NITHIN S
221IT085

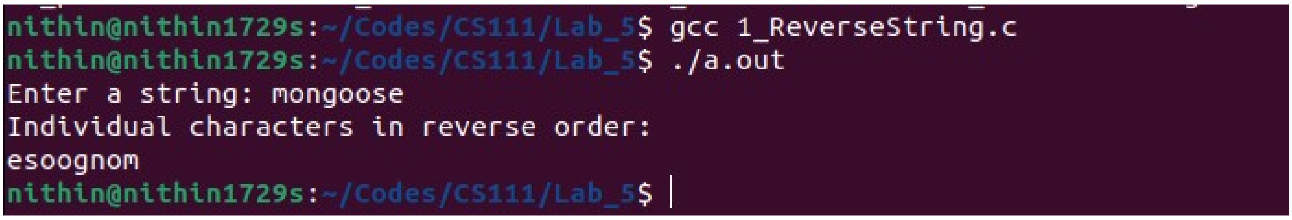
CS111 Lab Assignment 6

STRING

Q1) Write a C program to print individual characters of a string in reverse order

```
#include <stdio.h>
#include <string.h>

#define MAX_LENGTH 100

int main() {
    char str[MAX_LENGTH];
    int length, i;

    printf("Enter a string: ");
    scanf("%99[^\n]", str);

    length = strlen(str);

    printf("Individual characters in reverse order: \n");
    for (i = length - 1; i >= 0; i--) {
        printf("%c", str[i]);
    }
    printf("\n");

    return 0;
}
```

OUTPUT

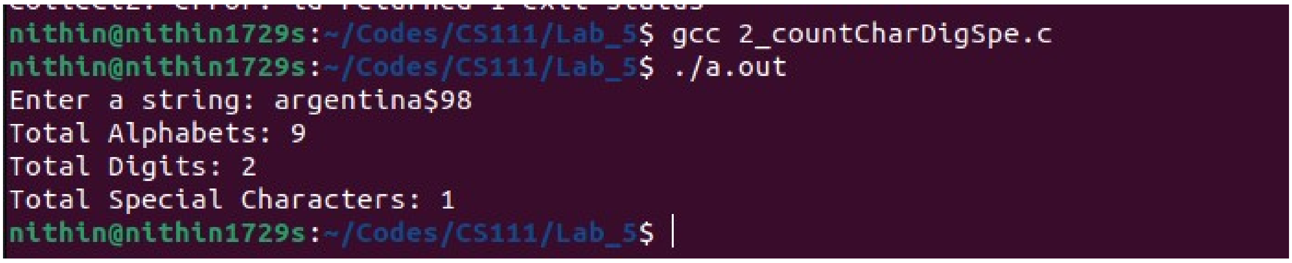


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 1_ReverseString.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: mongoose
Individual characters in reverse order:
esoognom
nithin@nithin1729s:~/Codes/CS111/Lab_5$
```

Q2) Write a C program to count total number of alphabets, digits and special characters in a string.

```
#include <stdio.h>
#include <ctype.h>

#define MAX_LENGTH 100

int main() {
    char str[MAX_LENGTH];
    int alphabets = 0, digits = 0, special_chars = 0,
i;

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    for (i = 0; str[i] != '\0'; i++) {
        if (isalpha(str[i])) {
            alphabets++;
        } else if (isdigit(str[i])) {
            digits++;
        } else if (str[i] != ' ' && str[i] != '\n' &&
str[i] != '\t') {
            special_chars++;
        }
    }
    printf("Total Alphabets: %d\n", alphabets);
    printf("Total Digits: %d\n", digits);
    printf("Total Special Characters: %d\n",
special_chars);

    return 0;
}
```

OUTPUT

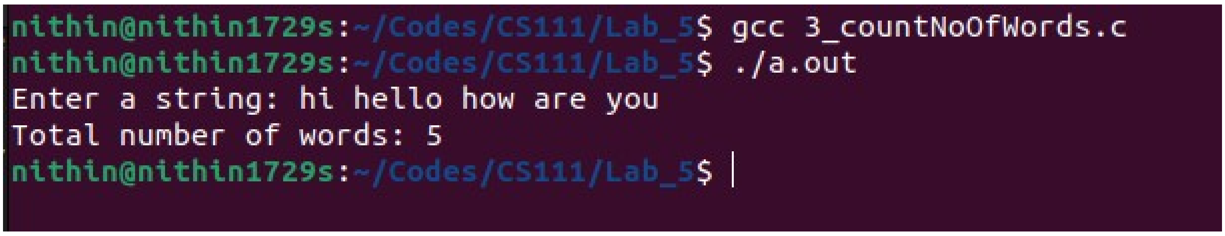


```
collect2: error: ld returned 1 exit status
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 2_countCharDigSpe.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: argentina$98
Total Alphabets: 9
Total Digits: 2
Total Special Characters: 1
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```

