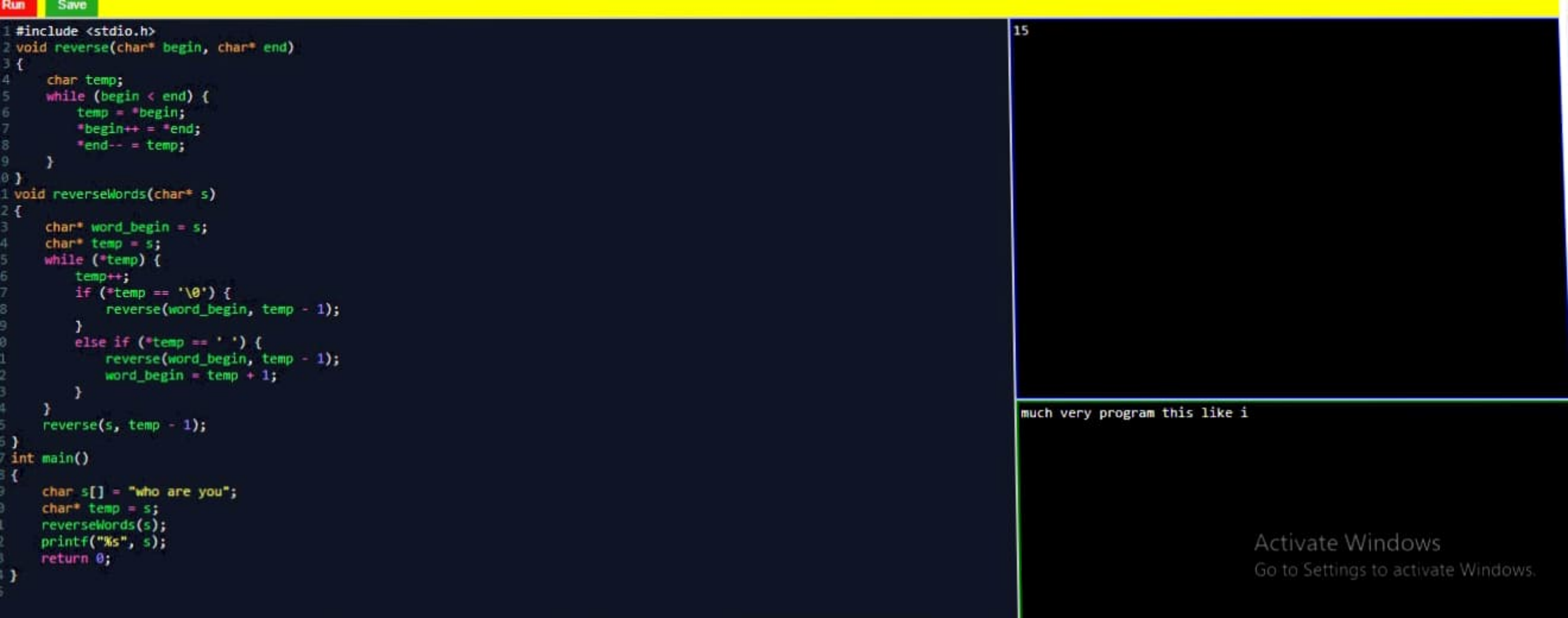
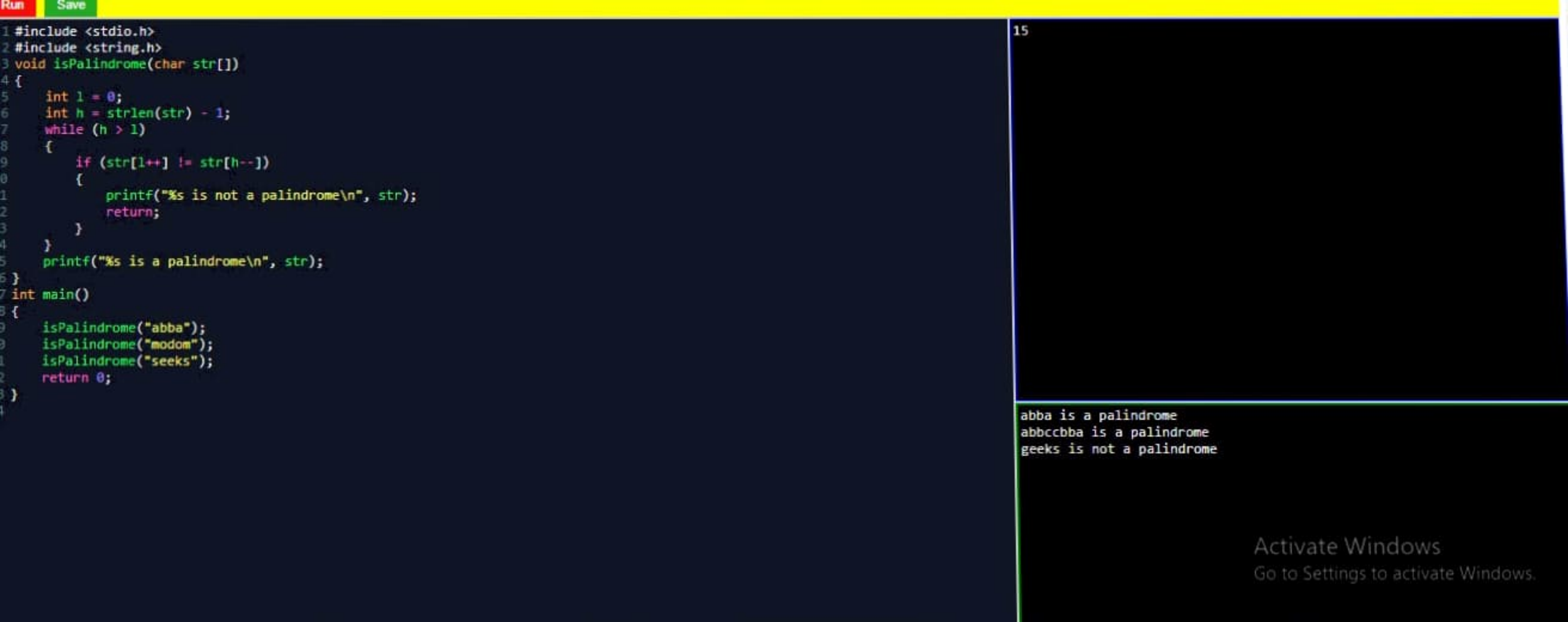


```c
#include <stdio.h>
void reverse(char* begin, char* end)
{
    char temp;
    while (begin < end) {
        temp = *begin;
        *begin++ = *end;
        *end-- = temp;
    }
}
void reverseWords(char* s)
{
    char* word_begin = s;
    char* temp = s;
    while (*temp) {
        temp++;
        if (*temp == '\0') {
            reverse(word_begin, temp - 1);
        }
        else if (*temp == ' ') {
            reverse(word_begin, temp - 1);
            word_begin = temp + 1;
        }
    }
    reverse(s, temp - 1);
}
int main()
{
    char s[] = "who are you";
    char* temp = s;
    reverseWords(s);
    printf("%s", s);
    return 0;
}
```

```
15
```

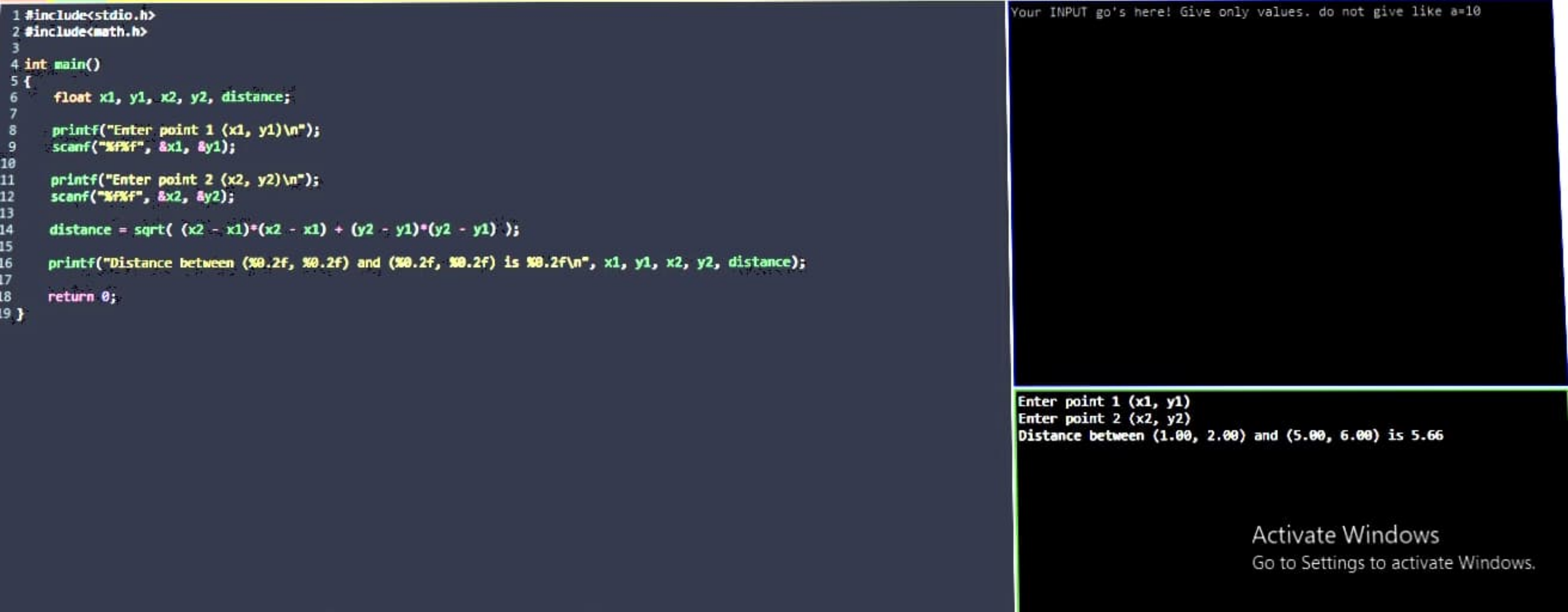
```
much very program this like i
```



```c
#include <stdio.h>
#include <string.h>
void isPalindrome(char str[])
{
    int l = 0;
    int h = strlen(str) - 1;
    while (h > l)
    {
        if (str[l++] != str[h--])
        {
            printf("%s is not a palindrome\n", str);
            return;
        }
    }
    printf("%s is a palindrome\n", str);
}
int main()
{
    isPalindrome("abba");
    isPalindrome("modom");
    isPalindrome("seeks");
    return 0;
}
```

```
15
```

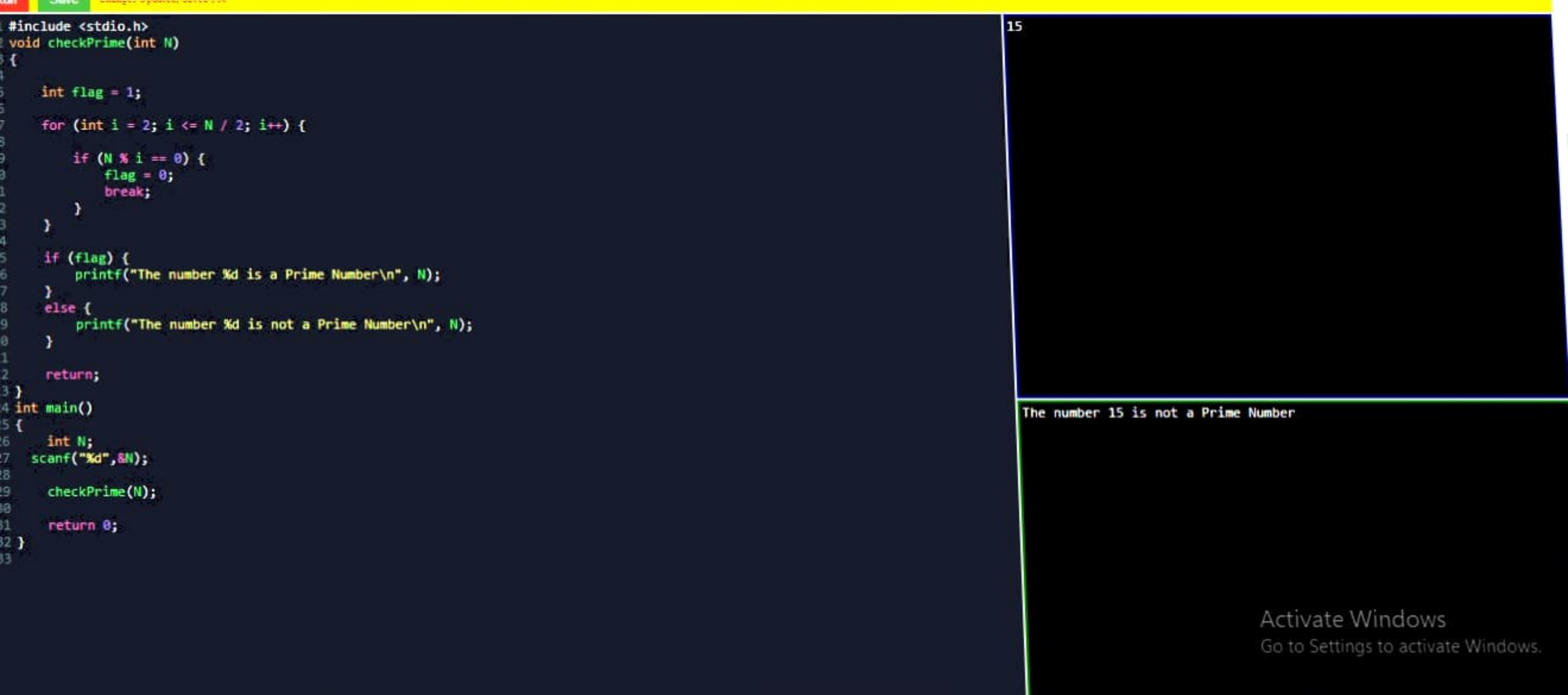
```
abba is a palindrome
abbccbba is a palindrome
geeks is not a palindrome
```



