85: add eax, 1
86: mov zero, eax
87: //pop ecx
88: jmp endif
89:
90: posorneg: //flag to check weather the number is
91: mov eax, num
92: cmp eax, -1
93: jge positivresult
94: pop eax
95: //pop ecx
96: mov eax, negative
97: add eax, 1 // adding 1 to the variable negat
98: mov negative, eax // checking weather the n
99: lea eax, negative // checking weather posit
100: //pop ecx
101:
102: positivresult: // flag for if num is greater t
103: pop eax
104: mov eax, positive
105: add eax, 1
106: mov positive, eax // adding 1 to the variab
107: lea eax, positive // checking weather posit
108: //pop ecx
```

Name	Value	Type
count	765879569756	unsigned int
negative	3	int
positive	8	int
zero	0	int