Q3) Write a C program to count the total number of words in a string

```
#include <stdio.h>
#include <stdbool.h>

#define MAX_LENGTH 100

bool isSeparator(char c) {
    return (c == ' ' || c == '\t' || c == '\n' || c == '\
r');
}

int countWords(char str[]) {
    int wordCount = 0;
    int i = 0;


    while (isSeparator(str[i])) {
        i++;
    }
    while (str[i] != '\0') {
        if (isSeparator(str[i])) {

            wordCount++;


            while (isSeparator(str[i])) {
                i++;
            }
        } else {
            i++;
        }
    }


    if (!isSeparator(str[i - 1])) {
        wordCount++;
    }

    return wordCount;
}

int main() {
    char str[MAX_LENGTH];

    printf("Enter a string: ");
    scanf("%99[^\n]", str);

    int totalWords = countWords(str);
    printf("Total number of words: %d\n", totalWords);

    return 0;
}
```

OUTPUT

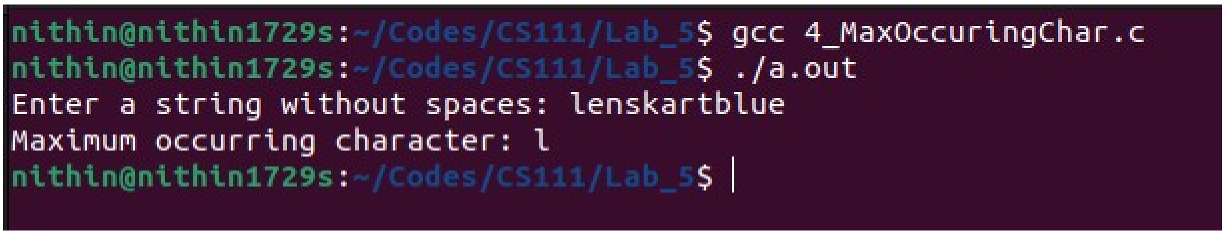


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 3_countNoOfWords.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: hi hello how are you
Total number of words: 5
nithin@nithin1729s:~/Codes/CS111/Lab_5$ |
```

Q4) Write a C program to find the maximum occurring character in a string.

```
#include <stdio.h>
#include <string.h>

#define MAX_LENGTH 100
#define ASCII_SIZE 256

char findMaxOccurringChar(char str[]) {
    int charCount[ASCII_SIZE] = {0};
    int maxCount = 0;
    char maxChar;
    int length = strlen(str);
    int i;


    for (i = 0; i < length; i++) {
        charCount[str[i]]++;
        if (charCount[str[i]] > maxCount) {
            maxCount = charCount[str[i]];
            maxChar = str[i];
        }
    }

    return maxChar;
}

int main() {
    char str[MAX_LENGTH];

    printf("Enter a string without spaces: ");
    scanf("%99s", str);

    char maxChar = findMaxOccurringChar(str);
    printf("Maximum occurring character: %c\n", maxChar);

    return 0;
}
```

OUTPUT

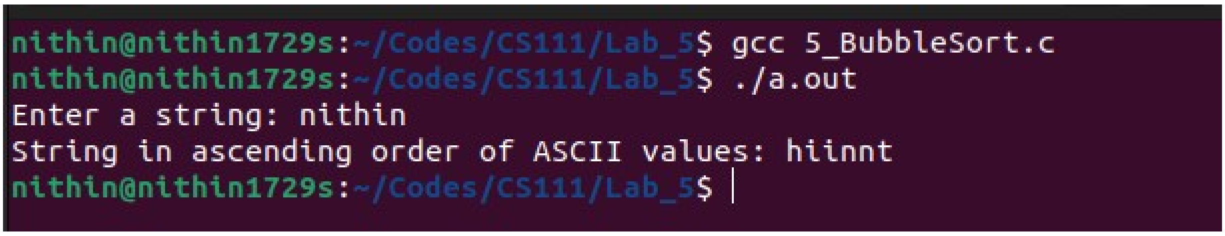


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 4_MaxOccuringChar.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string without spaces: lenskartblue
Maximum occurring character: l
nithin@nithin1729s:~/Codes/CS111/Lab_5$ |
```