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

int main()
{
    float x1, y1, x2, y2, distance;

    printf("Enter point 1 (x1, y1)\n");
    scanf("%f%f", &x1, &y1);

    printf("Enter point 2 (x2, y2)\n");
    scanf("%f%f", &x2, &y2);

    distance = sqrt( (x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1) );

    printf("Distance between (%0.2f, %0.2f) and (%0.2f, %0.2f) is %0.2f\n", x1, y1, x2, y2, distance);

    return 0;
}
```

Your INPUT go's here! Give only values. do not give like a=10

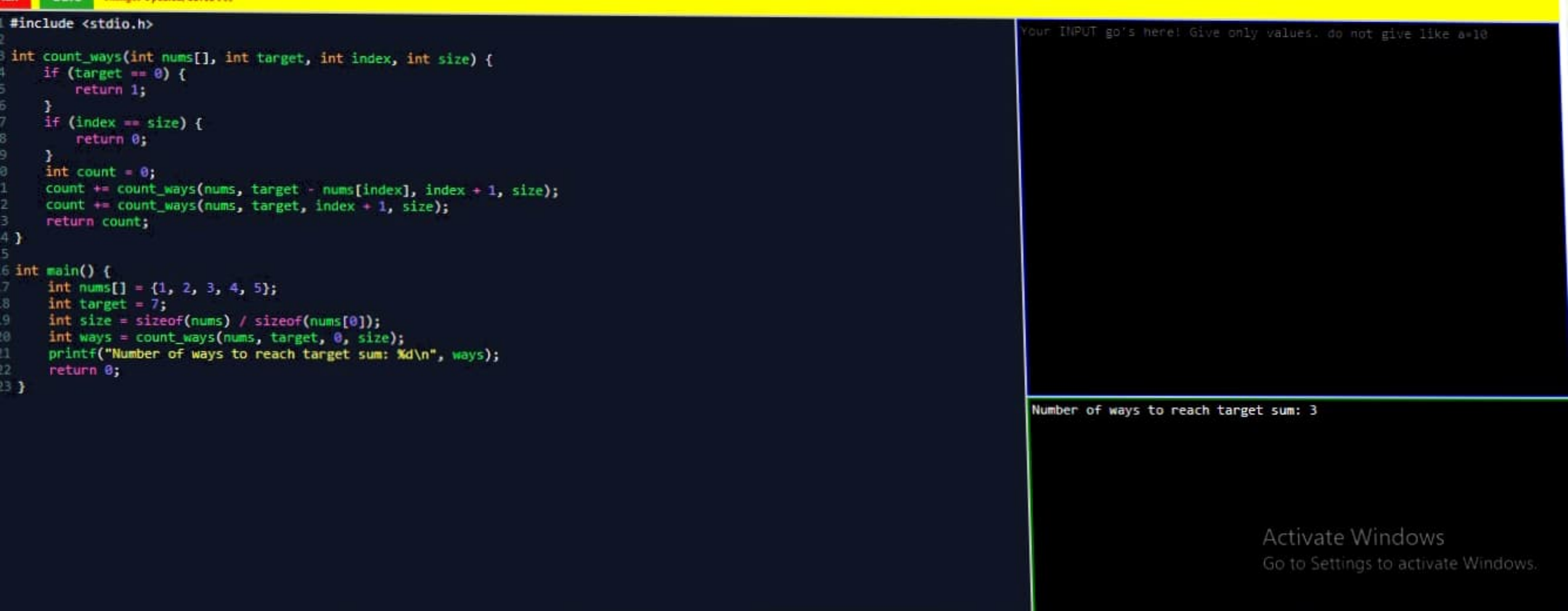
```
Enter point 1 (x1, y1)
Enter point 2 (x2, y2)
Distance between (1.00, 2.00) and (5.00, 6.00) is 5.66
```



```c
#include <stdio.h>
void checkPrime(int N)
{

    int flag = 1;

    for (int i = 2; i <= N / 2; i++) {

        if (N % i == 0) {
            flag = 0;
            break;
        }
    }

    if (flag) {
        printf("The number %d is a Prime Number\n", N);
    }
    else {
        printf("The number %d is not a Prime Number\n", N);
    }

    return;
}
int main()
{
    int N;
  scanf("%d",&N);

    checkPrime(N);

    return 0;
}
```

```
15
```

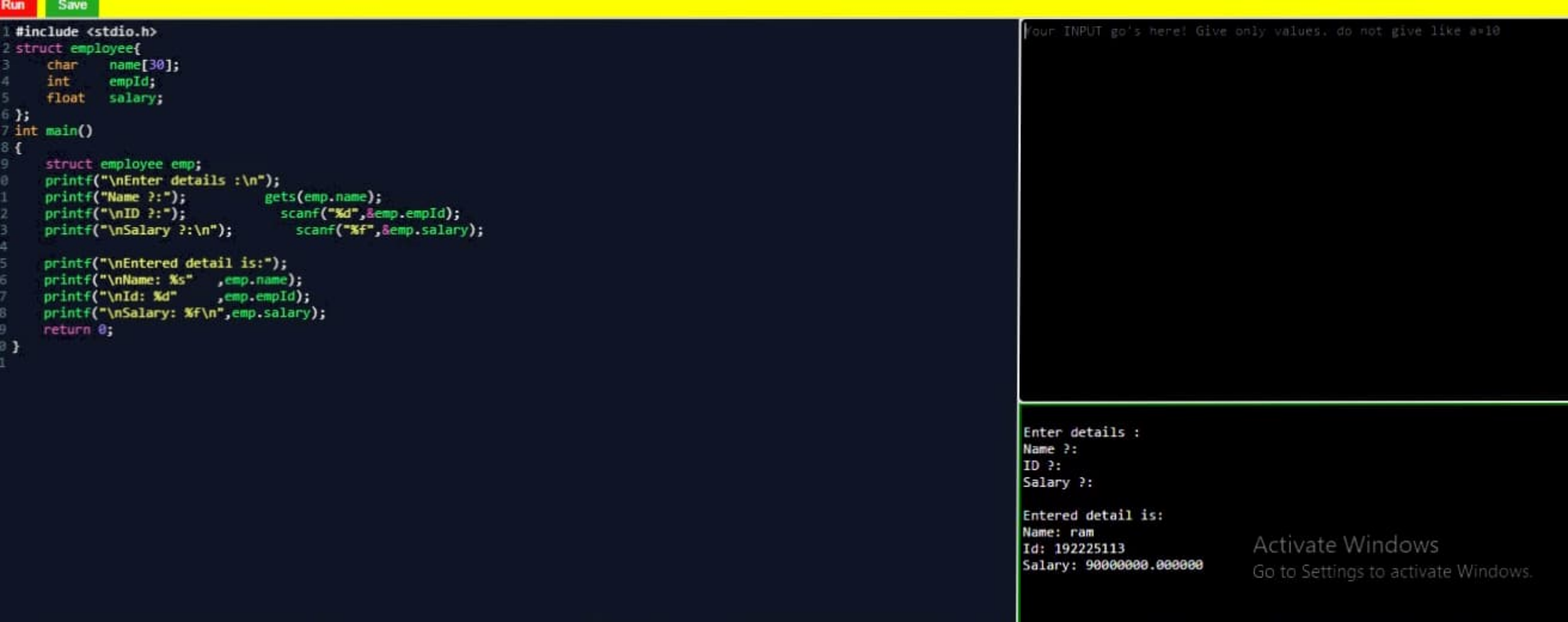
```
The number 15 is not a Prime Number
```



```c
#include <stdio.h>

int count_ways(int nums[], int target, int index, int size) {
    if (target == 0) {
        return 1;
    }
    if (index == size) {
        return 0;
    }
    int count = 0;
    count += count_ways(nums, target - nums[index], index + 1, size);
    count += count_ways(nums, target, index + 1, size);
    return count;
}

int main() {
    int nums[] = {1, 2, 3, 4, 5};
    int target = 7;
    int size = sizeof(nums) / sizeof(nums[0]);
    int ways = count_ways(nums, target, 0, size);
    printf("Number of ways to reach target sum: %d\n", ways);
    return 0;
}
```

Your INPUT go's here! Give only values. do not give like a=10

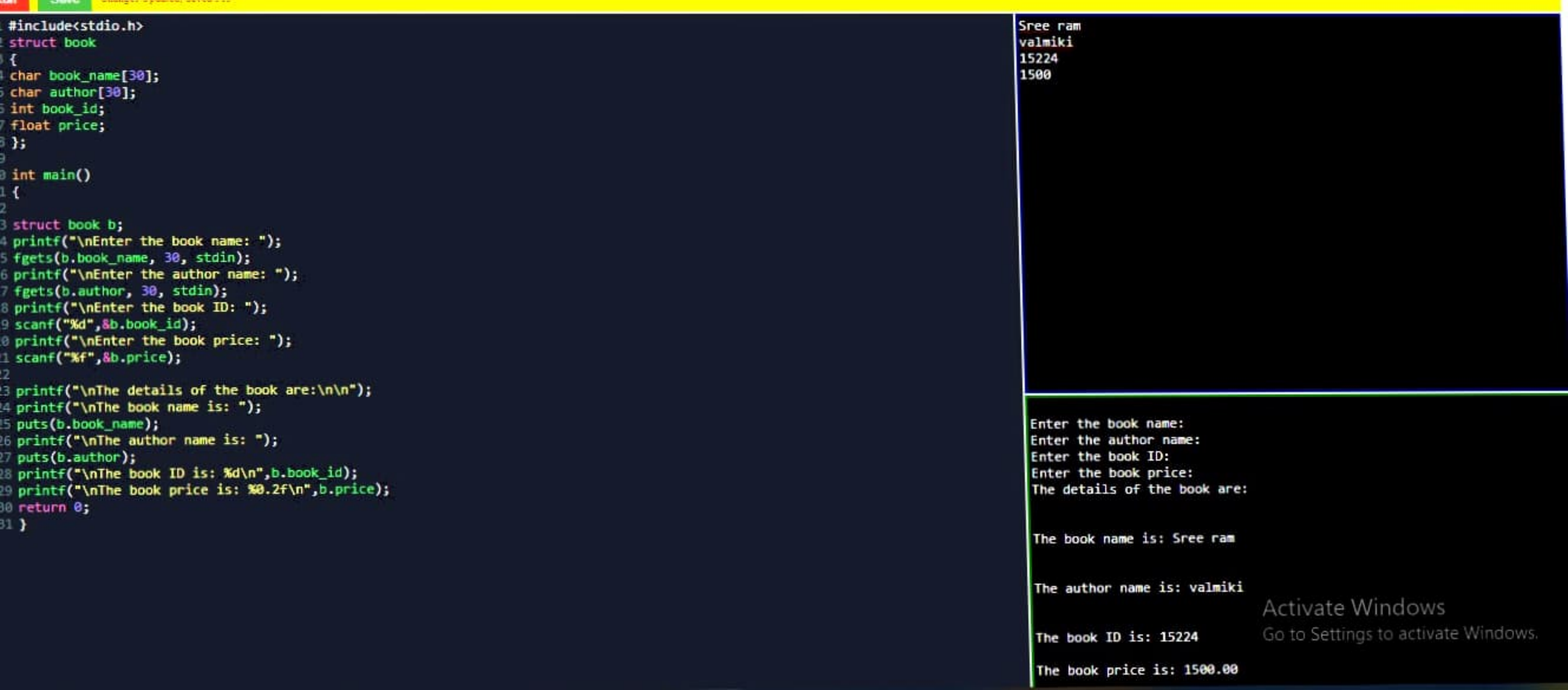
```
Number of ways to reach target sum: 3
```



Run
Save
Your INPUT go's here! Give only values. do not give like a=10

```c
#include <stdio.h>
struct employee{
    char    name[30];
    int     empId;
    float   salary;
};
int main()
{
    struct employee emp;
    printf("\nEnter details :\n");
    printf("Name ?:");          gets(emp.name);
    printf("\nID ?:");            scanf("%d",&emp.empId);
    printf("\nSalary ?:\n");       scanf("%f",&emp.salary);

    printf("\nEntered detail is:");
    printf("\nName: %s"   ,emp.name);
    printf("\nId: %d"     ,emp.empId);
    printf("\nSalary: %f\n",emp.salary);
    return 0;
}
```

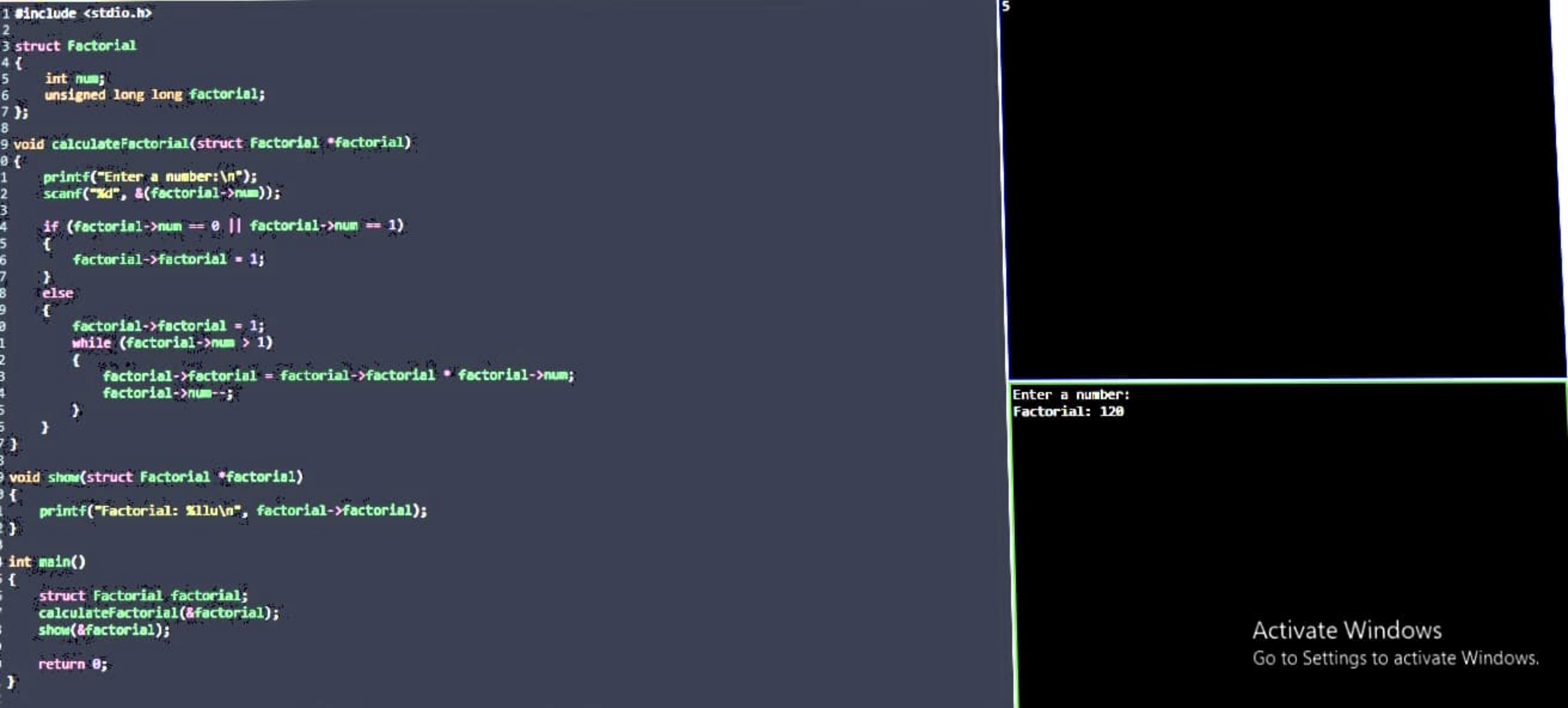
```

