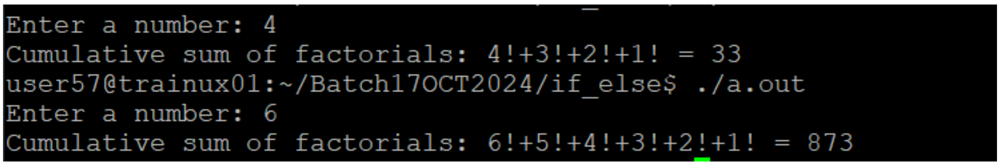
For While Do while Break Continue Assignment

1.WAP to read a number n and to display the cumulative sum of factorial of all numbers upto n . (use for or while)
Input: 4
Ouput: 4!+3!+2!+1! = 32

Sol:

```
#include <stdio.h>
int factorial(int num) {
   int fact = 1;
   for (int i = 1; i <= num; i++) {
      fact *= i;
   }
   return fact;
}


int main() {
   int n, cumulative_sum = 0;
   printf("Enter a number: ");
   scanf("%d", &n);


   printf("Cumulative sum of factorials: ");
   for (int i = n; i >= 1; i--) {
      cumulative_sum += factorial(i);
      printf("%d!", i);
      if (i > 1) {
         printf("+");
      }
   }
   printf(" = %d\n", cumulative_sum);
   return 0;
}
```

Output:



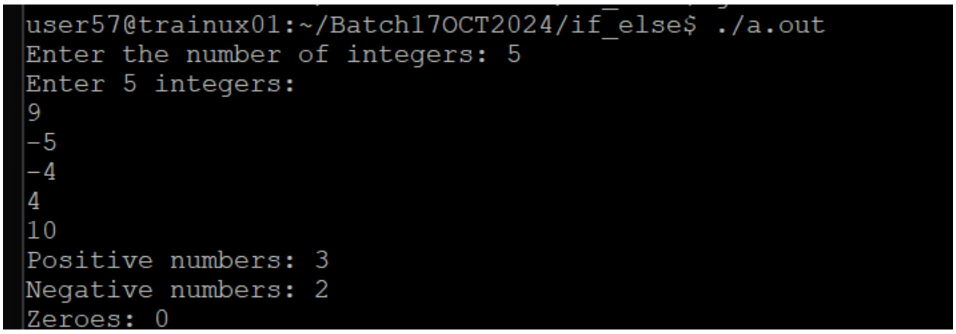
```
Enter a number: 4
Cumulative sum of factorials: 4!+3!+2!+1! = 33
user57@trainux01:~/Batch17OCT2024/if_else$ ./a.out
Enter a number: 6
Cumulative sum of factorials: 6!+5!+4!+3!+2!+1! = 873
```

2. Write a program to accept “N” integers from the user. “N” also has to be taken from the user. Take the count of +ve numbers, -ve numbers and 0’s.
However the program should not accept a non-integer value. If a non-integer value is entered, used must be asked to re-enter.
[Hint:
a. Use the return value of scanf to find out whether the user has entered integer or not.
b. You also will have to clear the input buffer before taking the next input.
For clearing the input buffer, use one of the following approaches
➢ while (getchar() != '\n'); // keep reading till newline and discard the characters
➢ scanf(“%*s”); // read and discard one string

Sol:

```
#include <stdio.h>

int main() {

  int N;

  int pos_count = 0, neg_count = 0, zero_count = 0;

  int input;

  int ret;


  printf("Enter the number of integers: ");

  while (1) {

    ret = scanf("%d", &N);


    if (ret == 1 && N > 0) {

      break;

    } else {

      printf("Invalid input. Please enter a positive integer for N: ");

      while (getchar() != '\n');
}
    }

    printf("Enter %d integers:\n", N);
 for (int i = 0; i < N; i++) {
        ret = scanf("%d", &input) ;
        while (ret != 1) {
            printf("Invalid input. Please enter an integer: ");
            while (getchar() != '\n');
            ret = scanf("%d", &input);
      }

        if (input > 0) {
            pos_count++;
        } else if (input < 0) {
            neg_count++;
        } else {
            zero_count++;
        }
    }
    printf("Positive numbers: %d\n", pos_count);
    printf("Negative numbers: %d\n", neg_count);
    printf("Zeroes: %d\n", zero_count);

    return 0;
}
```

Output:



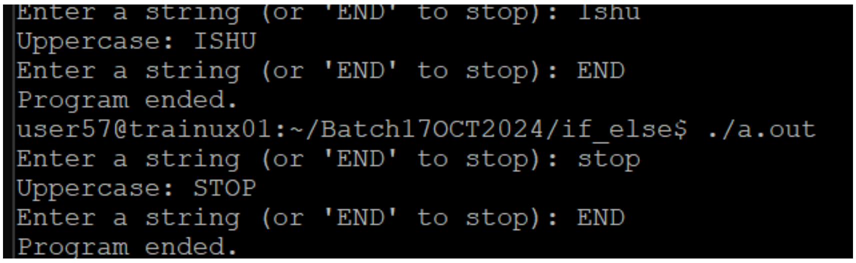
```
user57@trainux01:~/Batch17OCT2024/if_else$ ./a.out
Enter the number of integers: 5
Enter 5 integers:
9
-5
-4
4
10
Positive numbers: 3
Negative numbers: 2
Zeroes: 0
```

3. Write a program to continuously read a string of maximum length 80 chars, End the program if string is END, else convert to upper case, display and continue. (use while).

