```c
#include <stdio.h>
int checkPrime(int n);
int main() {
  int n, i, flag = 0;
  printf("Enter a positive integer: ");
  scanf("%d", &n);

  for (i = 2; i <= n / 2; ++i) {
    // condition for i to be a prime number
    if (checkPrime(i) == 1) {
      // condition for n-i to be a prime number
      if (checkPrime(n - i) == 1) {
        printf("%d = %d + %d\n", n, i, n - i);
        flag = 1;
      }
    }
  }

  if (flag == 0)
    printf("%d cannot be expressed as the sum of two prime numbers.", n);

  return 0;
}

// function to check prime number
int checkPrime(int n) {
  int i, isPrime = 1;

  // 0 and 1 are not prime numbers
  if (n == 0 || n == 1) {
    isPrime = 0;
  }
  else {
    for(i = 2; i <= n/2; ++i) {
      if(n % i == 0) {
        isPrime = 0;
        break;
      }
    }
  }

  return isPrime;
}
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Users\lakshmanan> cd "C:\Users\LAKSHM-1\AppData\Local\Temp"
PS C:\Users\lakshmanan\AppData\Local\Temp> cd "C:\Users\LAKSHM-1\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter a positive integer: 34
34 = 3 + 31
34 = 5 + 29
34 = 11 + 23
34 = 17 + 17
PS C:\Users\lakshmanan\AppData\Local\Temp>

```c
#include <stdio.h>

// user-defined function to check prime number
int checkPrimeNumber(int number)
{
    int i, f = 1;

    // condition for finding the
    // prime numbers between the
    // given intervals
    for (i = 2; i <= number / 2; ++i) {

        if (number % i == 0) {
            f = 0;
            break;
        }
    }

    return f;
}

int main()
{

    int num1,num2,j,f;

    printf("enter the two ranges");
    scanf("%d%d",&num1,&num2);
    for (j = num1; j < num2; ++j) {

        // if flag is equal to 1 then
        // it is a prime number
        // calling the function
        f = checkPrimeNumber(j);

