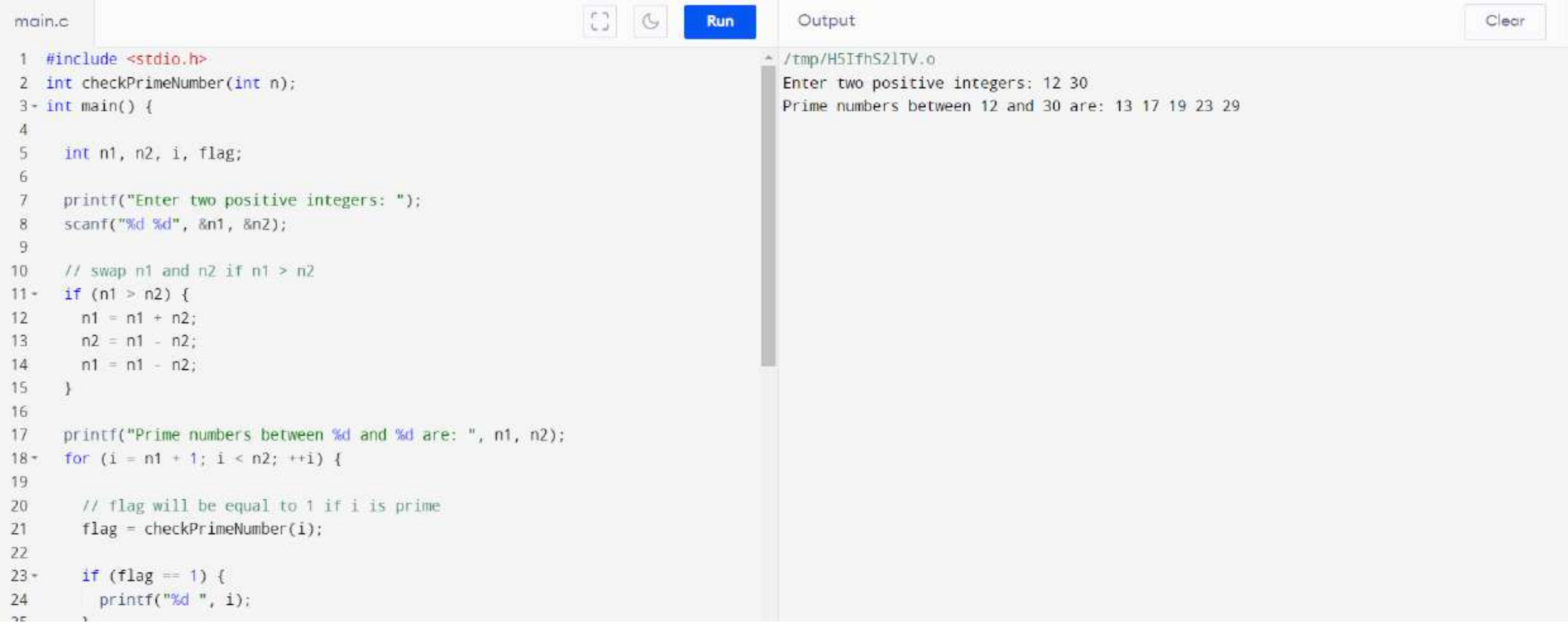
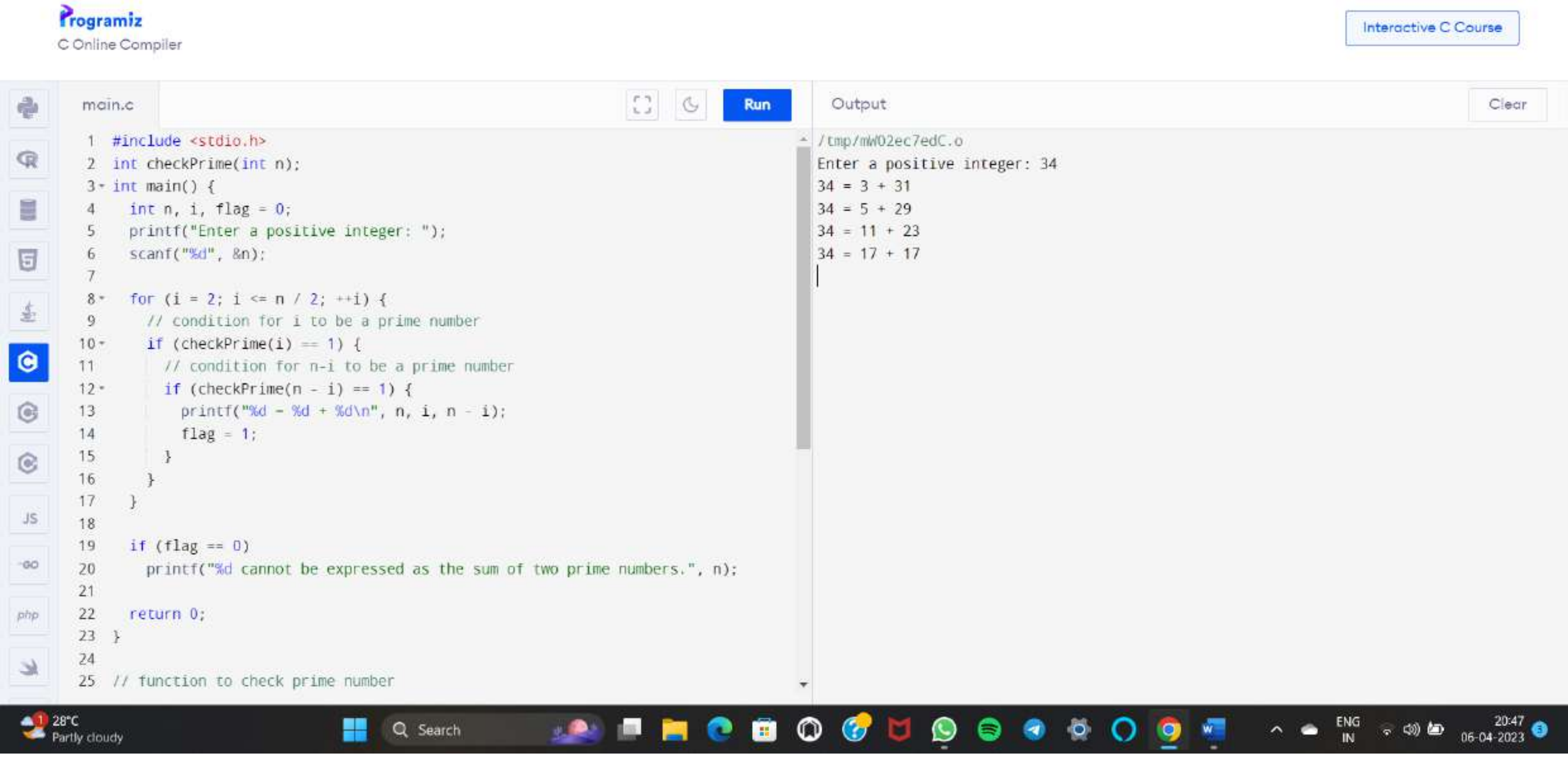