Sol:

```
#include <stdio.h>

#include <string.h>

int main() {

    char str[80];


    while (1) {

        printf("Enter a string (or 'END' to stop): ");

        fgets(str, sizeof(str), stdin);

        str[strcspn(str, "\n")] = '\0';


        if (strcmp(str, "END") == 0) {

            break;

        }

        for (int i = 0; str[i] != '\0'; i++) {

            if (str[i] >= 'a' && str[i] <= 'z') {

                str[i] = str[i] - 'a' + 'A';

            }

        }

        printf("Uppercase: %s\n", str);

    }
    printf("Program ended.\n");


    return 0;
}
```

Output:



```
Enter a string (or 'END' to stop): Ishu
Uppercase: ISHU
Enter a string (or 'END' to stop): END
Program ended.
user57@trainux01:~/Batch17OCT2024/if_else$ ./a.out
Enter a string (or 'END' to stop): stop
Uppercase: STOP
Enter a string (or 'END' to stop): END
Program ended.
```

4. Refer the program “value_out_of_domain.c”. Try to run the program with a large value say 255. Check the output? Is it correct? Fix the issue observed.
What improvements do you suggest?

Sol:

```
#include <stdio.h>

int main() {

    long long fact = 1;

    int count, num;


    printf("Enter a number: ");

    scanf("%d", &num);


    if (num < 0) {

        printf("Factorial is not defined for negative numbers.\n");

        return 1;

    }


    for (count = 1; count <= num; count++) {

        fact = fact * count;

    }
    printf("Factorial of %d is: %lld\n", num, fact);

    return 0;

}
```

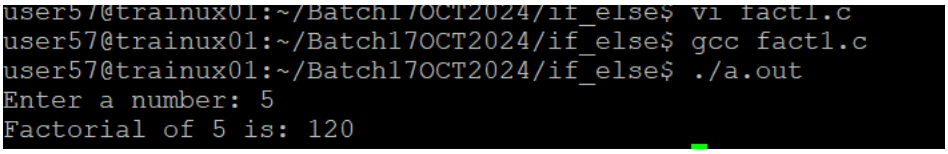
-> Changed int fact to long long fact:

This increases the range of the factorial calculation, but it still has a limit. For very large numbers (like 255!), you would still need to handle larger numbers using a different approach.->

-> Negative Input Check:

Factorials are not defined for negative numbers, so I added a check to handle negative inputs and print an appropriate message

Output:



```
user57@trainux01:~/Batch17OCT2024/if_else$ vi fact1.c
user57@trainux01:~/Batch17OCT2024/if_else$ gcc fact1.c
user57@trainux01:~/Batch17OCT2024/if_else$ ./a.out
Enter a number: 5
Factorial of 5 is: 120
```

5. Refer the code below. It does not output anything. Fix it.

```
#include <stdio.h>
int main()
{
int x = 5;
while (x > 0);
{
printf( "Value of x :%d \n", x);
x--;
}
return 0;
```

Sol:

```
#include <stdio.h>

int main() {

    int x = 5;

    while (x > 0) {

        printf("Value of x: %d \n", x);

        x--;

    }

return 0;

}
```

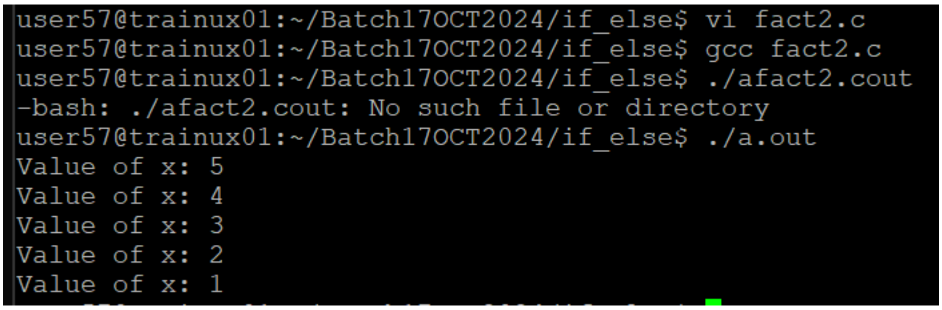
6. Analyse the code, identify the issues

```
#include <stdio.h>
int main()
{
float cnt = 0, num = 1000;
do

{
printf ("\n%d\n%d", num,cnt);
num /= cnt;
} while (cnt --); /* End of while

return 0;

}
```

Sol :

```
#include <stdio.h>

int main() {

   int x = 5;

   while (x > 0) {

      printf("Value of x: %d \n", x);

      x--;

   }

   return 0;

}
```

Output:



```
user57@trainux01:~/Batch17OCT2024/if_else$ vi fact2.c
user57@trainux01:~/Batch17OCT2024/if_else$ gcc fact2.c
user57@trainux01:~/Batch17OCT2024/if_else$ ./afact2.cout
-bash: ./afact2.cout: No such file or directory
user57@trainux01:~/Batch17OCT2024/if_else$ ./a.out
Value of x: 5
Value of x: 4
Value of x: 3
Value of x: 2
Value of x: 1
```