Enter details :
Name ?:
ID ?:
Salary ?:

Entered detail is:
Name: ram
Id: 192225113
Salary: 90000000.000000
```



```c
#include<stdio.h>
struct book
{
char book_name[30];
char author[30];
int book_id;
float price;
};

int main()
{

struct book b;
printf("\nEnter the book name: ");
fgets(b.book_name, 30, stdin);
printf("\nEnter the author name: ");
fgets(b.author, 30, stdin);
printf("\nEnter the book ID: ");
scanf("%d",&b.book_id);
printf("\nEnter the book price: ");
scanf("%f",&b.price);

printf("\nThe details of the book are:\n\n");
printf("\nThe book name is: ");
puts(b.book_name);
printf("\nThe author name is: ");
puts(b.author);
printf("\nThe book ID is: %d\n",b.book_id);
printf("\nThe book price is: %0.2f\n",b.price);
return 0;
}
```

```
Sree ram
valmiki
15224
1500
```

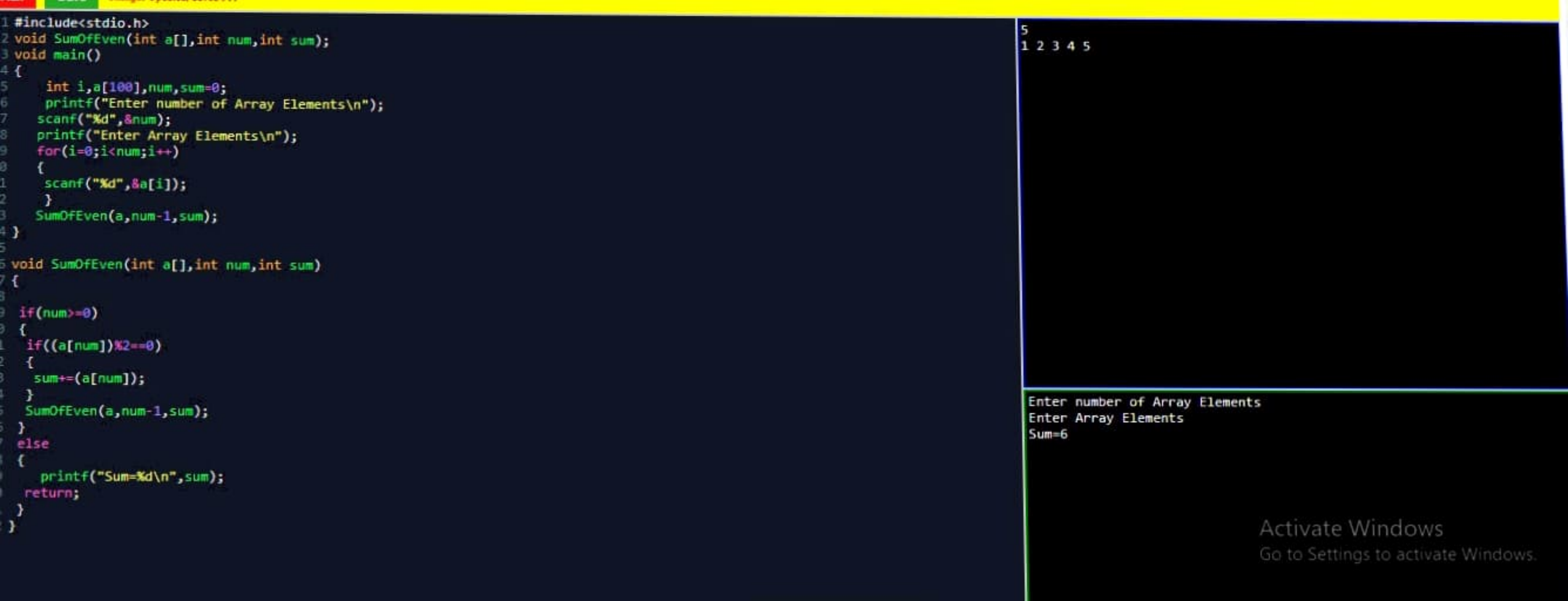
```
Enter the book name:
Enter the author name:
Enter the book ID:
Enter the book price:
The details of the book are:


The book name is: Sree ram


The author name is: valmiki


The book ID is: 15224

The book price is: 1500.00
```



```c
#include <stdio.h>

struct Factorial
{
    int num;
    unsigned long long factorial;
};

void calculateFactorial(struct Factorial *factorial)
{
    printf("Enter a number:\n");
    scanf("%d", &(factorial->num));

    if (factorial->num == 0 || factorial->num == 1)
    {
        factorial->factorial = 1;
    }
    else
    {
        factorial->factorial = 1;
        while (factorial->num > 1)
        {
            factorial->factorial = factorial->factorial * factorial->num;
            factorial->num--;
        }
    }
}

void show(struct Factorial *factorial)
{
    printf("Factorial: %llu\n", factorial->factorial);
}

int main()
{
    struct Factorial factorial;
    calculateFactorial(&factorial);
    show(&factorial);

    return 0;
}
```

```
5
```

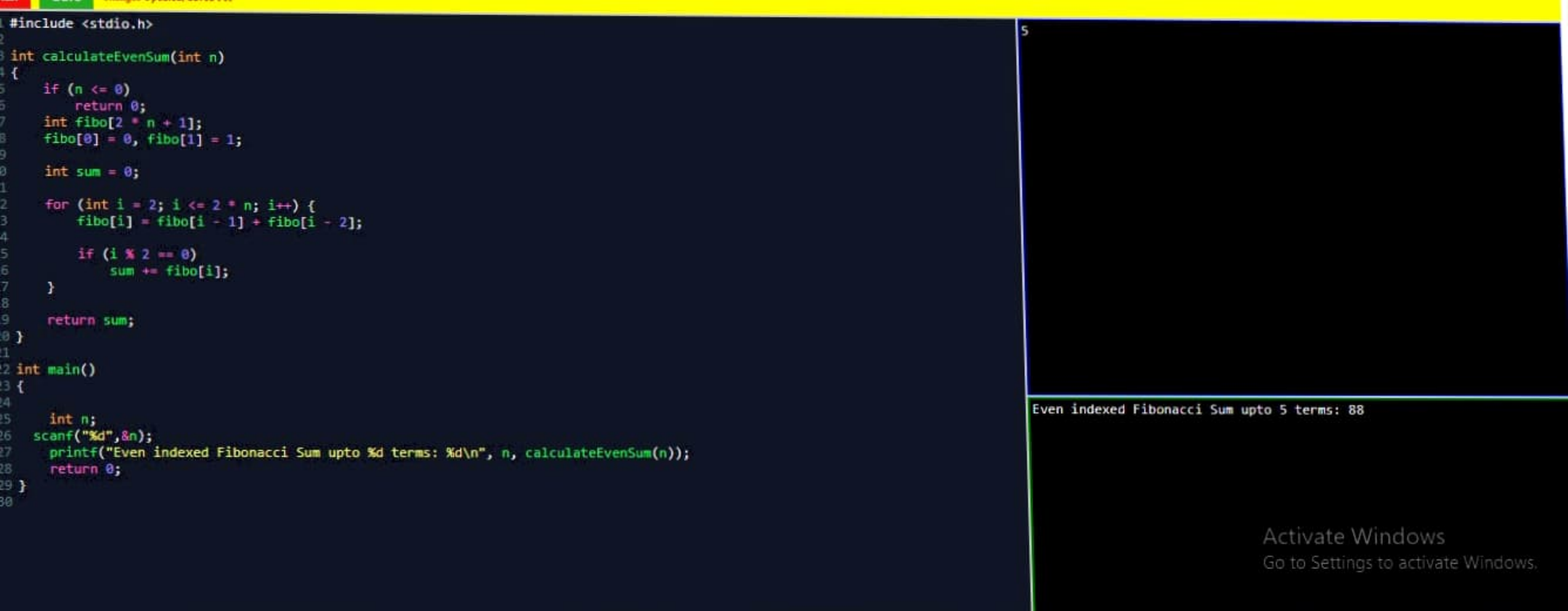
```
Enter a number:
Factorial: 120
```



```c
#include<stdio.h>
void SumOfEven(int a[],int num,int sum);
void main()
{
     int i,a[100],num,sum=0;
     printf("Enter number of Array Elements\n");
    scanf("%d",&num);
    printf("Enter Array Elements\n");
    for(i=0;i<num;i++)
    {
     scanf("%d",&a[i]);
     }
    SumOfEven(a,num-1,sum);
}

void SumOfEven(int a[],int num,int sum)
{

 if(num>=0)
 {
  if((a[num])%2==0)
  {
   sum+=(a[num]);
  }
  SumOfEven(a,num-1,sum);
 }
 else
 {
    printf("Sum=%d\n",sum);
  return;
 }
}
```

```
5
1 2 3 4 5
```

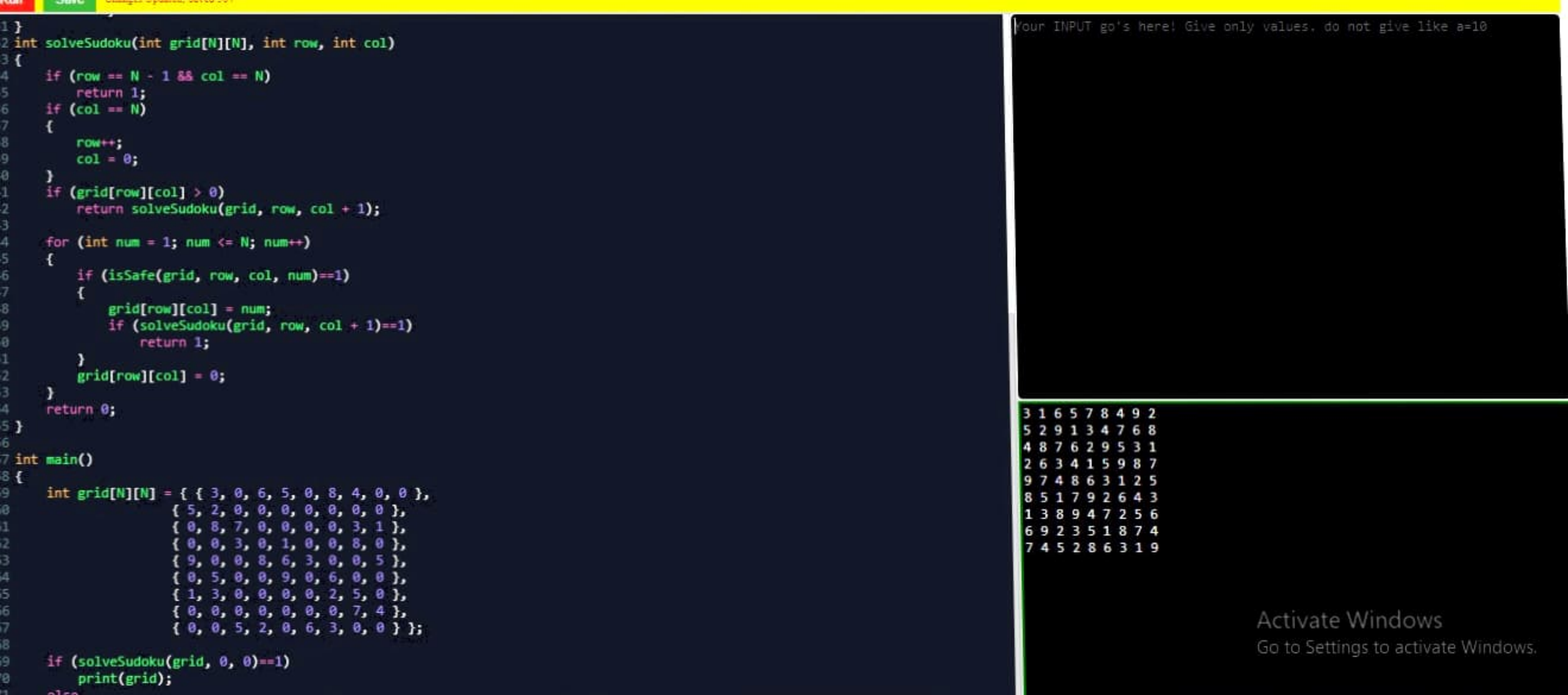
```
Enter number of Array Elements
Enter Array Elements
Sum=6
```