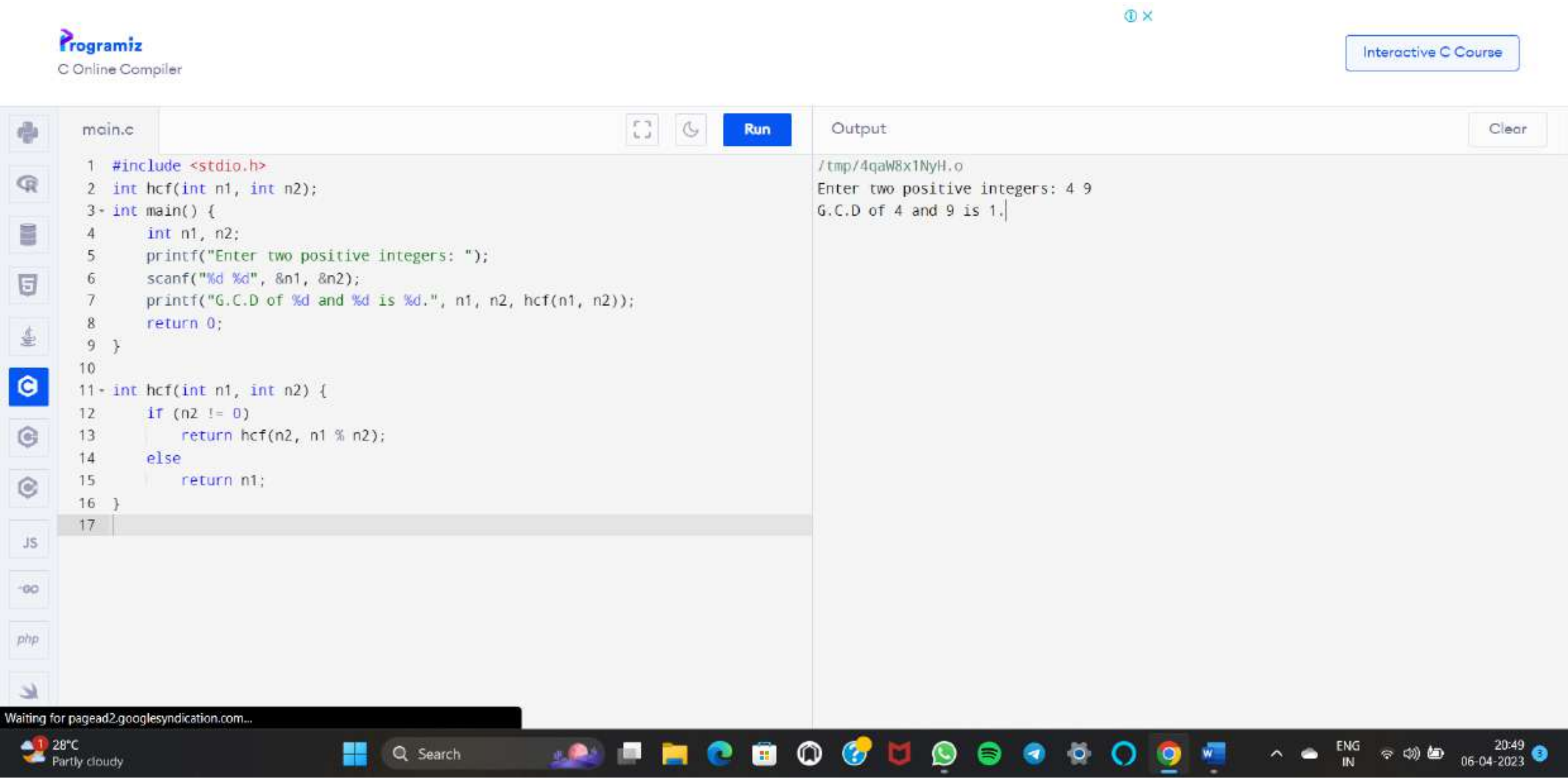
```c
#include <stdio.h>
int checkPrimeNumber(int n);
int main() {

  int n1, n2, i, flag;

  printf("Enter two positive integers: ");
  scanf("%d %d", &n1, &n2);

  // swap n1 and n2 if n1 > n2
  if (n1 > n2) {
    n1 = n1 + n2;
    n2 = n1 - n2;
    n1 = n1 - n2;
  }

  printf("Prime numbers between %d and %d are: ", n1, n2);
  for (i = n1 + 1; i < n2; ++i) {

    // flag will be equal to 1 if i is prime
    flag = checkPrimeNumber(i);

    if (flag == 1) {
      printf("%d ", i);
    }
```

```
/tmp/H5IfhS2lTV.o
Enter two positive integers: 12 30
Prime numbers between 12 and 30 are: 13 17 19 23 29
```



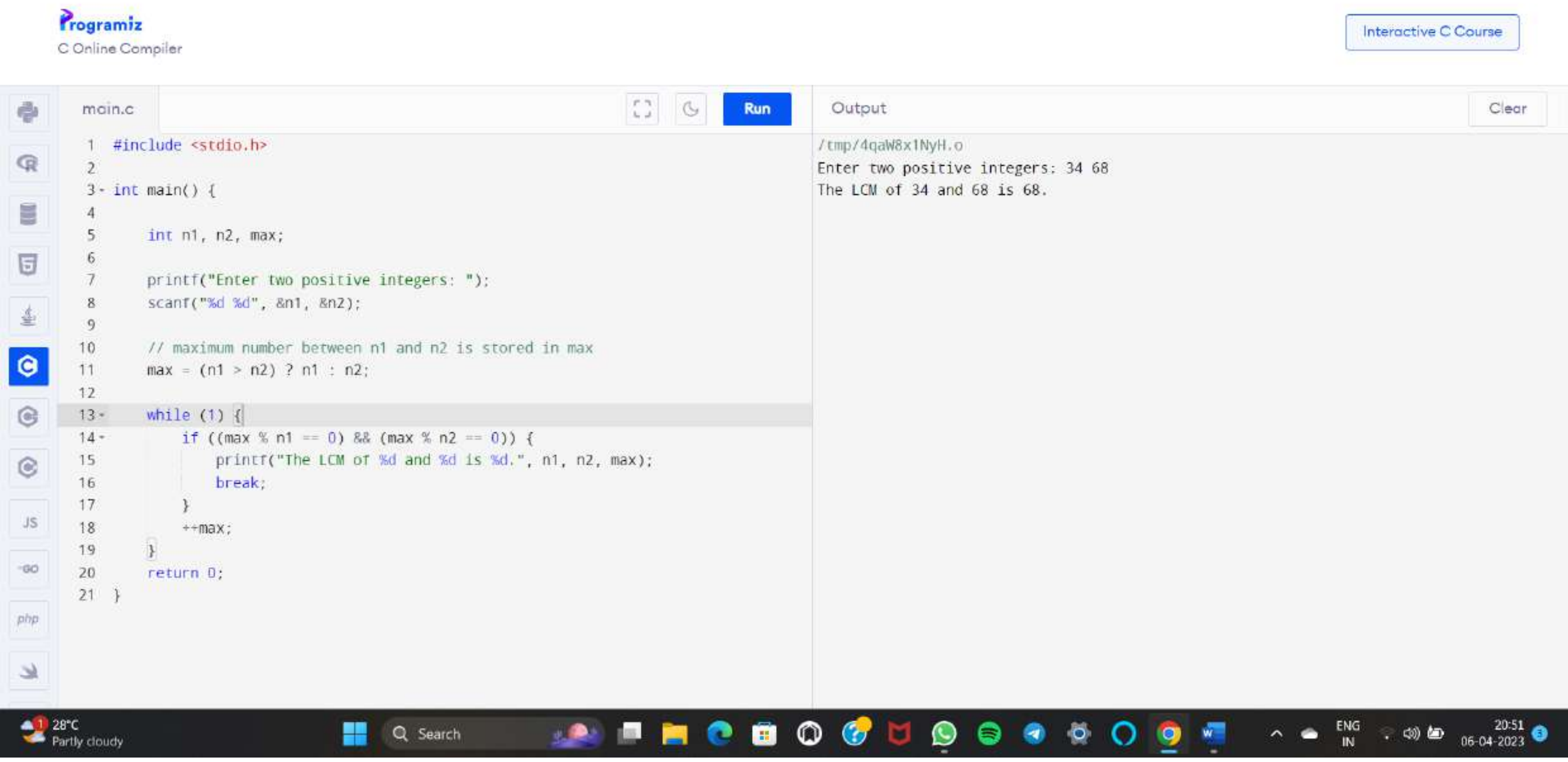
```c
#include <stdio.h>
int checkPrime(int n);
int main() {
  int n, i, flag = 0;
  printf("Enter a positive integer: ");
  scanf("%d", &n);

  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
```

```
/tmp/mW02ec7edC.o
Enter a positive integer: 34
34 = 3 + 31
34 = 5 + 29
34 = 11 + 23
34 = 17 + 17
```