        if (f == 1) {

            // printing the result
            printf("%d ", j);
        }
    }

    return 0;
```

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    int num1, num2, max_div, flag = 1;
    // accept any two positive number from the user
    printf( " Enter any two positive numbers to get the LCM \n ");
    scanf(" %d %d", &num1, &num2);

    // max_div variable holds the max divisible number between num1 and num2.
    max_div = (num1 > num2) ? num1 : num2;

    while (flag) // (flag = 1)
    {
        if (max_div % num1 == 0 && max_div % num2 == 0)
        {
            printf( " The LCM of %d and %d is %d. ", num1, num2, max_div);
            break;
        }
        ++max_div; // pre-increment max_div
    }
}
```

```c
#include <stdio.h>
int hcf(int n1, int n2);
int main() {
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
    return 0;
}

int hcf(int n1, int n2) {
    if (n2 != 0)
        return hcf(n2, n1 % n2);
    else
        return n1;
}
```

```c
#include <stdio.h>

int find_anagram(char [], char []);

int main()
{
    char array1[100], array2[100];
    int flag;

    printf("Enter the string\n");
    gets(array1);
    printf("Enter another string\n");
    gets(array2);
    flag = find_anagram(array1, array2);
    if (flag == 1)
        printf("%s and %s are anagrams.\n", array1, array2);
    else
        printf(" %s and %s are not anagrams.\n", array1, array2);
    return 0;
}

int find_anagram(char array1[], char array2[])
{
    int num1[26] = {0}, num2[26] = {0}, i = 0;

    while (array1[i] != '\0')
    {
        num1[array1[i] - 'a']++;
        i++;
    }
    i = 0;
    while (array2[i] != '\0')
    {
        num2[array2[i] -'a']++;
        i++;
    }
    for (i = 0; i < 26; i++)
    {
        if (num1[i] != num2[i])
            return 0;
    }
    return 1;
}
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

PS C:\Users\lakshmanan> cd "C:\Users\LAKSHM-1\AppData\Local\Temp"
PS C:\Users\lakshmanan\AppData\Local\Temp> cd "C:\Users\LAKSHM-1\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter the string
hectare
Enter another string
teacher
hectare and teacher are anagrams.
PS C:\Users\lakshmanan\AppData\Local\Temp>

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

char string1[100], visited[100];
int count[100] = {0}, flag = 0;

void main()
{
    int i, j = 0, k = 0, l, max, index;

    printf("Enter a string : ");
    scanf("%[^\n]s", string1);

    l = strlen(string1);

    for (i = 0; i < l; i++)
    {
        if (i == 0)
        {
            visited[j++] = string1[i];
            count[j - 1]++;
        }
        else
        {
            for (k = 0; k  < j; k++)
            {
                if (string1[i] == visited[k])
                {
                    count[k]++;
                    flag = 1;
                }
            }
            if (flag == 0)
            {
                visited[j++] = string1[i];
                count[j - 1]++;
            }
            flag = 0;
        }
    }

    for (i = 0; i < j; i++)
    {
```

```c
1   // C program to print all permutations with duplicates
2   // allowed
3   #include <stdio.h>
4   #include <string.h>
5
6   /* Function to swap values at two pointers */
7   void swap(char* x, char* y)
8   {
9       char temp;
10      temp = *x;
11      *x = *y;
12      *y = temp;
13  }
14
15  /* Function to print permutations of string
16  This function takes three parameters:
17  1. String
18  2. Starting index of the string
19  3. Ending index of the string. */
20  void permute(char* a, int l, int r)
21  {
22      int i;
23      if (l == r)
24          printf("%s\n", a);
25      else {
26          for (i = l; i <= r; i++) {
27              swap((a + l), (a + i));
28              permute(a, l + 1, r);
29              swap((a + l), (a + i)); // backtrack
30          }
31      }
32  }
33
34  /* Driver code */
35  int main()
36  {
37      char str[] = "SAN";
38      int n = strlen(str);
39      permute(str, 0, n - 1);
40      return 0;
41  }
42
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
> cd "C:\Users\LAKSHM-1\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
SAN
SNA
ASN
ANS
NAS
NSA
PS C:\Users\lakshmanan\AppData\Local\Temp>
```

```c
#include <stdio.h>

int main()
{
    //Initialize array
    int arr1[] = {10, -1, 100, 90, 87, 0, 15, 10, 20, 30};

    //Calculate length of array arr1
    int length = sizeof(arr1)/sizeof(arr1[0]);

    //Create another array arr2 with the size of arr1.

    int arr2[length];

    //Copying all elements of one array into another
    for (int i = 0; i < length; i++) {
        arr2[i] = arr1[i];
    }

    //Displaying elements of array arr1
    printf("Elements of original array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr1[i]);
    }

    printf("\n");

    //Displaying elements of array arr2
    printf("Elements of new array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr2[i]);
    }
    return 0;
}
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
> cd "C:\Users\LAKSHM-1\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Elements of original array:
10 -1 100 90 87 0 15 10 20 30
Elements of new array:
10 -1 100 90 87 0 15 10 20 30
PS C:\Users\lakshmanan\AppData\Local\Temp>
```

```c
1  #include <stdio.h>
2  #include <string.h>
3  int main()
4  {
5      char str[40]; // declare the size of character string
6      printf (" \n Enter a string to be reversed: ");
7      scanf ("%s", str);
8
9      // use strrev() function to reverse a string
10     printf (" \n After the reverse of a string: %s ", strrev(str));
11     return 0;
12 }
```

> cd "C:\Users\LAKSHN-1\AppData\Local\Temp\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }

Enter a string to be reversed: Saveetha

After the reverse of a string: ahteevaS
PS C:\Users\lakshmanan\AppData\Local\Temp>

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

void main()
{
    int sum = 0, i, len;
    char string1[100];

    printf("Enter the string : ");
    scanf("%[^\n]s", string1);
        len = strlen(string1);
    for (i = 0; i < len; i++)
    {
        sum = sum + string1[i];
    }
    printf("\nSum of all characters : %d ",sum);
}
```