```c
#include <stdio.h>

int calculateEvenSum(int n)
{
    if (n <= 0)
        return 0;
    int fibo[2 * n + 1];
    fibo[0] = 0, fibo[1] = 1;

    int sum = 0;

    for (int i = 2; i <= 2 * n; i++) {
        fibo[i] = fibo[i - 1] + fibo[i - 2];

        if (i % 2 == 0)
            sum += fibo[i];
    }

    return sum;
}

int main()
{

    int n;
  scanf("%d",&n);
    printf("Even indexed Fibonacci Sum upto %d terms: %d\n", n, calculateEvenSum(n));
    return 0;
}

```

```
5
```

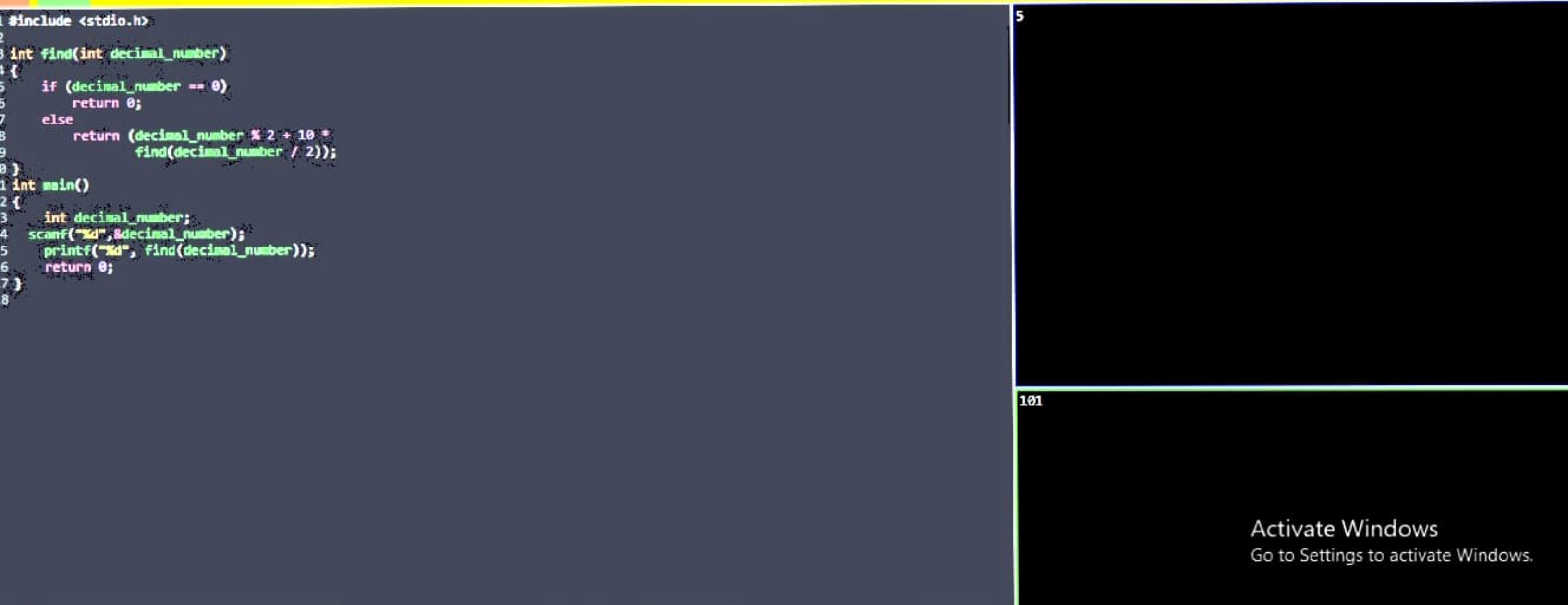
```
Even indexed Fibonacci Sum upto 5 terms: 88
```



```
}
int solveSudoku(int grid[N][N], int row, int col)
{
    if (row == N - 1 && col == N)
        return 1;
    if (col == N)
    {
        row++;
        col = 0;
    }
    if (grid[row][col] > 0)
        return solveSudoku(grid, row, col + 1);

    for (int num = 1; num <= N; num++)
    {
        if (isSafe(grid, row, col, num)==1)
        {
            grid[row][col] = num;
            if (solveSudoku(grid, row, col + 1)==1)
                return 1;
        }
        grid[row][col] = 0;
    }
    return 0;
}

int main()
{
    int grid[N][N] = { { 3, 0, 6, 5, 0, 8, 4, 0, 0 },
                       { 5, 2, 0, 0, 0, 0, 0, 0, 0 },
                       { 0, 8, 7, 0, 0, 0, 0, 3, 1 },
                       { 0, 0, 3, 0, 1, 0, 0, 8, 0 },
                       { 9, 0, 0, 8, 6, 3, 0, 0, 5 },
                       { 0, 5, 0, 0, 9, 0, 6, 0, 0 },
                       { 1, 3, 0, 0, 0, 0, 2, 5, 0 },
                       { 0, 0, 0, 0, 0, 0, 0, 7, 4 },
                       { 0, 0, 5, 2, 0, 6, 3, 0, 0 } };

    if (solveSudoku(grid, 0, 0)==1)
        print(grid);
    else
```

Your INPUT go's here! Give only values. do not give like a=10

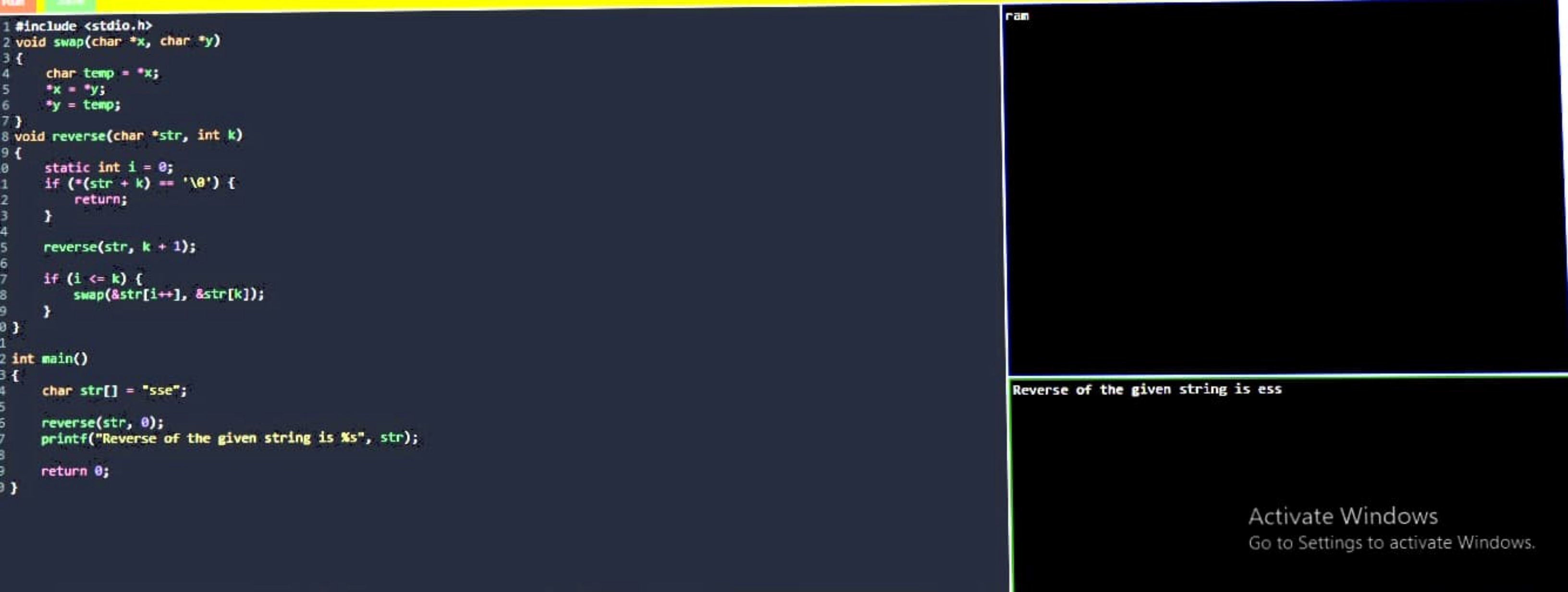
```
3 1 6 5 7 8 4 9 2
5 2 9 1 3 4 7 6 8
4 8 7 6 2 9 5 3 1
2 6 3 4 1 5 9 8 7
9 7 4 8 6 3 1 2 5
8 5 1 7 9 2 6 4 3
1 3 8 9 4 7 2 5 6
6 9 2 3 5 1 8 7 4
7 4 5 2 8 6 3 1 9
```



```c
#include <stdio.h>

int find(int decimal_number)
{
    if (decimal_number == 0)
        return 0;
    else
        return (decimal_number % 2 + 10 *
                find(decimal_number / 2));
}
int main()
{
    int decimal_number;
  scanf("%d",&decimal_number);
    printf("%d", find(decimal_number));
    return 0;
}
```

```
5
```

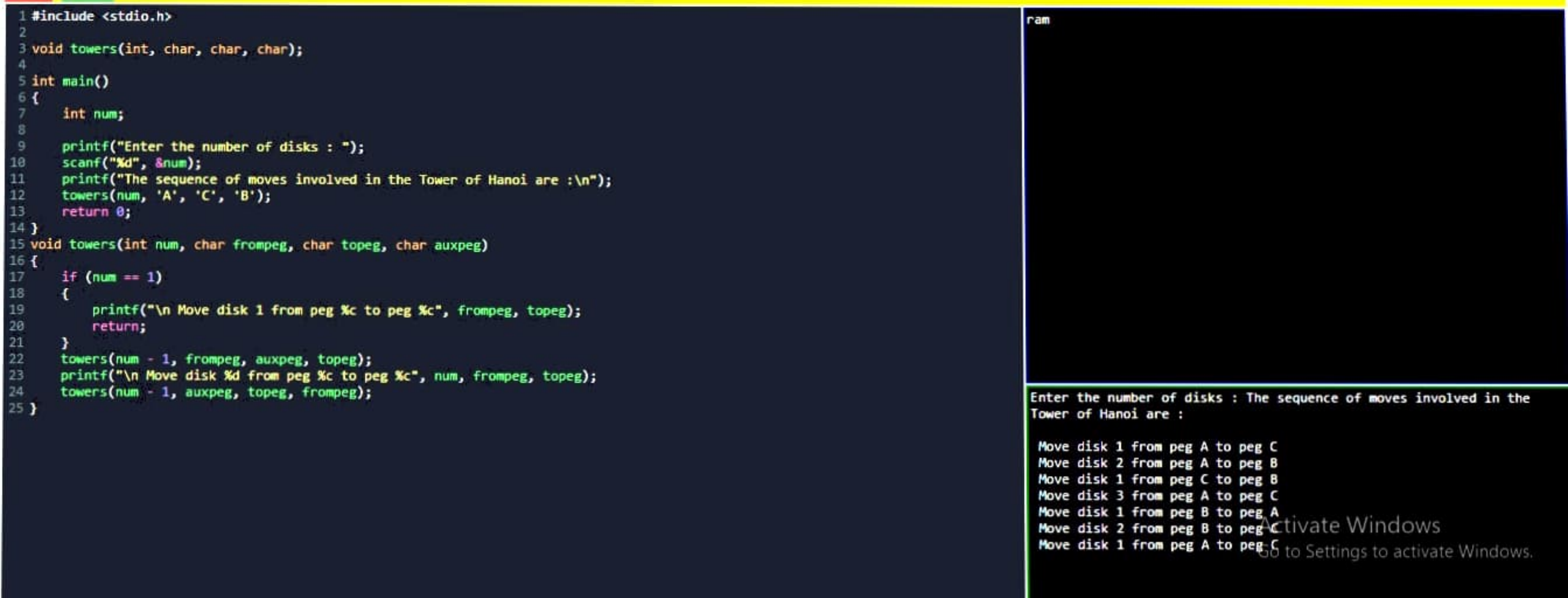
```
101
```



```c
#include <stdio.h>
void swap(char *x, char *y)
{
    char temp = *x;
    *x = *y;
    *y = temp;
}
void reverse(char *str, int k)
{
    static int i = 0;
    if (*(str + k) == '\0') {
        return;
    }

    reverse(str, k + 1);

    if (i <= k) {
        swap(&str[i++], &str[k]);
    }
}

int main()
{
    char str[] = "sse";

    reverse(str, 0);
    printf("Reverse of the given string is %s", str);

    return 0;
}
```

```
ram
```

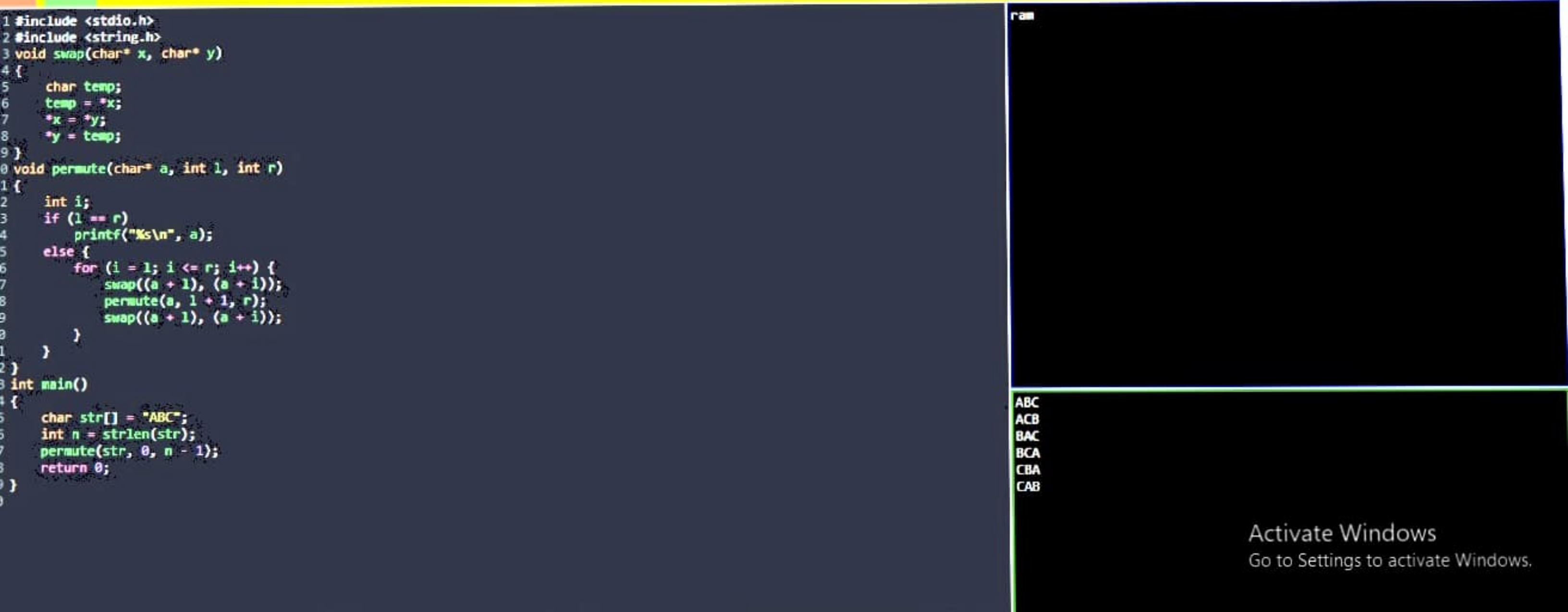
```
Reverse of the given string is ess
```