Q5) Write a C program to sort a string array in ascending order using bubble sort.

```
#include <stdio.h>
#include <string.h>

#define MAX_LENGTH 100

void bubbleSort(char str[]) {
    int len = strlen(str);
    int i, j;
    char temp;

    for (i = 0; i < len - 1; i++) {
        for (j = 0; j < len - i - 1; j++) {
            if (str[j] > str[j + 1]) {
                temp = str[j];
                str[j] = str[j + 1];
                str[j + 1] = temp;
            }
        }
    }
}

int main() {
    char inputString[MAX_LENGTH];

    printf("Enter a string: ");
    scanf("%99s", inputString);

    bubbleSort(inputString);

    printf("String in ascending order of ASCII values:
%s\n", inputString);

    return 0;
}
```

OUTPUT

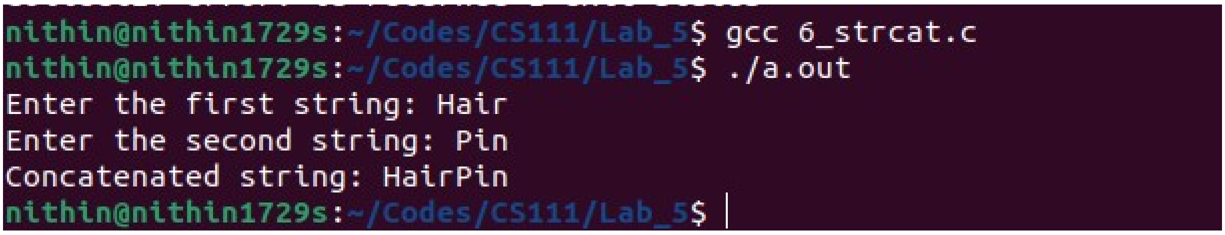


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 5_BubbleSort.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: nithin
String in ascending order of ASCII values: hiinnt
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```

Q6) Concatenate Two Strings Without Using strcat()

```
#include <stdio.h>
#define MAX_LENGTH 100

void concatenateStrings(char result[], const char str1[],
const char str2[]) {
    int i = 0, j = 0;


    while (str1[i] != '\0') {
        result[j] = str1[i];
        i++;
        j++;
    }


    i = 0;
    while (str2[i] != '\0') {
        result[j] = str2[i];
        i++;
        j++;
    }


    result[j] = '\0';
}

int main() {
    char str1[MAX_LENGTH], str2[MAX_LENGTH],
result[MAX_LENGTH * 2];

    printf("Enter the first string: ");
    scanf("%99s", str1);

    printf("Enter the second string: ");
    scanf("%99s", str2);

    concatenateStrings(result, str1, str2);

    printf("Concatenated string: %s\n", result);

    return 0;
}
```

OUTPUT

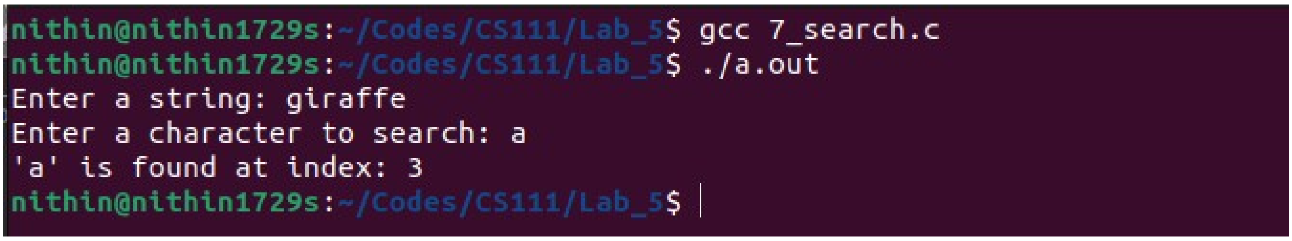


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 6_strcat.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter the first string: Hair
Enter the second string: Pin
Concatenated string: HairPin
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```