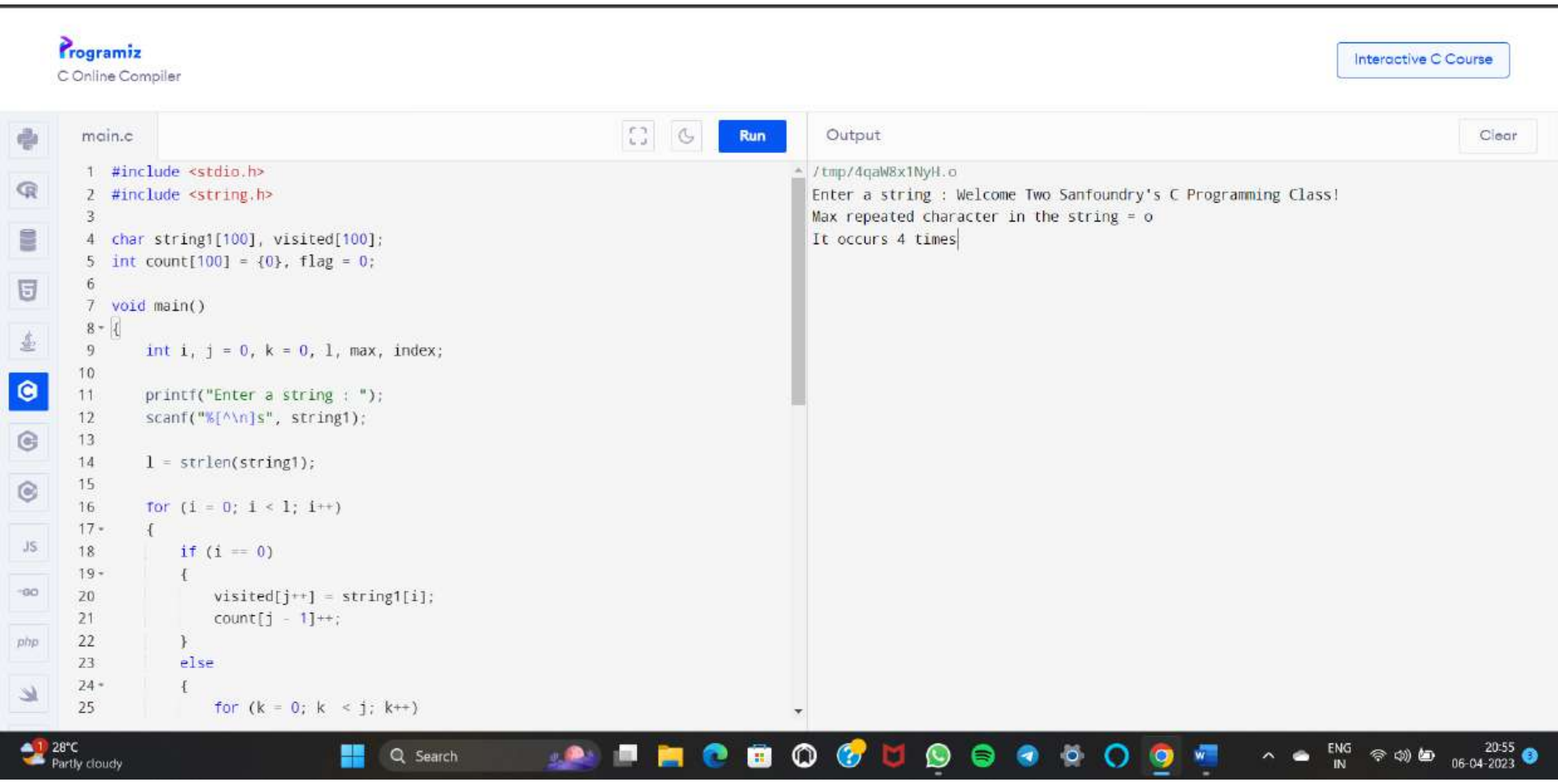
```c
#include <stdio.h>
int hcf(int n1, int n2);
int main() {
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
    return 0;
}

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

```
/tmp/4qaW8x1NyH.o
Enter two positive integers: 4 9
G.C.D of 4 and 9 is 1.
```



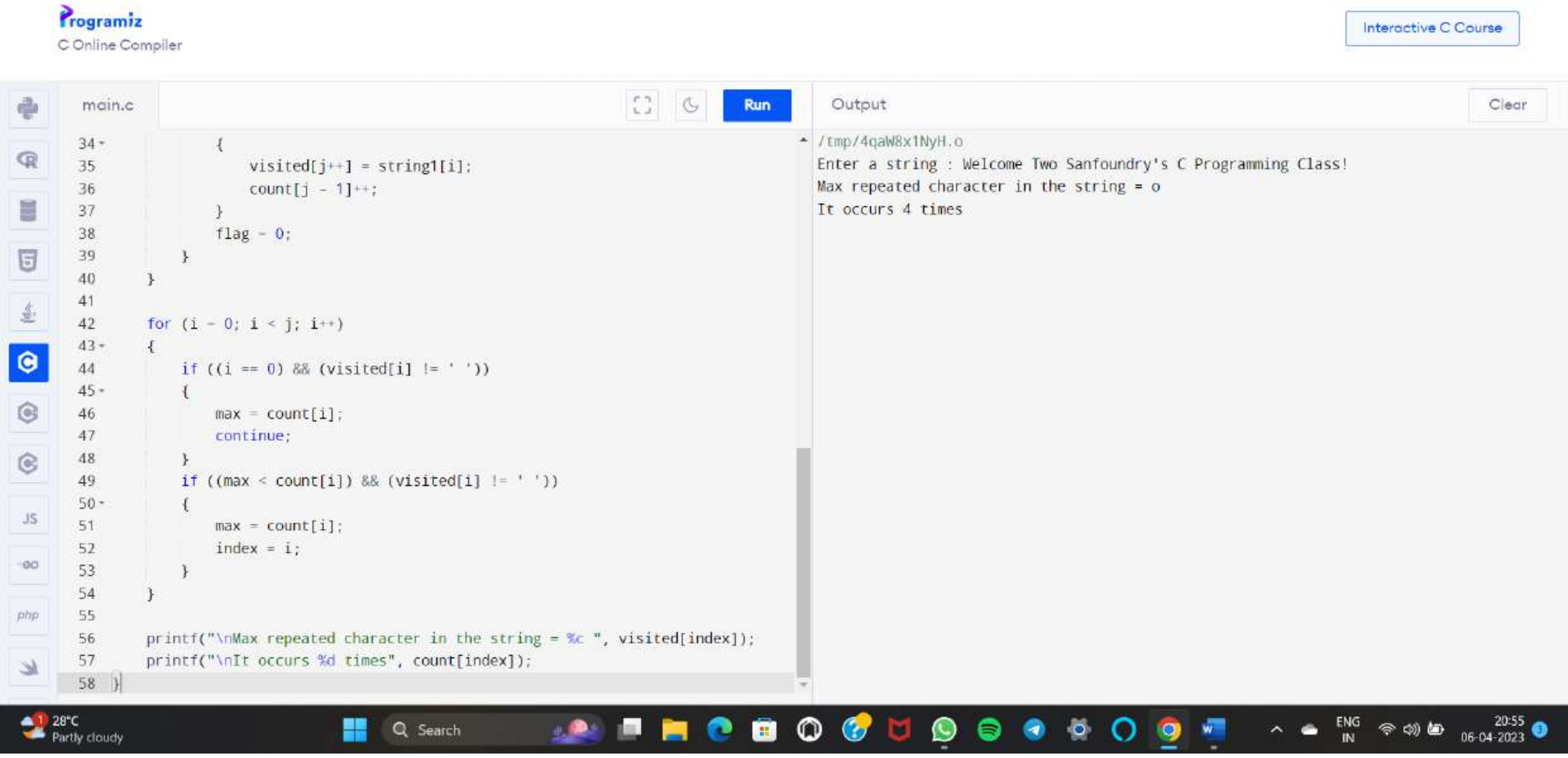
```c
#include <stdio.h>

int main() {

    int n1, n2, max;

    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);

    // maximum number between n1 and n2 is stored in max
    max = (n1 > n2) ? n1 : n2;

    while (1) {
        if ((max % n1 == 0) && (max % n2 == 0)) {
            printf("The LCM of %d and %d is %d.", n1, n2, max);
            break;
        }
        ++max;
    }
    return 0;
}
```