```c
#include <stdio.h>

void towers(int, char, char, char);

int main()
{
    int num;

    printf("Enter the number of disks : ");
    scanf("%d", &num);
    printf("The sequence of moves involved in the Tower of Hanoi are :\n");
    towers(num, 'A', 'C', 'B');
    return 0;
}
void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);
        return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}
```

```
ram
```

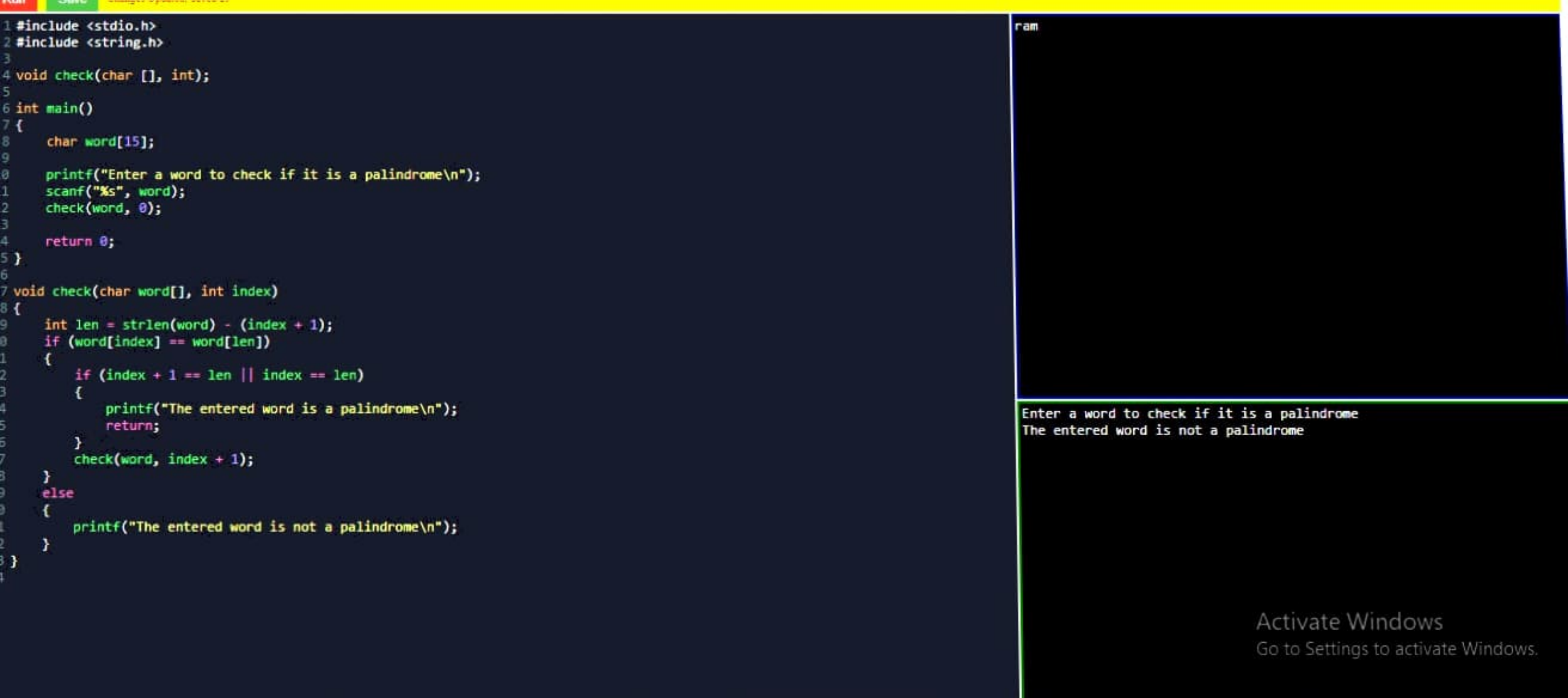
```
Enter the number of disks : The sequence of moves involved in the
Tower of Hanoi are :

 Move disk 1 from peg A to peg C
 Move disk 2 from peg A to peg B
 Move disk 1 from peg C to peg B
 Move disk 3 from peg A to peg C
 Move disk 1 from peg B to peg A
 Move disk 2 from peg B to peg C
 Move disk 1 from peg A to peg C
```



```c
#include <stdio.h>
#include <string.h>
void swap(char* x, char* y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
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));
        }
    }
}
int main()
{
    char str[] = "ABC";
    int n = strlen(str);
    permute(str, 0, n - 1);
    return 0;
}
```

```
ram
```

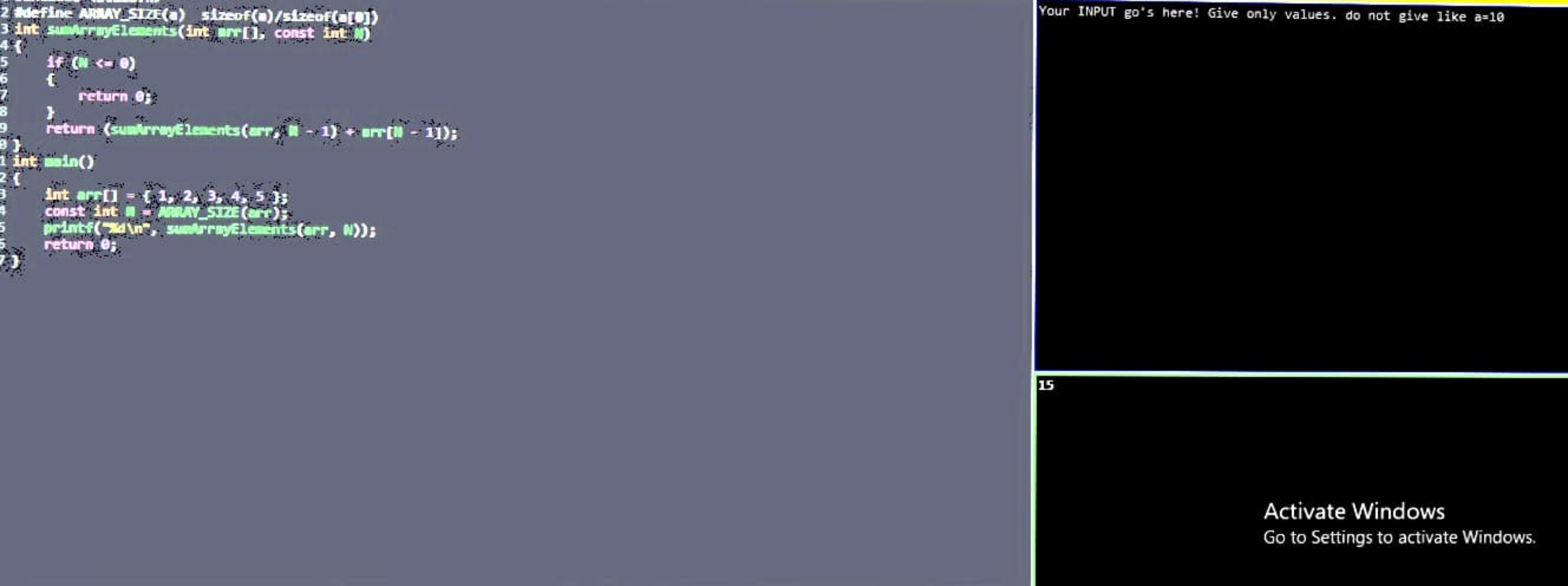
```
ABC
ACB
BAC
BCA
CBA
CAB
```



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

void check(char [], int);

int main()
{
    char word[15];

    printf("Enter a word to check if it is a palindrome\n");
    scanf("%s", word);
    check(word, 0);

    return 0;
}

void check(char word[], int index)
{
    int len = strlen(word) - (index + 1);
    if (word[index] == word[len])
    {
        if (index + 1 == len || index == len)
        {
            printf("The entered word is a palindrome\n");
            return;
        }
        check(word, index + 1);
    }
    else
    {
        printf("The entered word is not a palindrome\n");
    }
}
```

```
ram
```

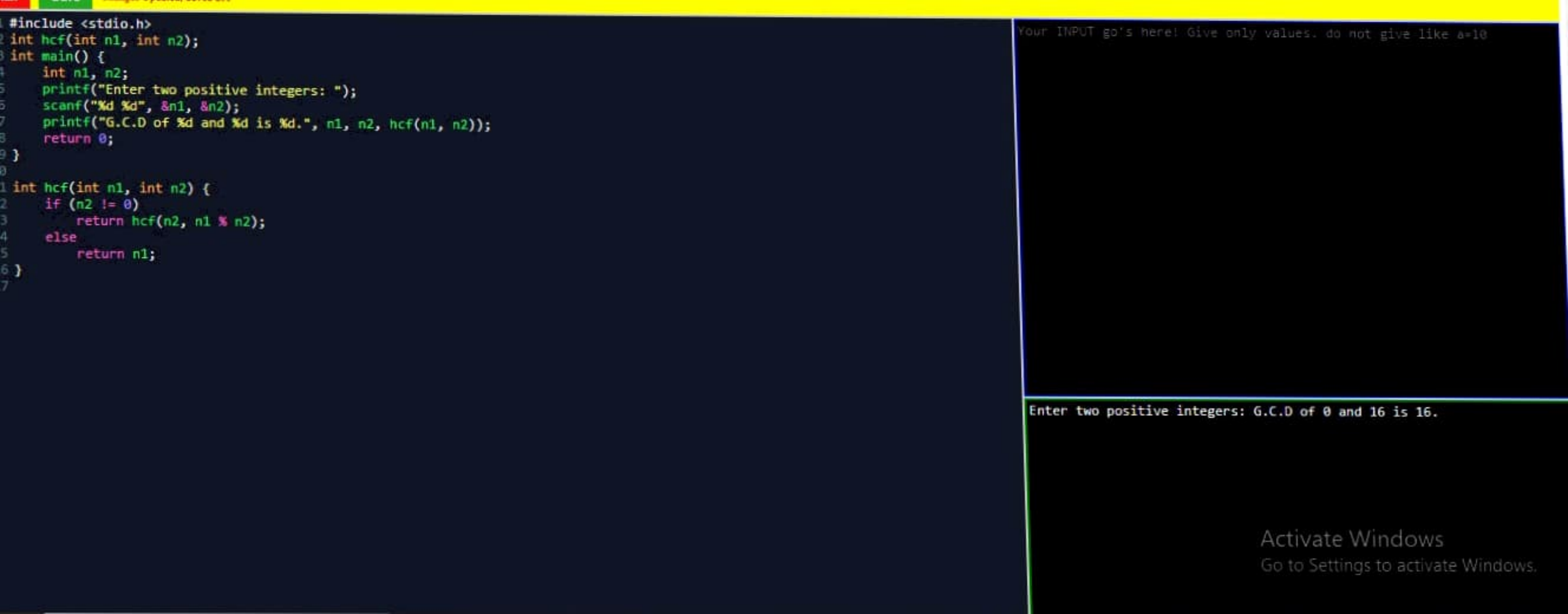
```
Enter a word to check if it is a palindrome
The entered word is not a palindrome
```



```c
#define ARRAY_SIZE(a)  sizeof(a)/sizeof(a[0])
int sumArrayElements(int arr[], const int N)
{
    if (N <= 0)
    {
        return 0;
    }
    return (sumArrayElements(arr, N - 1) + arr[N - 1]);
}
int main()
{
    int arr[] = { 1, 2, 3, 4, 5 };
    const int N = ARRAY_SIZE(arr);
    printf("%d\n", sumArrayElements(arr, N));
    return 0;
}
```