Q7) Program to search a string

```
#include <stdio.h>
#include <stdbool.h>

int findCharIndex(const char str[], char letter) {
    int i = 0;
    while (str[i] != '\0') {
        if (str[i] == letter) {
            return i;
        }
        i++;
    }
    return -1;
}

int main() {
    char inputString[100];
    char searchChar;

    printf("Enter a string: ");
    scanf("%99s", inputString);

    printf("Enter a character to search: ");
    scanf(" %c", &searchChar);

    int index = findCharIndex(inputString, searchChar);

    if (index != -1) {
        printf("'%c' is found at index: %d\n", 
searchChar, index);
    } else {
        printf("'%c' is not found in the string.\n", 
searchChar);
    }

    return 0;
}
```

OUTPUT

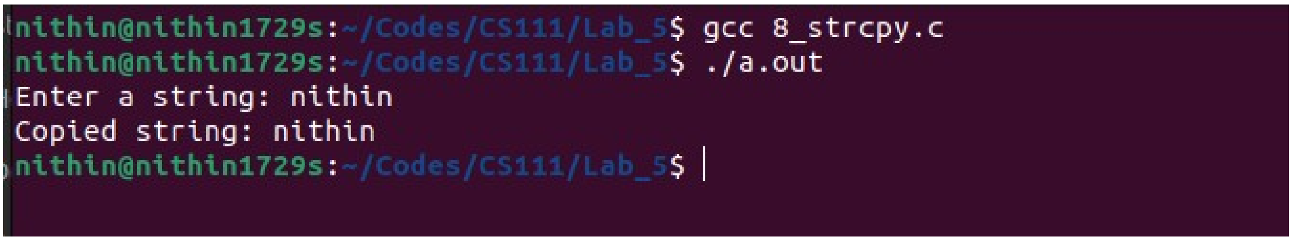


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 7_search.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: giraffe
Enter a character to search: a
'a' is found at index: 3
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```

Q8) Program to Copy String Without Using strcpy()

```
#include <stdio.h>

void copyString(char *destination, const char *source) {
    while (*source != '\0') {
        *destination = *source;
        source++;
        destination++;
    }
    *destination = '\0';
}

int main() {
    char sourceString[100];
    char destinationString[100];

    printf("Enter a string: ");
    scanf("%99s", sourceString);

    copyString(destinationString, sourceString);

    printf("Copied string: %s\n", destinationString);

    return 0;
}
```

OUTPUT



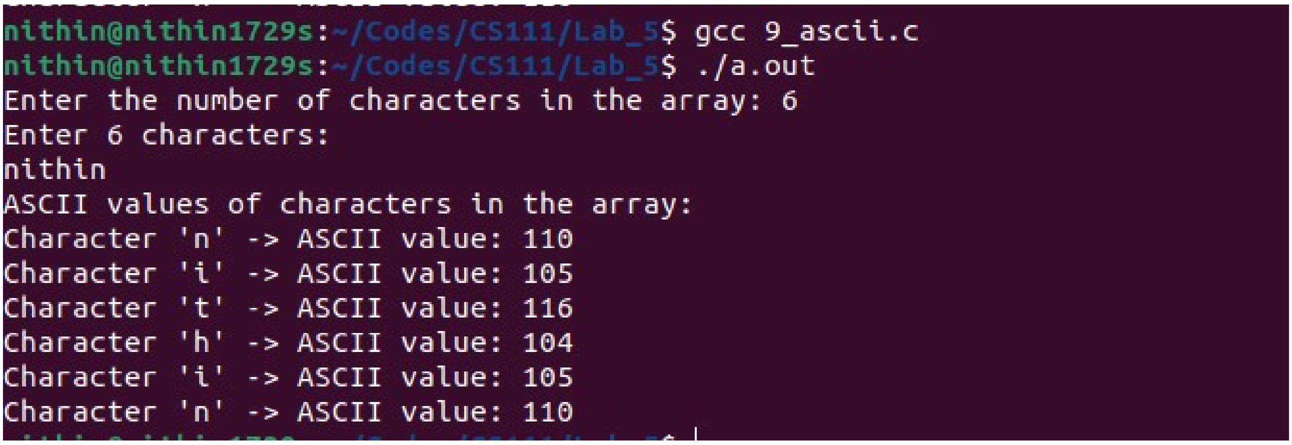
```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 8_strcpy.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a string: nithin
Copied string: nithin
nithin@nithin1729s:~/Codes/CS111/Lab_5$
```