```
/tmp/4qaW8x1NyH.o
Enter two positive integers: 34 68
The LCM of 34 and 68 is 68.
```



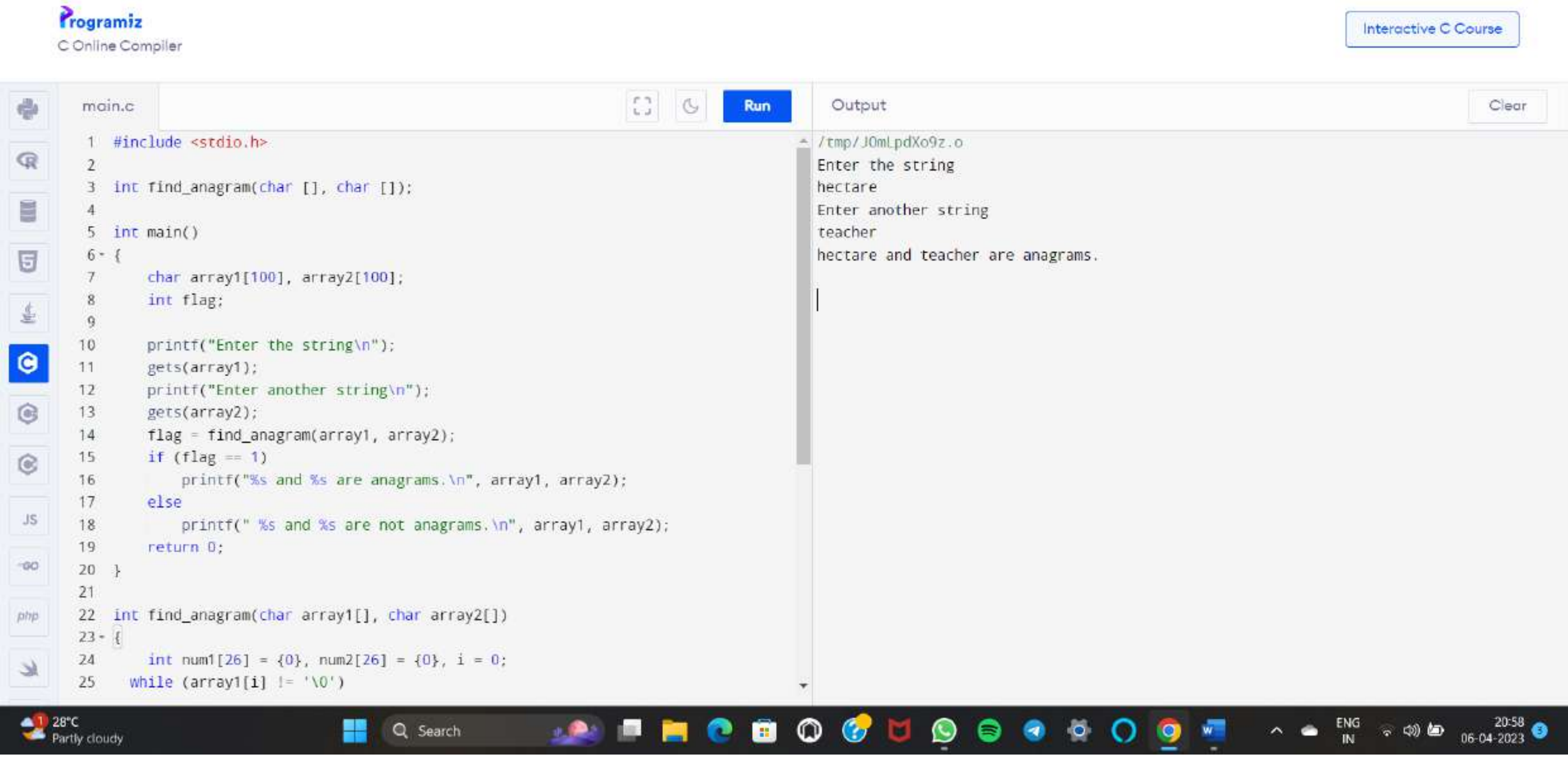
```c
#include <stdio.h>
#include <string.h>

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++)
```

```
/tmp/4qaW8x1NyH.o
Enter a string : Welcome Two Sanfoundry's C Programming Class!
Max repeated character in the string = o
It occurs 4 times
```



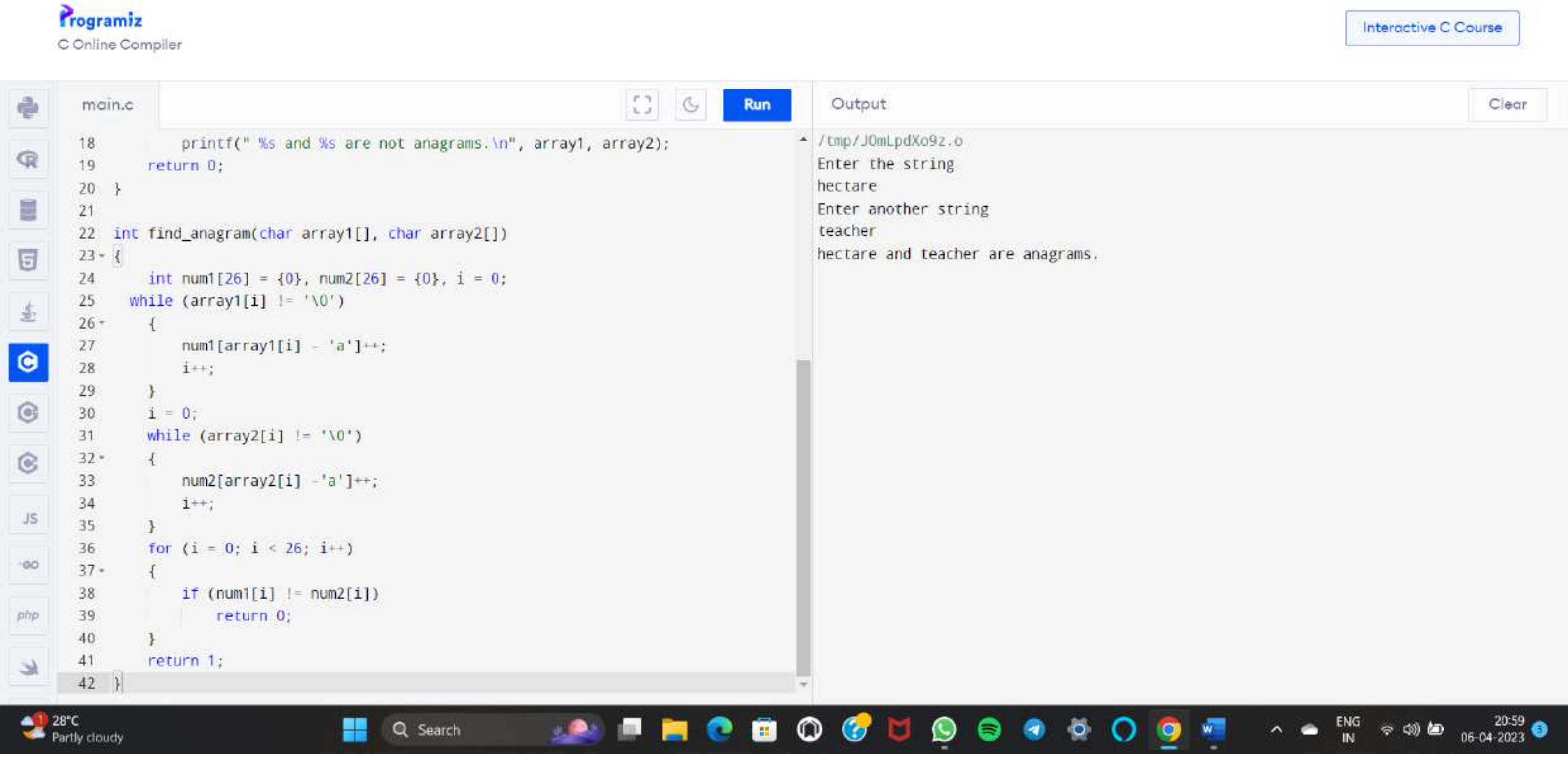
```
            {
                visited[j++] = string1[i];
                count[j - 1]++;
            }
            flag = 0;
        }
    }

    for (i = 0; i < j; i++)
    {
        if ((i == 0) && (visited[i] != ' '))
        {
            max = count[i];
            continue;
        }
        if ((max < count[i]) && (visited[i] != ' '))
        {
            max = count[i];
            index = i;
        }
    }

    printf("\nMax repeated character in the string = %c ", visited[index]);
    printf("\nIt occurs %d times", count[index]);
}
```