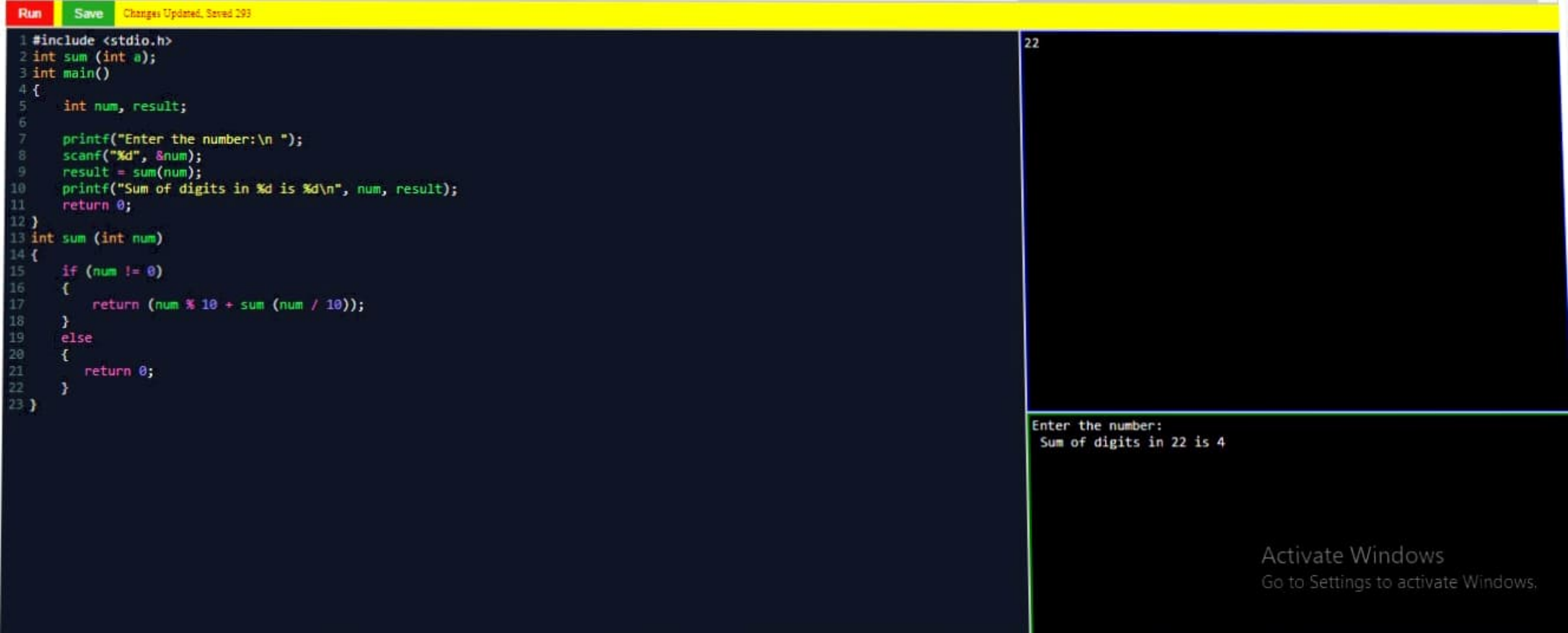
```
Your INPUT go's here! Give only values. do not give like a=10
```

```
15
```


Your INPUT go's here! Give only values. do not give like a=10

```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;
}
```

```
Enter two positive integers: G.C.D of 0 and 16 is 16.
```

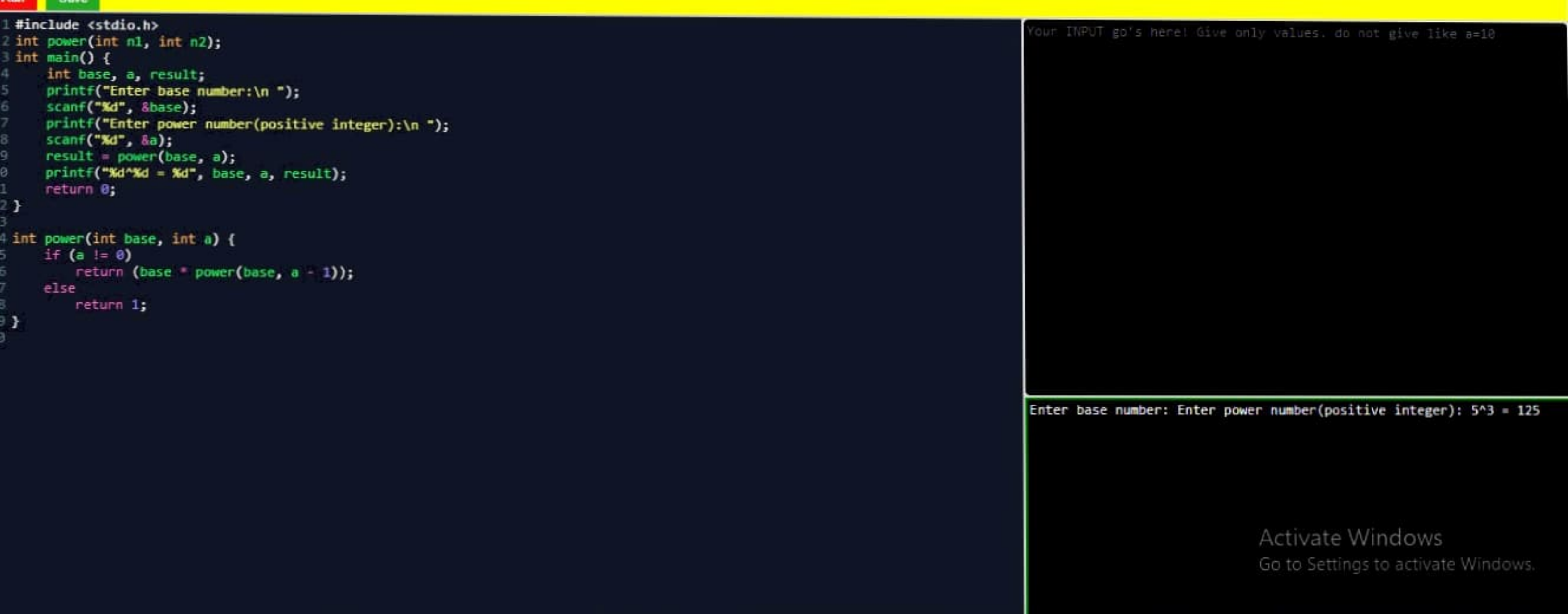


Changes Updated, Saved 293

```c
#include <stdio.h>
int sum (int a);
int main()
{
    int num, result;

    printf("Enter the number:\n ");
    scanf("%d", &num);
    result = sum(num);
    printf("Sum of digits in %d is %d\n", num, result);
    return 0;
}
int sum (int num)
{
    if (num != 0)
    {
        return (num % 10 + sum (num / 10));
    }
    else
    {
       return 0;
    }
}
```

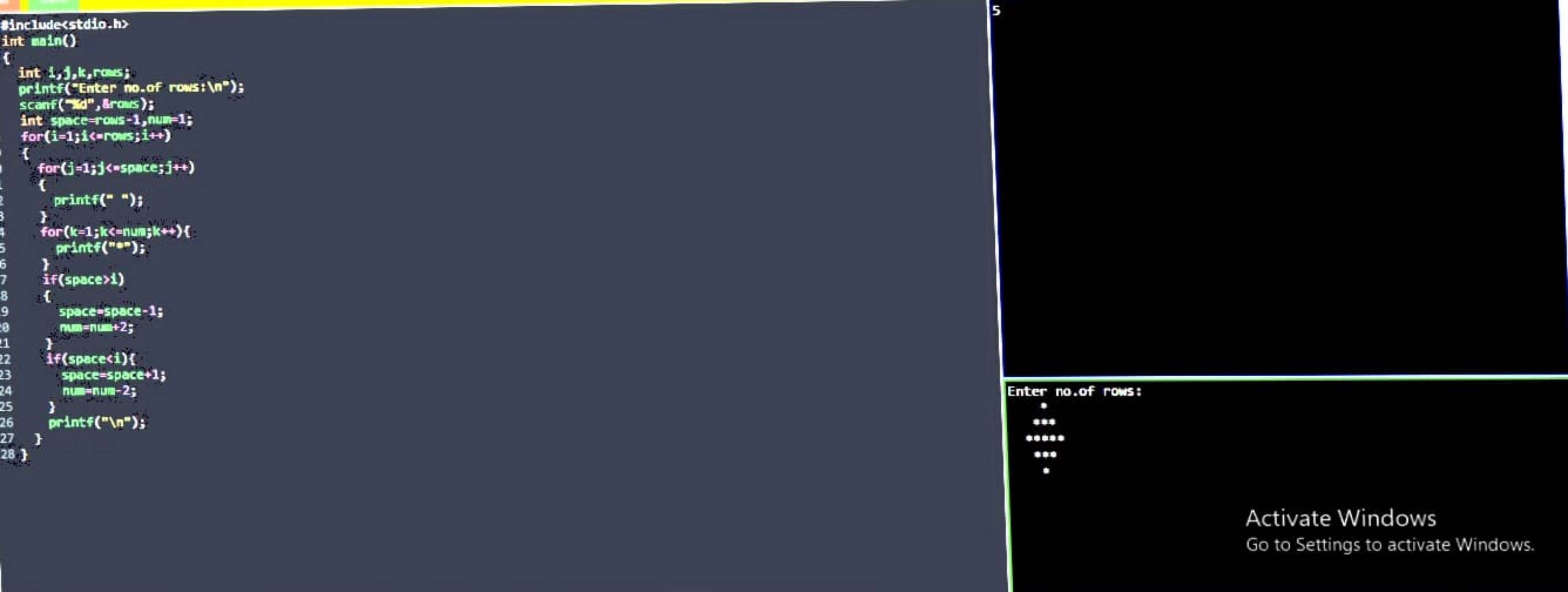
```
22
```

```
Enter the number:
 Sum of digits in 22 is 4
```


Run
Save
Your INPUT go's here! Give only values. do not give like a=10
Activate Windows
Go to Settings to activate Windows.

```c
#include <stdio.h>
int power(int n1, int n2);
int main() {
    int base, a, result;
    printf("Enter base number:\n ");
    scanf("%d", &base);
    printf("Enter power number(positive integer):\n ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d^%d = %d", base, a, result);
    return 0;
}

int power(int base, int a) {
    if (a != 0)
        return (base * power(base, a - 1));
    else
        return 1;
}
```

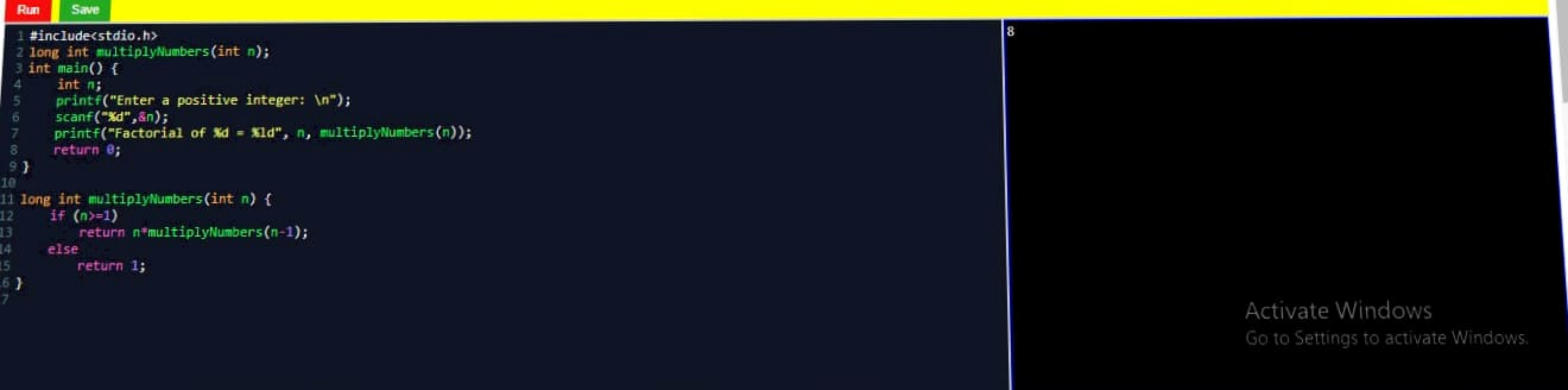
```
Enter base number: Enter power number(positive integer): 5^3 = 125
```



```
#include<stdio.h>
int main()
{
  int i,j,k,rows;
  printf("Enter no.of rows:\n");
  scanf("%d",&rows);
  int space=rows-1,num=1;
  for(i=1;i<=rows;i++)
  {
    for(j=1;j<=space;j++)
    {
      printf(" ");
    }
    for(k=1;k<=num;k++){
      printf("*");
    }
    if(space>i)
    {
      space=space-1;
      num=num+2;
    }
    if(space<i){
      space=space+1;
      num=num-2;
    }
    printf("\n");
  }
}
```

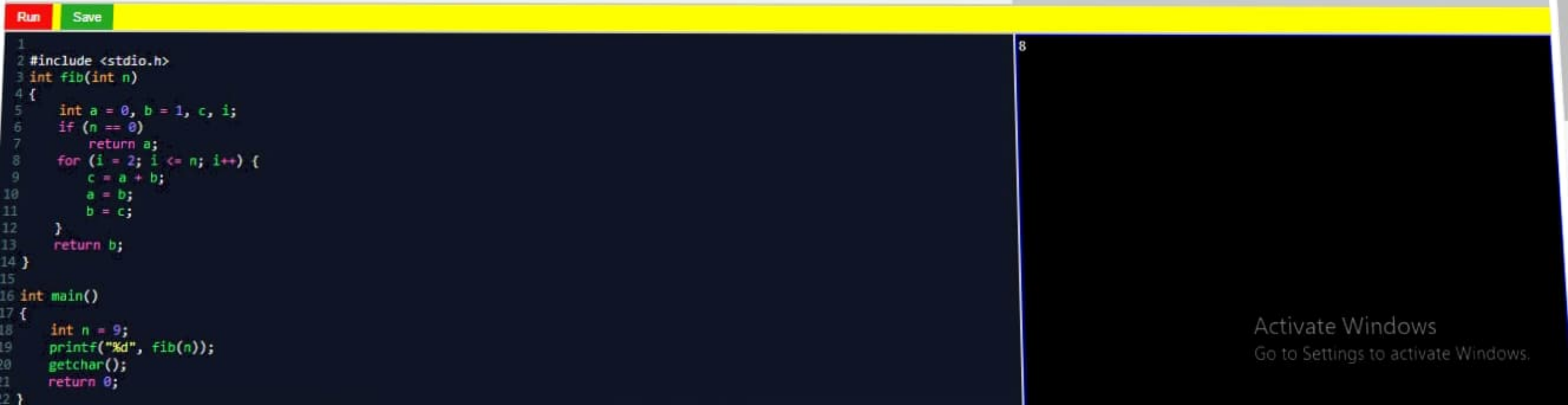
```
5
```

```
Enter no.of rows:
    *
   ***
  *****
   ***
    *
```