FUNCTIONS

Q1) Write a c program to read characters from an array and display its ascii value, using functions

```
#include <stdio.h>

#define MAX_SIZE 100

void displayAsciiValues(const char arr[], int size) {
    printf("ASCII values of characters in the array:\n");
    for (int i = 0; i < size; i++) {
        printf("Character '%c' -> ASCII value: %d\n",
arr[i], arr[i]);
    }
}

int main() {
    char characters[MAX_SIZE];
    int size;

    printf("Enter the number of characters in the array:
");
    scanf("%d", &size);

    if (size > MAX_SIZE || size <= 0) {
        printf("Invalid array size.\n");
        return 1;
    }

    printf("Enter %d characters:\n", size);
    for (int i = 0; i < size; i++) {
        scanf(" %c", &characters[i]);
    }

    displayAsciiValues(characters, size);

    return 0;
}
```

OUTPUT

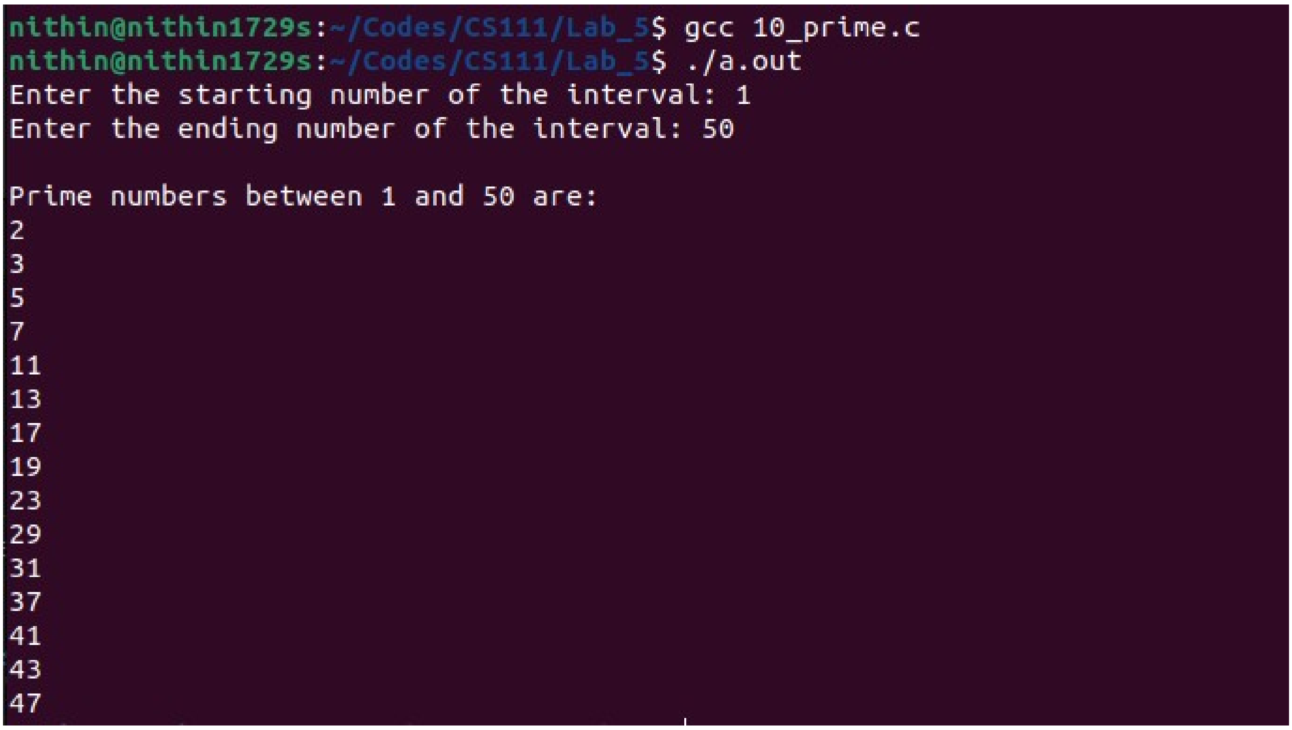


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 9_ascii.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter the number of characters in the array: 6
Enter 6 characters:
nithin
ASCII values of characters in the array:
Character 'n' -> ASCII value: 110
Character 'i' -> ASCII value: 105
Character 't' -> ASCII value: 116
Character 'h' -> ASCII value: 104
Character 'i' -> ASCII value: 105
Character 'n' -> ASCII value: 110
```

Q2) Write a C program to display prime numbers between intervals using function.