```
/tmp/4qaW8x1NyH.o
Enter a string : Welcome Two Sanfoundry's C Programming Class!
Max repeated character in the string = o
It occurs 4 times
```



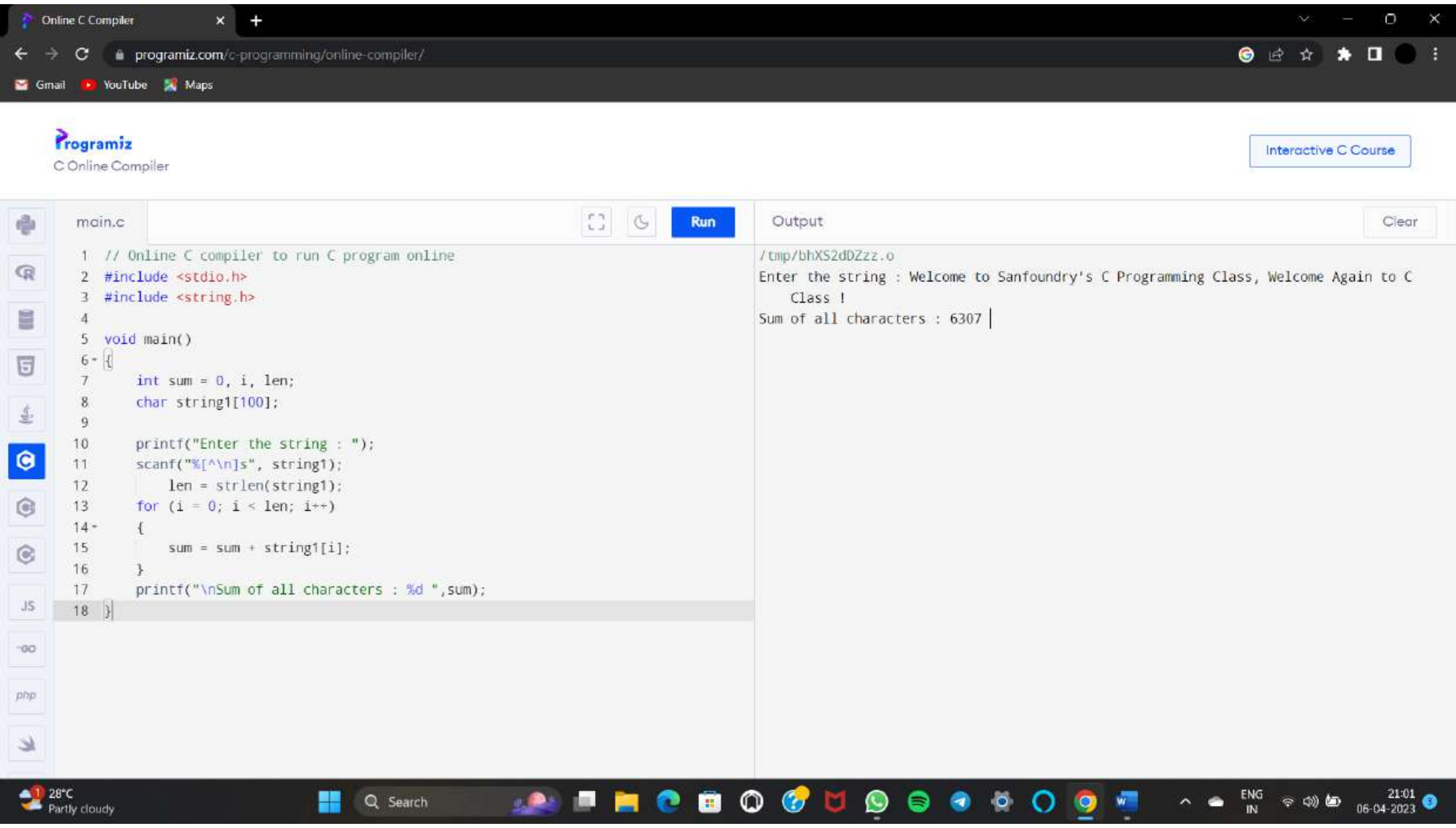
```c
#include <stdio.h>

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')
```

```
/tmp/J0mLpdXo9z.o
Enter the string
hectare
Enter another string
teacher
hectare and teacher are anagrams.
```



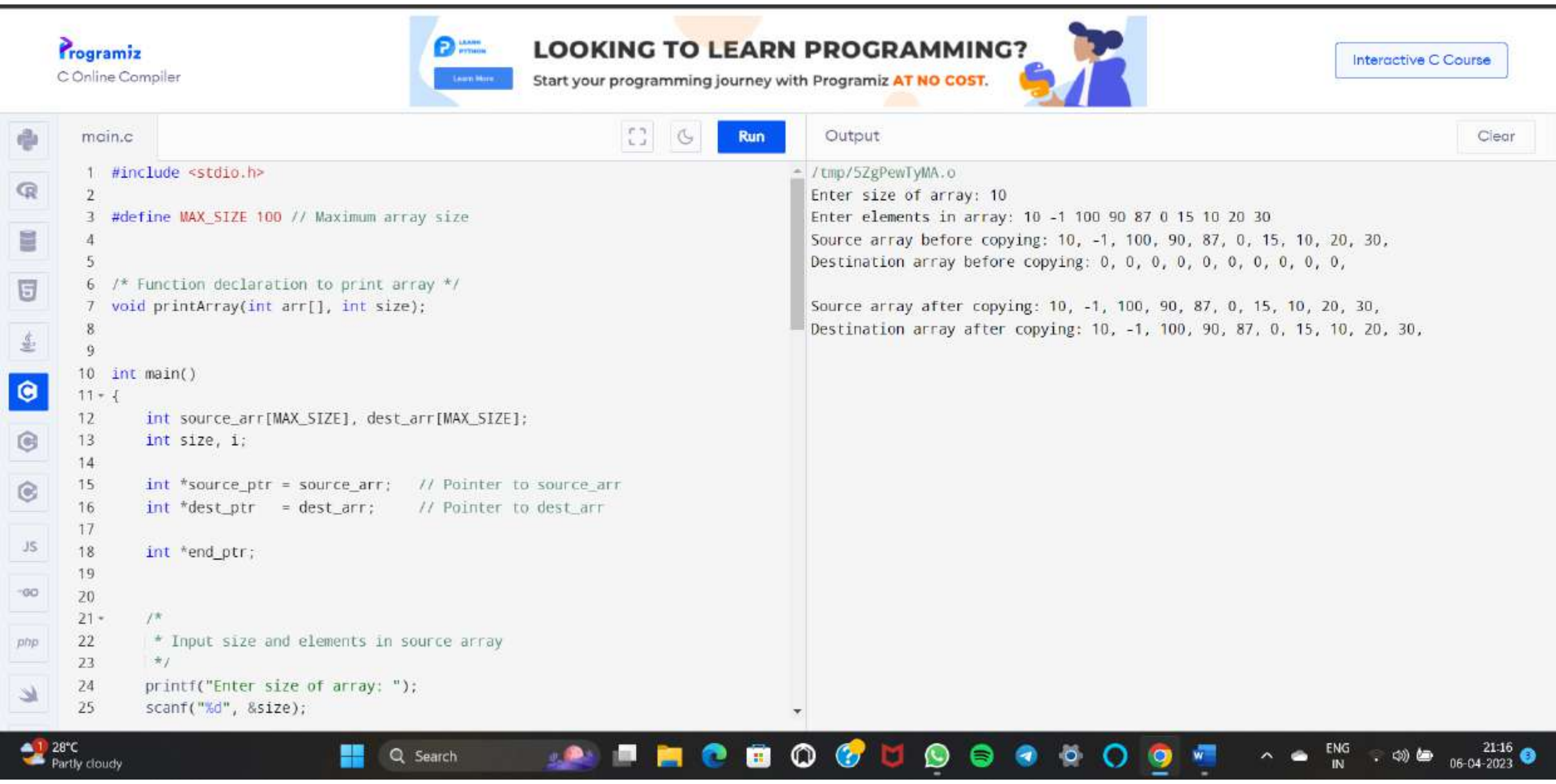
```
        printf(" %s and %s are not anagrams.\n", array1, array2);
    return 0;
}

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;
}
```

```
/tmp/JOmLpdXo9z.o
Enter the string
hectare
Enter another string
teacher
hectare and teacher are anagrams.
```



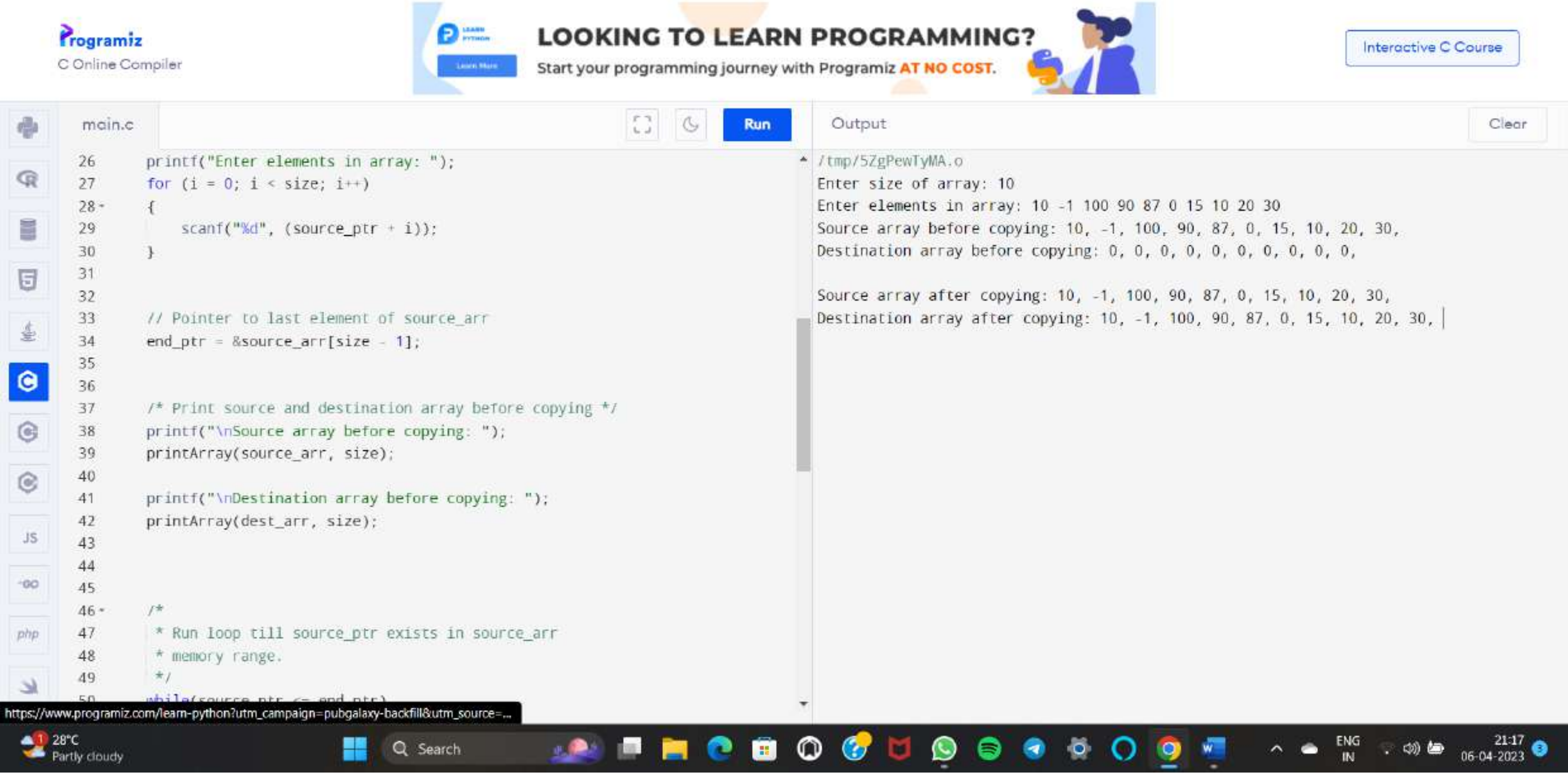
```
// Online C compiler to run C program online
#include <stdio.h>
#include <string.h>