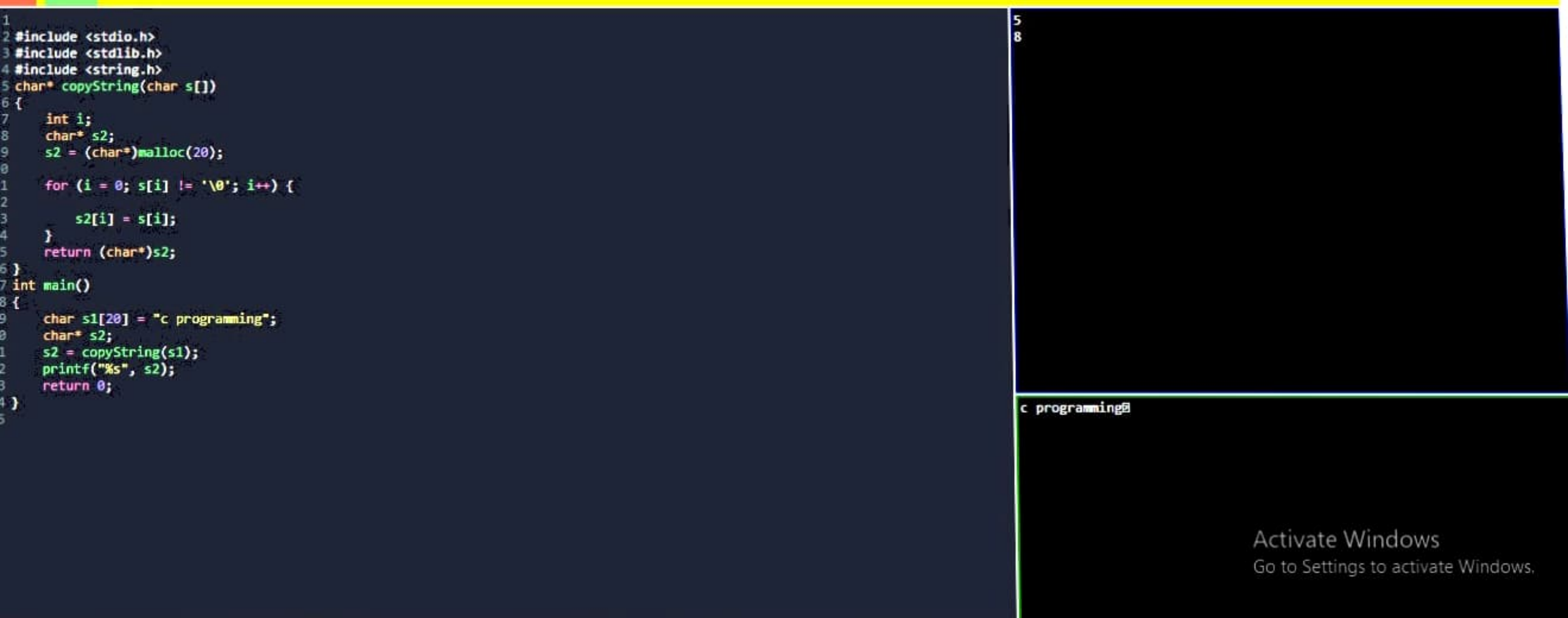
```c
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: \n");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

```
8
```



```c

#include <stdio.h>
int fib(int n)
{
    int a = 0, b = 1, c, i;
    if (n == 0)
        return a;
    for (i = 2; i <= n; i++) {
        c = a + b;
        a = b;
        b = c;
    }
    return b;
}

int main()
{
    int n = 9;
    printf("%d", fib(n));
    getchar();
    return 0;
}
```

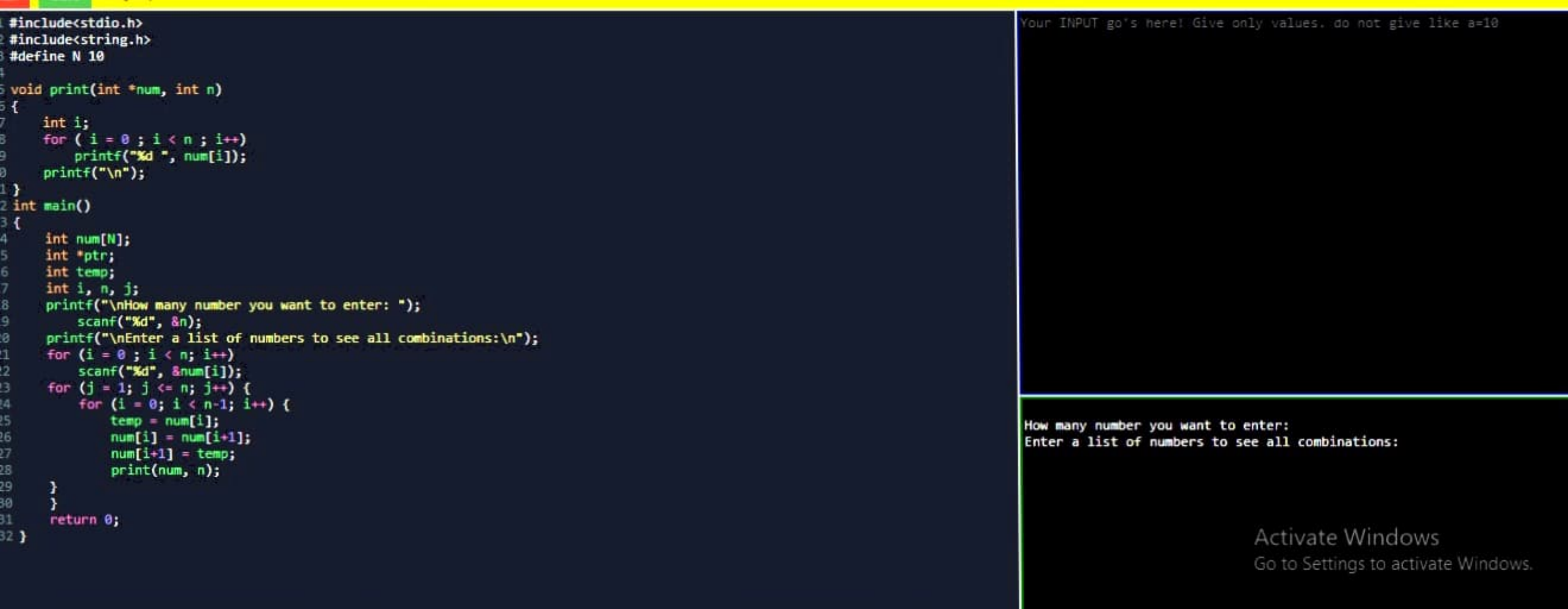
```
8
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char* copyString(char s[])
{
    int i;
    char* s2;
    s2 = (char*)malloc(20);

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

        s2[i] = s[i];
    }
    return (char*)s2;
}
int main()
{
    char s1[20] = "c programming";
    char* s2;
    s2 = copyString(s1);
    printf("%s", s2);
    return 0;
}
```

```
5
8
```

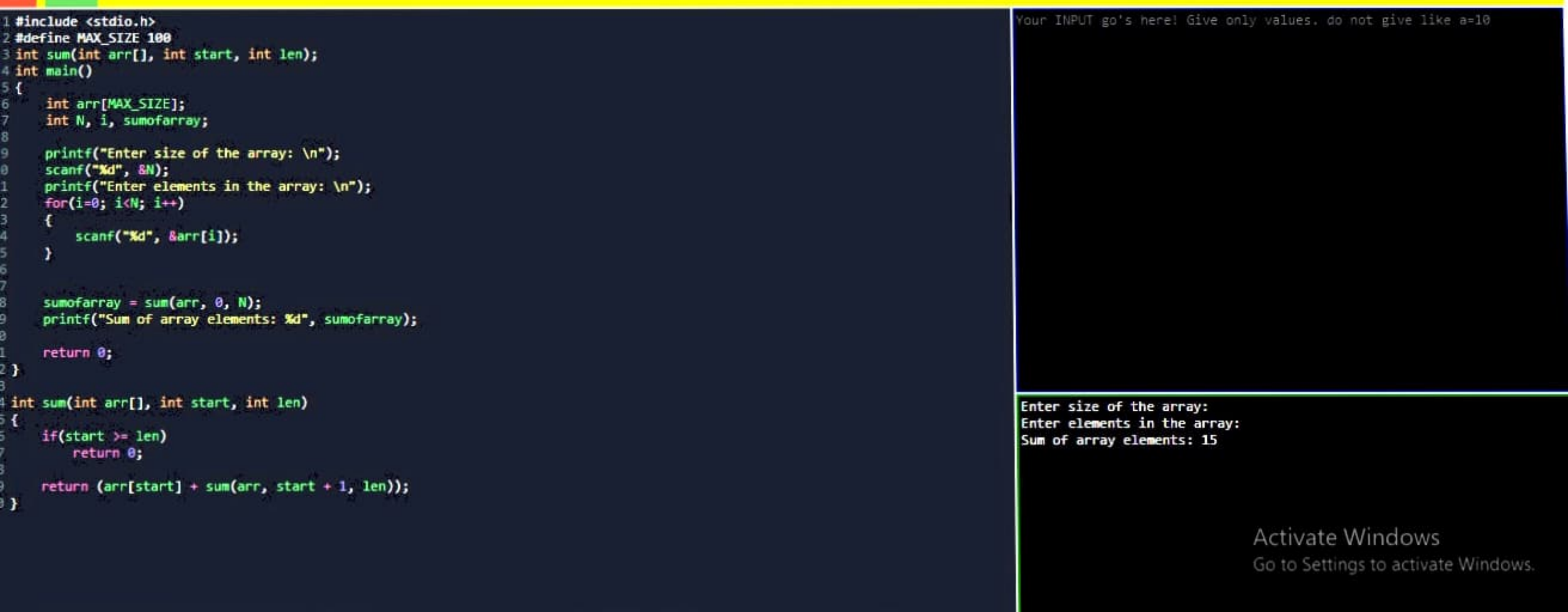
```
c programming⊠
```



```c
#include<stdio.h>
#include<string.h>
#define N 10

void print(int *num, int n)
{
    int i;
    for ( i = 0 ; i < n ; i++)
        printf("%d ", num[i]);
    printf("\n");
}
int main()
{
    int num[N];
    int *ptr;
    int temp;
    int i, n, j;
    printf("\nHow many number you want to enter: ");
        scanf("%d", &n);
    printf("\nEnter a list of numbers to see all combinations:\n");
    for (i = 0 ; i < n; i++)
        scanf("%d", &num[i]);
    for (j = 1; j <= n; j++) {
        for (i = 0; i < n-1; i++) {
            temp = num[i];
            num[i] = num[i+1];
            num[i+1] = temp;
            print(num, n);
    }
    }
    return 0;
}
```

Your INPUT go's here! Give only values. do not give like a=10

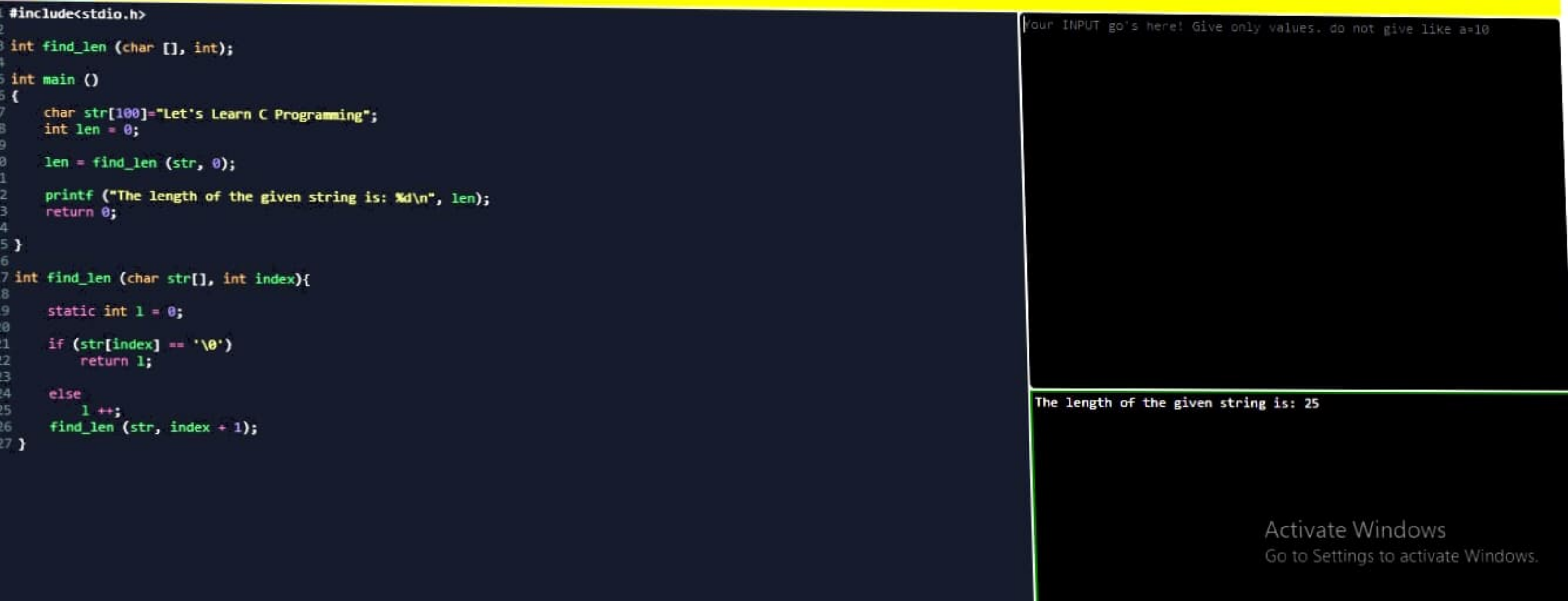
```
How many number you want to enter:
Enter a list of numbers to see all combinations:
```