```
#include <stdio.h>
#include <stdbool.h>

bool isPrime(int num) {
    if (num <= 1) {
        return false;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

void displayPrimeNumbers(int start, int end) {
    printf("Prime numbers between %d and %d are:\n",
start, end);
    for (int i = start; i <= end; i++) {
        if (isPrime(i)) {
            printf("%d\n", i);
        }
    }
}

int main() {
    int start, end;

    printf("Enter the starting number of the interval: 
");
    scanf("%d", &start);

    printf("Enter the ending number of the interval: ");
    scanf("%d", &end);

    printf("\n");

    if (start >= end || start < 0 || end < 0) {
        printf("Invalid input for interval.\n");
        return 1;
    }

    displayPrimeNumbers(start, end);

    return 0;
}
```

OUTPUT

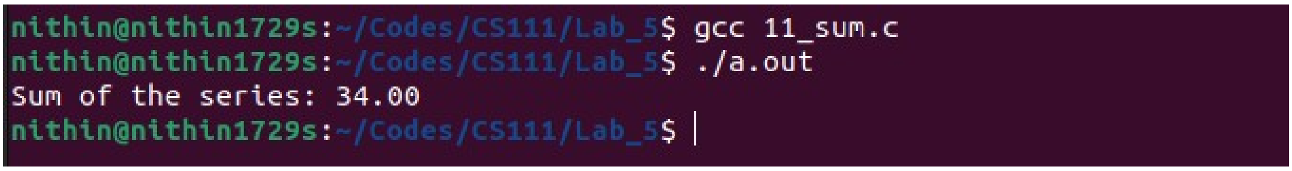


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 10_prime.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter the starting number of the interval: 1
Enter the ending number of the interval: 50

Prime numbers between 1 and 50 are:
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

Q3) Write a program in C to find the sum of the series 1!/1+2!/2+3!/3+4!/4+5!/5 using the function by passing appropriate parameters and return value.

```
#include <stdio.h>


unsigned long long factorial(int num) {
    unsigned long long fact = 1;
    for (int i = 1; i <= num; i++) {
        fact *= i;
    }
    return fact;
}


double sumOfSeries() {
    double sum = 0.0;
    for (int i = 1; i <= 5; i++) {
        sum += (double)factorial(i) / i;
    }
    return sum;
}

int main() {
    double result = sumOfSeries();

    printf("Sum of the series: %.2f\n", result);

    return 0;
}
```

OUTPUT

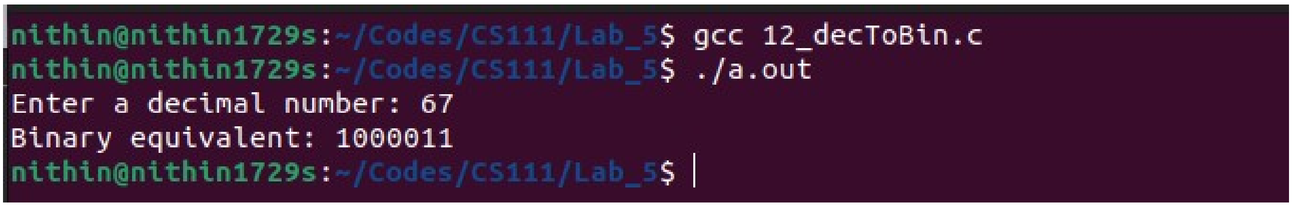


```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 11_sum.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Sum of the series: 34.00
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```

Q4) Write a C program to convert decimal numbers to binary using functions

```
#include <stdio.h>


void decimalToBinary(int num) {
    int binary[32];
    int index = 0;


    if (num == 0) {
        printf("Binary equivalent: 0\n");
        return;
    }

    while (num > 0) {
        binary[index] = num % 2;
        num = num / 2;
        index++;
    }

    printf("Binary equivalent: ");
    for (int i = index - 1; i >= 0; i--) {
        printf("%d", binary[i]);
    }
    printf("\n");
}

int main() {
    int decimalNumber;

    printf("Enter a decimal number: ");
    scanf("%d", &decimalNumber);

    decimalToBinary(decimalNumber);

    return 0;
}
```

OUTPUT



```
nithin@nithin1729s:~/Codes/CS111/Lab_5$ gcc 12_decToBin.c
nithin@nithin1729s:~/Codes/CS111/Lab_5$ ./a.out
Enter a decimal number: 67
Binary equivalent: 1000011
nithin@nithin1729s:~/Codes/CS111/Lab_5$ 
```