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);
}
```

```
/tmp/bhXS2dDZzz.o
Enter the string : Welcome to Sanfoundry's C Programming Class, Welcome Again to C
    Class !
Sum of all characters : 6307
```



```c
#include <stdio.h>

#define MAX_SIZE 100 // Maximum array size


/* Function declaration to print array */
void printArray(int arr[], int size);


int main()
{
    int source_arr[MAX_SIZE], dest_arr[MAX_SIZE];
    int size, i;

    int *source_ptr = source_arr;   // Pointer to source_arr
    int *dest_ptr   = dest_arr;     // Pointer to dest_arr

    int *end_ptr;


    /*
     * Input size and elements in source array
     */
    printf("Enter size of array: ");
    scanf("%d", &size);
```

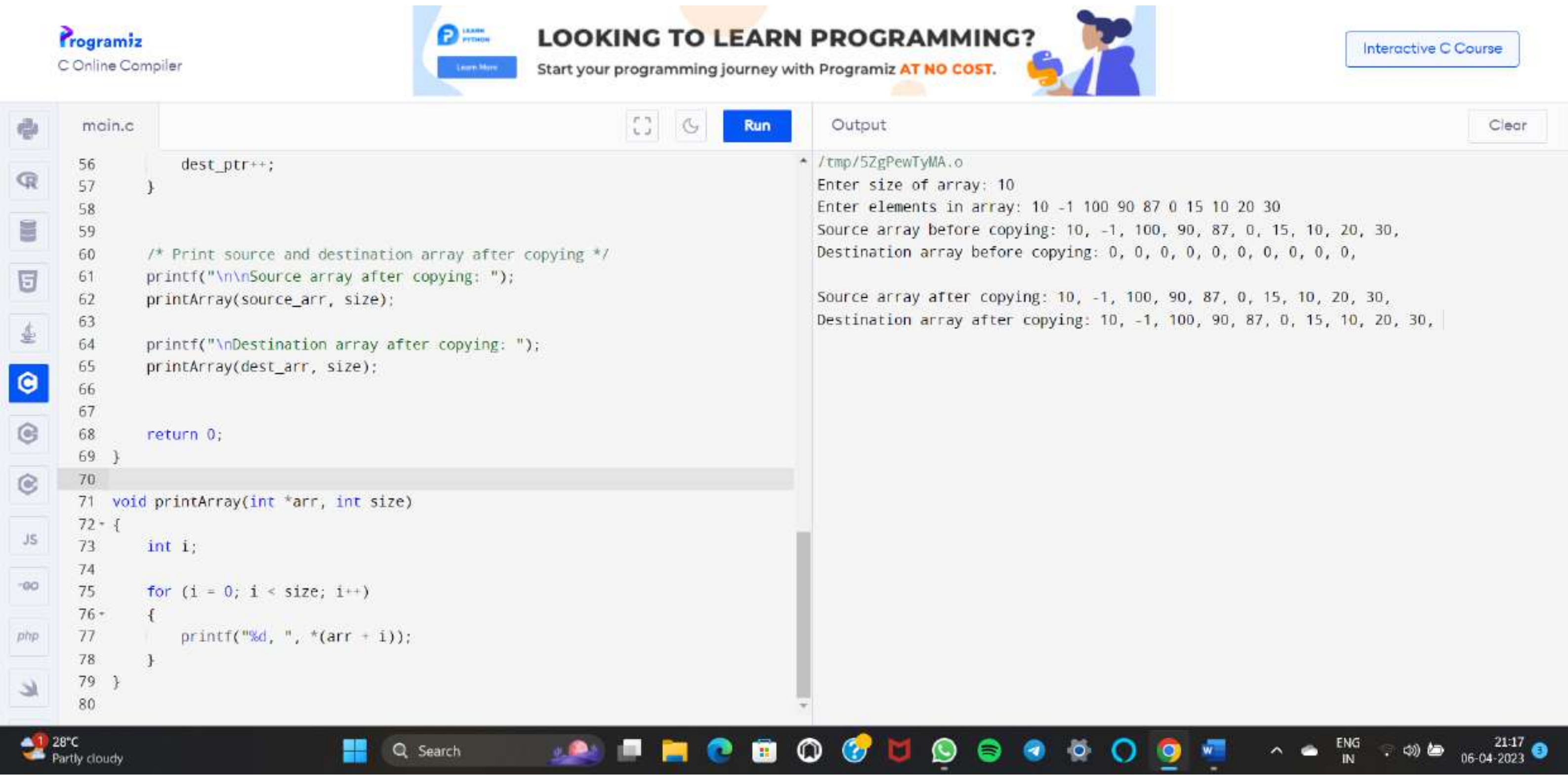
```
/tmp/5ZgPewTyMA.o
Enter size of array: 10
Enter elements in array: 10 -1 100 90 87 0 15 10 20 30
Source array before copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array before copying: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Source array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
```



Programiz
C Online Compiler
LOOKING TO LEARN PROGRAMMING?
Start your programming journey with Programiz AT NO COST.
Interactive C Course
main.c
Run
Output
Clear
https://www.programiz.com/learn-python?utm_campaign=pubgalaxy-backfill&utm_source=...
28°C
Partly cloudy
Search
ENG
IN
21:17
06-04-2023

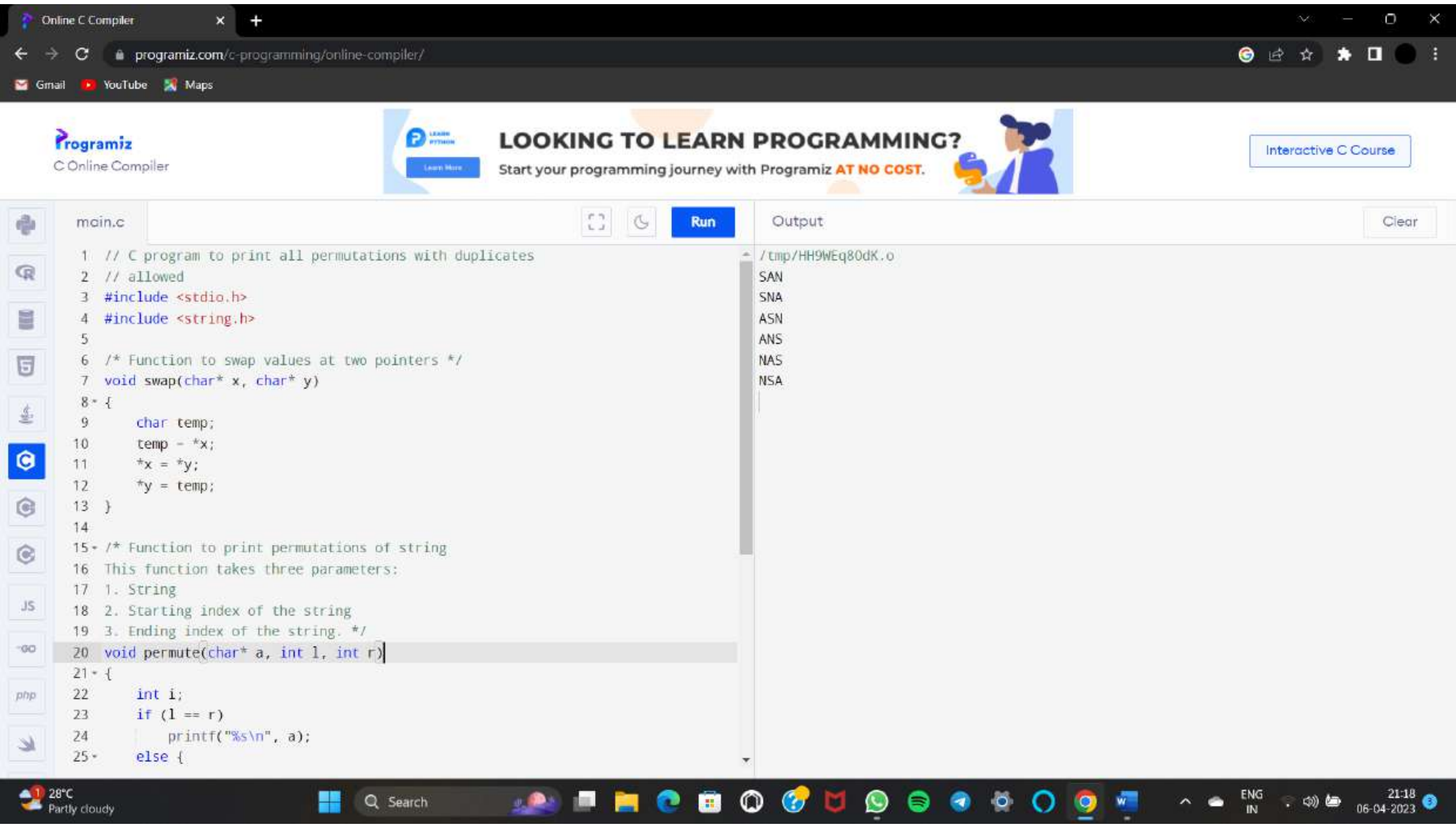
```c
    printf("Enter elements in array: ");
    for (i = 0; i < size; i++)
    {
        scanf("%d", (source_ptr + i));
    }


    // Pointer to last element of source_arr
    end_ptr = &source_arr[size - 1];


    /* Print source and destination array before copying */
    printf("\nSource array before copying: ");
    printArray(source_arr, size);

    printf("\nDestination array before copying: ");
    printArray(dest_arr, size);



    /*
     * Run loop till source_ptr exists in source_arr
     * memory range.
     */
    while(source_ptr <= end_ptr)
```

```
/tmp/5ZgPewTyMA.o
Enter size of array: 10
Enter elements in array: 10 -1 100 90 87 0 15 10 20 30
Source array before copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array before copying: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Source array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
```



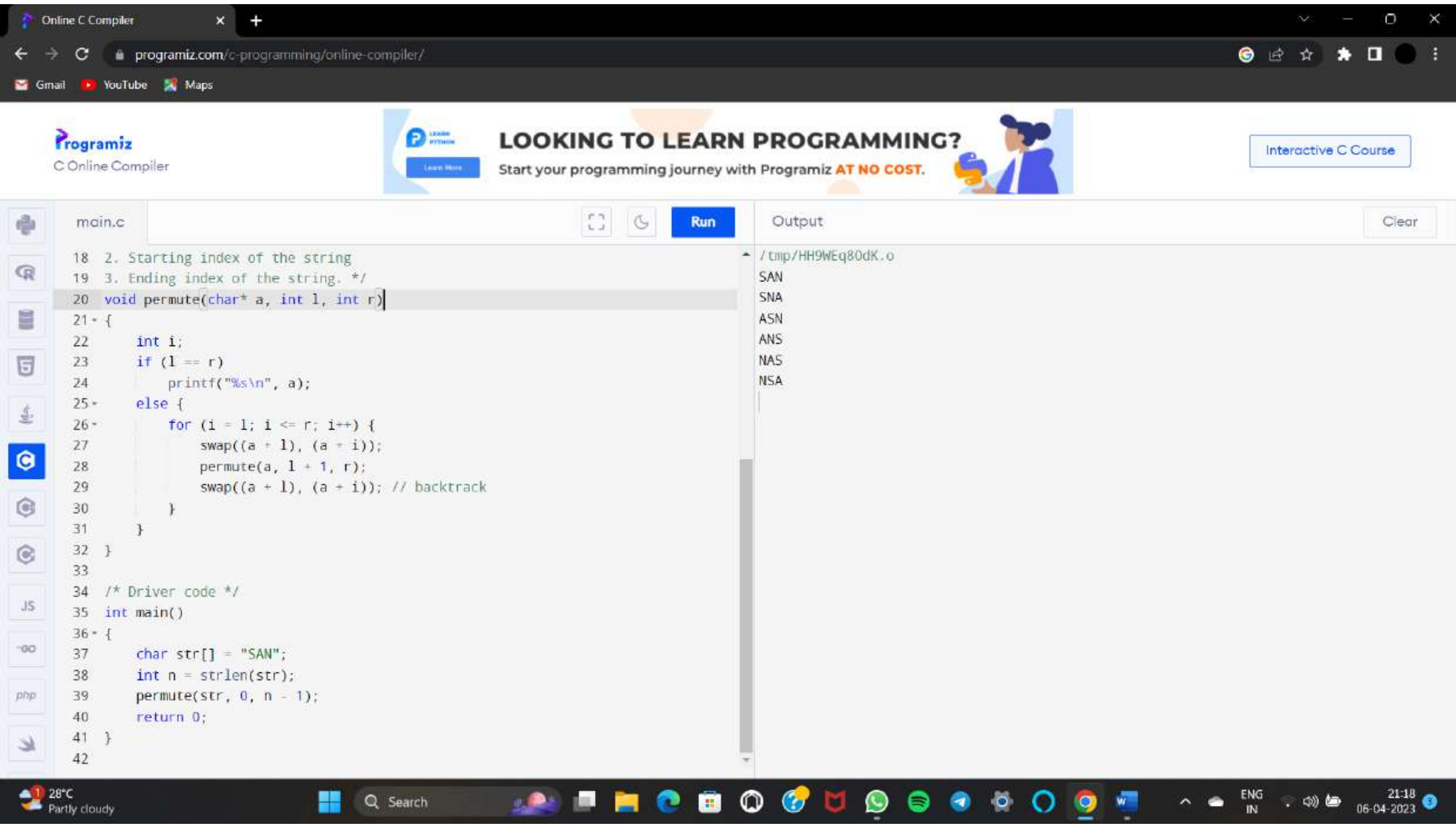
```c
        dest_ptr++;
    }


    /* Print source and destination array after copying */
    printf("\n\nSource array after copying: ");
    printArray(source_arr, size);

    printf("\nDestination array after copying: ");
    printArray(dest_arr, size);


    return 0;
}

void printArray(int *arr, int size)
{
    int i;

    for (i = 0; i < size; i++)
    {
        printf("%d, ", *(arr + i));
    }
}

```

```
/tmp/5ZgPewTyMA.o
Enter size of array: 10
Enter elements in array: 10 -1 100 90 87 0 15 10 20 30
Source array before copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array before copying: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

Source array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
Destination array after copying: 10, -1, 100, 90, 87, 0, 15, 10, 20, 30,
```



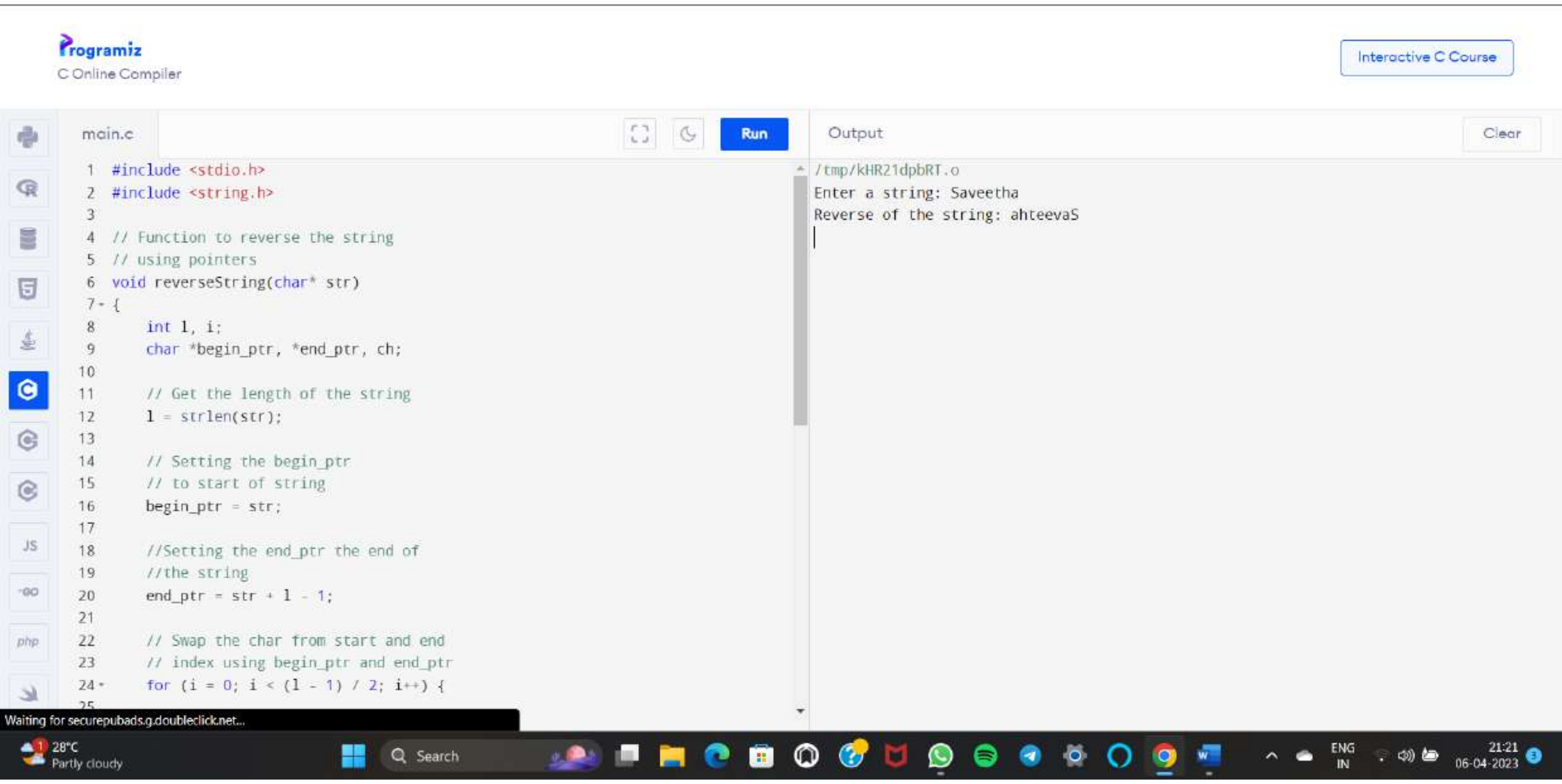
```c
// C program to print all permutations with duplicates
// allowed
#include <stdio.h>
#include <string.h>

/* Function to swap values at two pointers */
void swap(char* x, char* y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}

/* Function to print permutations of string
This function takes three parameters:
1. String
2. Starting index of the string
3. Ending index of the string. */
void permute(char* a, int l, int r)
{
    int i;
    if (l == r)
        printf("%s\n", a);
    else {
```

```
/tmp/HH9WEq80dK.o
SAN
SNA
ASN
ANS
NAS
NSA
```



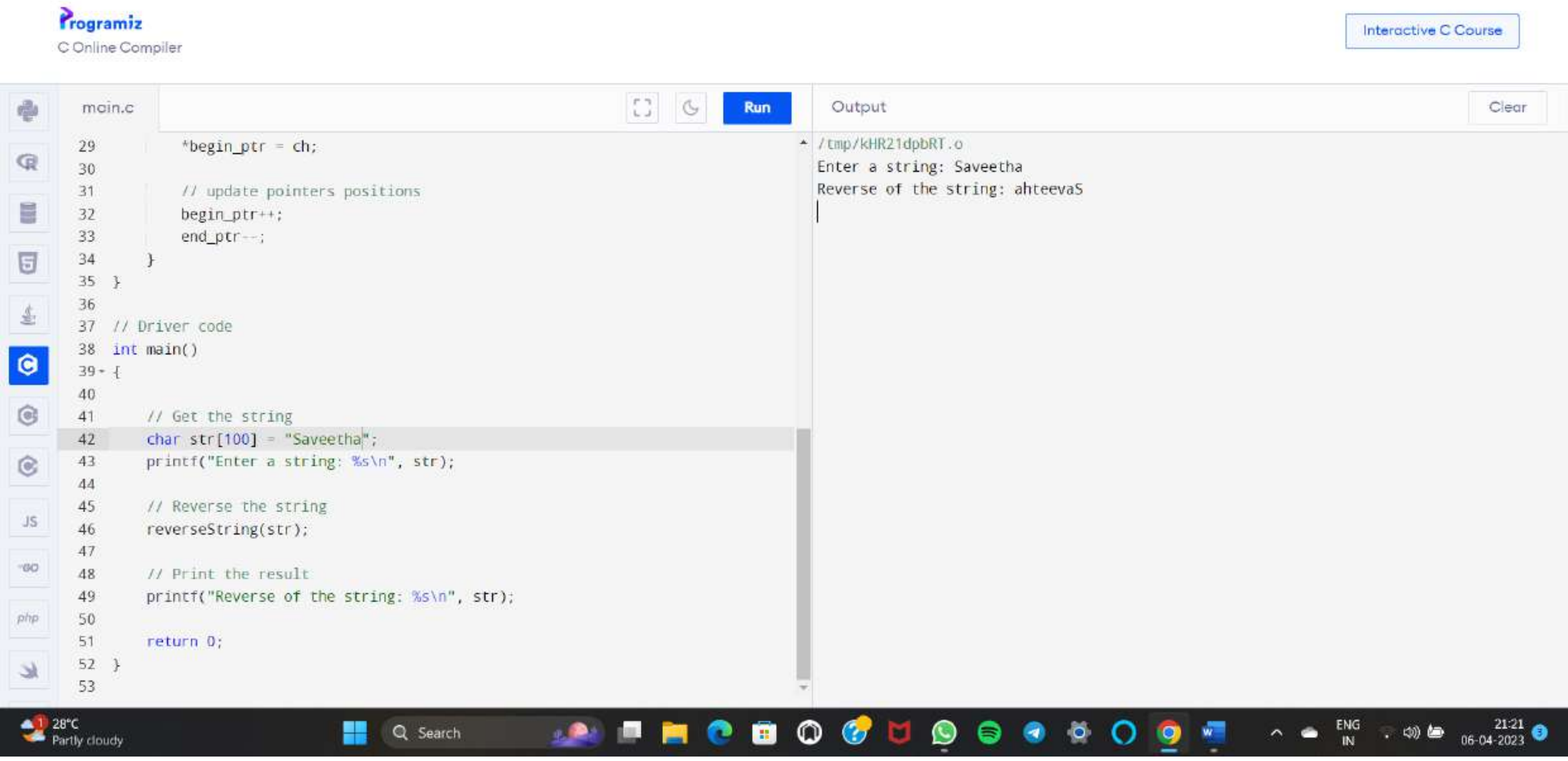
```c
2. Starting index of the string
3. Ending index of the string. */
void permute(char* a, int l, int r)
{
    int i;
    if (l == r)
        printf("%s\n", a);
    else {
        for (i = l; i <= r; i++) {
            swap((a + l), (a + i));
            permute(a, l + 1, r);
            swap((a + l), (a + i)); // backtrack
        }
    }
}

/* Driver code */
int main()
{
    char str[] = "SAN";
    int n = strlen(str);
    permute(str, 0, n - 1);
    return 0;
}
```

```
/tmp/HH9WEq80dK.o
SAN
SNA
ASN
ANS
NAS
NSA
```



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

// Function to reverse the string
// using pointers
void reverseString(char* str)
{
    int l, i;
    char *begin_ptr, *end_ptr, ch;

    // Get the length of the string
    l = strlen(str);

    // Setting the begin_ptr
    // to start of string
    begin_ptr = str;

    //Setting the end_ptr the end of
    //the string
    end_ptr = str + l - 1;

    // Swap the char from start and end
    // index using begin_ptr and end_ptr
    for (i = 0; i < (l - 1) / 2; i++) {
```

```
/tmp/kHR21dpbRT.o
Enter a string: Saveetha
Reverse of the string: ahteevaS
```



```c
        *begin_ptr = ch;

        // update pointers positions
        begin_ptr++;
        end_ptr--;
    }
}

// Driver code
int main()
{

    // Get the string
    char str[100] = "Saveetha";
    printf("Enter a string: %s\n", str);

    // Reverse the string
    reverseString(str);

    // Print the result
    printf("Reverse of the string: %s\n", str);

    return 0;
}
```

```
/tmp/kHR21dpbRT.o
Enter a string: Saveetha
Reverse of the string: ahteevaS
```