```c
#include <stdio.h>
#define MAX_SIZE 100
int sum(int arr[], int start, int len);
int main()
{
    int arr[MAX_SIZE];
    int N, i, sumofarray;

    printf("Enter size of the array: \n");
    scanf("%d", &N);
    printf("Enter elements in the array: \n");
    for(i=0; i<N; i++)
    {
        scanf("%d", &arr[i]);
    }


    sumofarray = sum(arr, 0, N);
    printf("Sum of array elements: %d", sumofarray);

    return 0;
}

int sum(int arr[], int start, int len)
{
    if(start >= len)
        return 0;

    return (arr[start] + sum(arr, start + 1, len));
}
```

Your INPUT go's here! Give only values. do not give like a=10

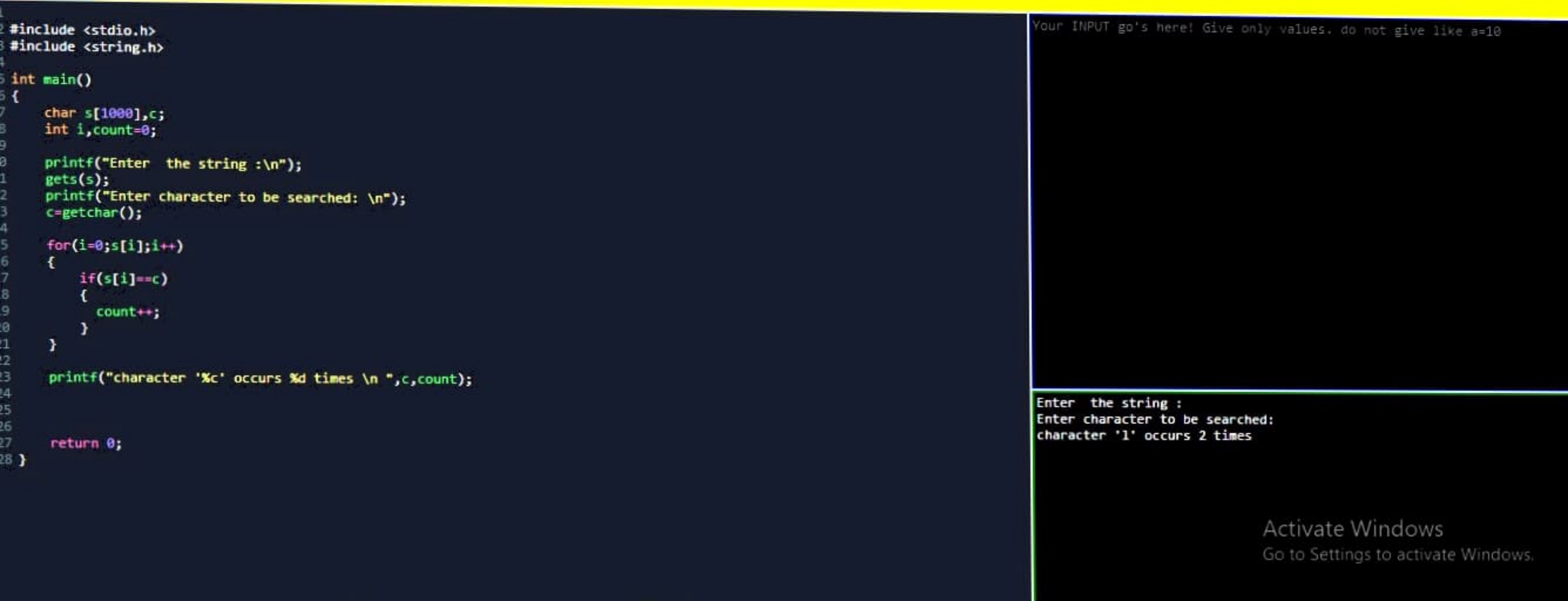
```
Enter size of the array:
Enter elements in the array:
Sum of array elements: 15
```



```c
#include<stdio.h>

int find_len (char [], int);

int main ()
{
    char str[100]="Let's Learn C Programming";
    int len = 0;

    len = find_len (str, 0);

    printf ("The length of the given string is: %d\n", len);
    return 0;

}

int find_len (char str[], int index){

    static int l = 0;

    if (str[index] == '\0')
        return l;

    else
        l ++;
    find_len (str, index + 1);
}
```

Your INPUT go's here! Give only values. do not give like a=10

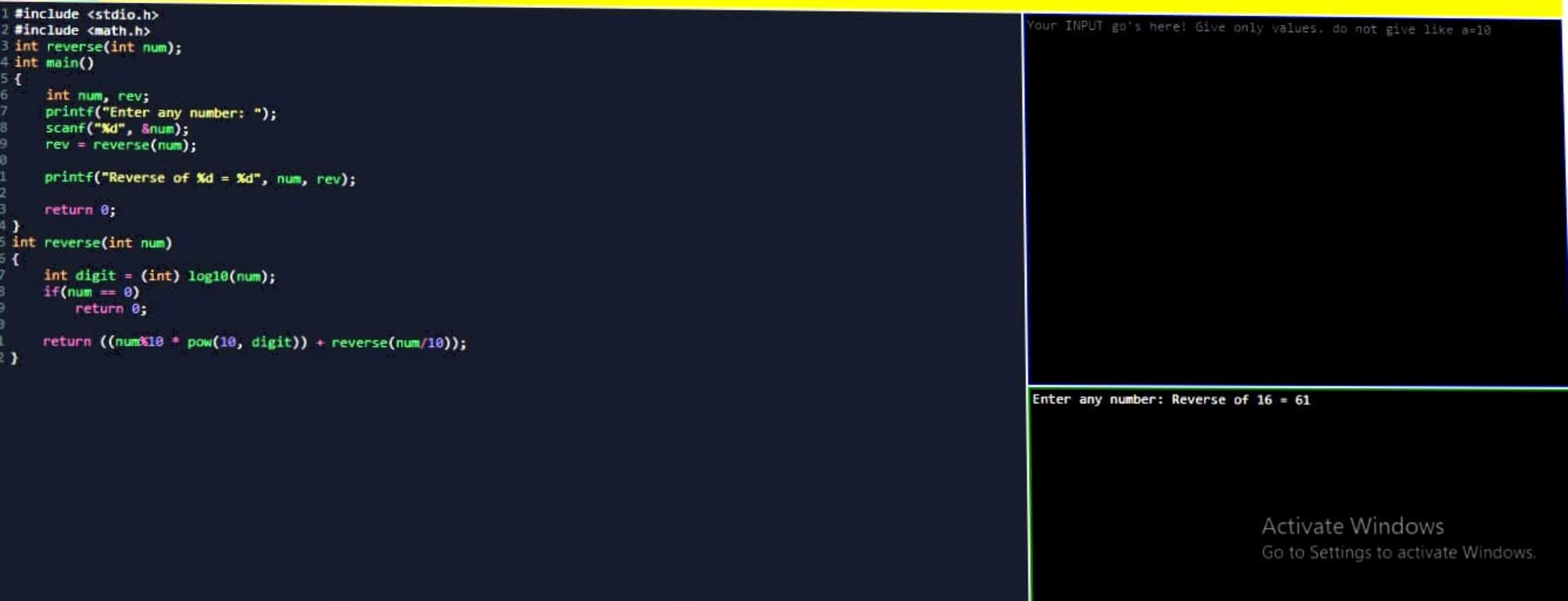
```
The length of the given string is: 25
```



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

int main()
{
    char s[1000],c;
    int i,count=0;

    printf("Enter  the string :\n");
    gets(s);
    printf("Enter character to be searched: \n");
    c=getchar();

    for(i=0;s[i];i++)
    {
        if(s[i]==c)
        {
          count++;
        }
    }

     printf("character '%c' occurs %d times \n ",c,count);



     return 0;
 }
```

Your INPUT go's here! Give only values. do not give like a=10

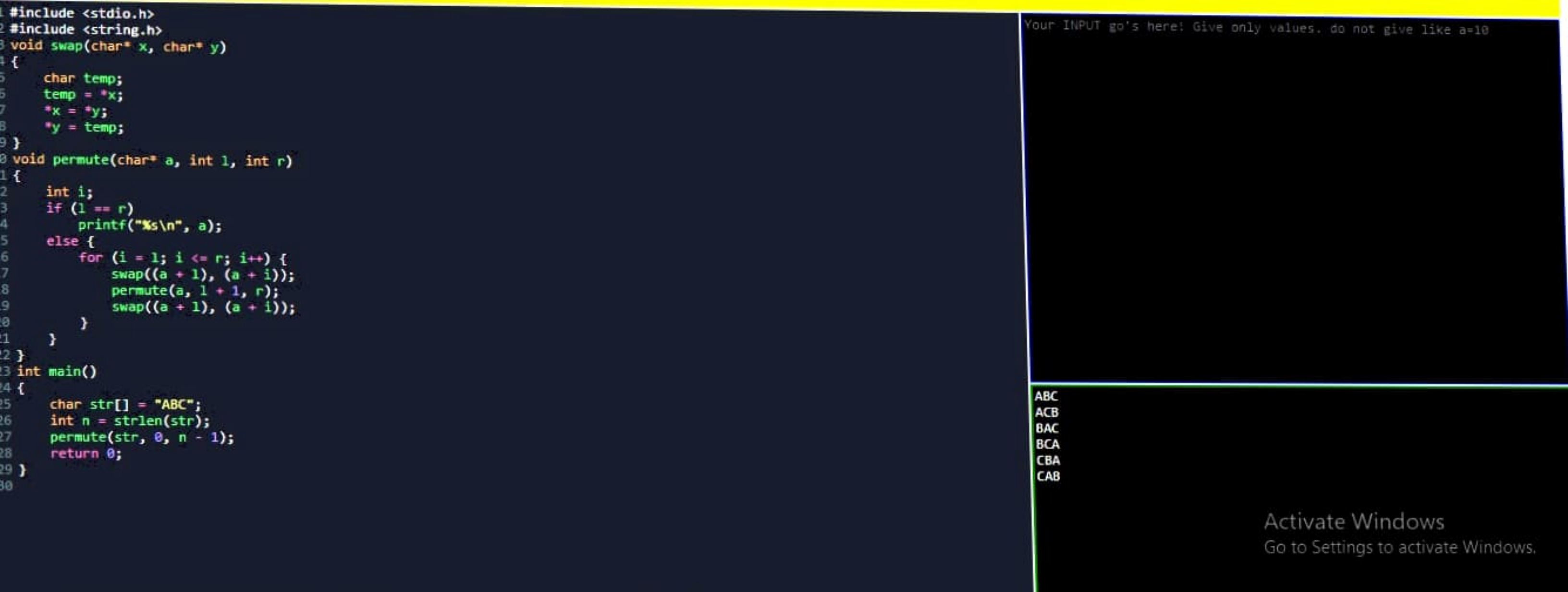
```
Enter  the string :
Enter character to be searched:
character 'l' occurs 2 times
```



```c
#include <stdio.h>
#include <math.h>
int reverse(int num);
int main()
{
    int num, rev;
    printf("Enter any number: ");
    scanf("%d", &num);
    rev = reverse(num);

    printf("Reverse of %d = %d", num, rev);

    return 0;
}
int reverse(int num)
{
    int digit = (int) log10(num);
    if(num == 0)
        return 0;

    return ((num%10 * pow(10, digit)) + reverse(num/10));
}
```

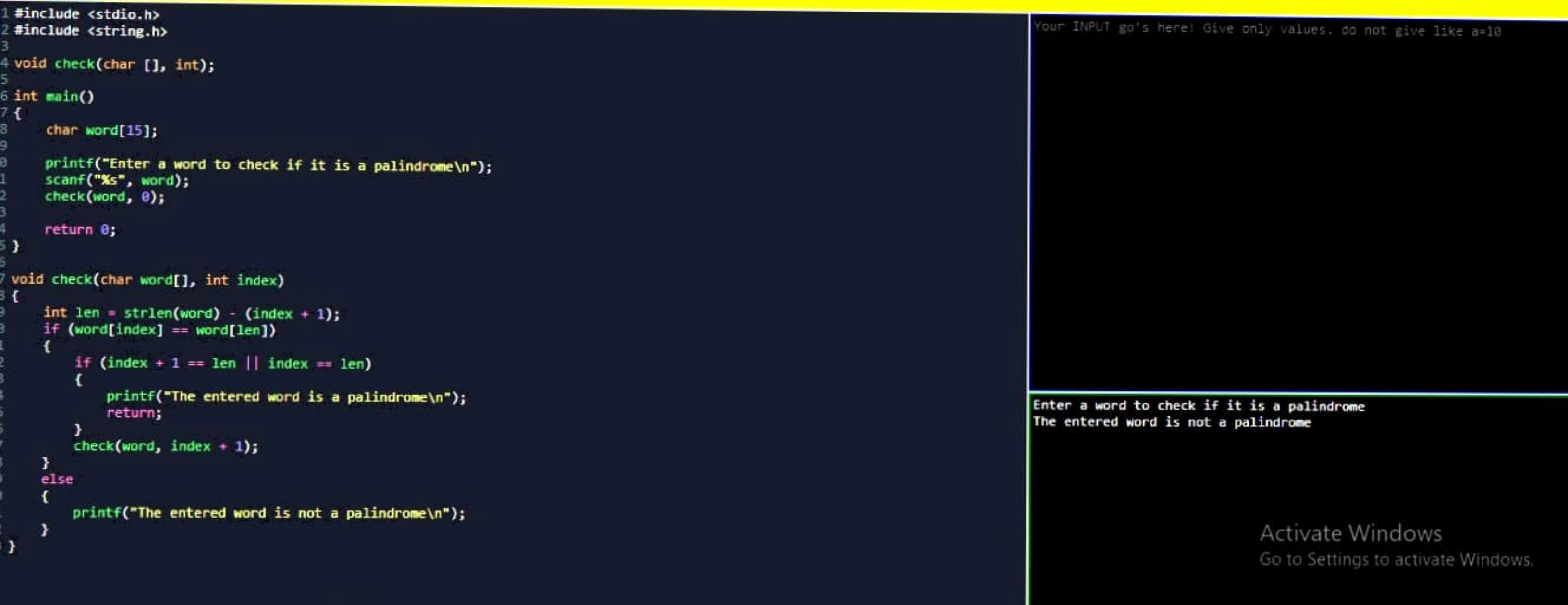
Your INPUT go's here! Give only values. do not give like a=10

```
Enter any number: Reverse of 16 = 61
```


Your INPUT go's here! Give only values. do not give like a=10
Activate Windows
Go to Settings to activate Windows.

```c
#include <stdio.h>
#include <string.h>
void swap(char* x, char* y)
{
    char temp;
    temp = *x;
    *x = *y;
    *y = temp;
}
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));
        }
    }
}
int main()
{
    char str[] = "ABC";
    int n = strlen(str);
    permute(str, 0, n - 1);
    return 0;
}
```

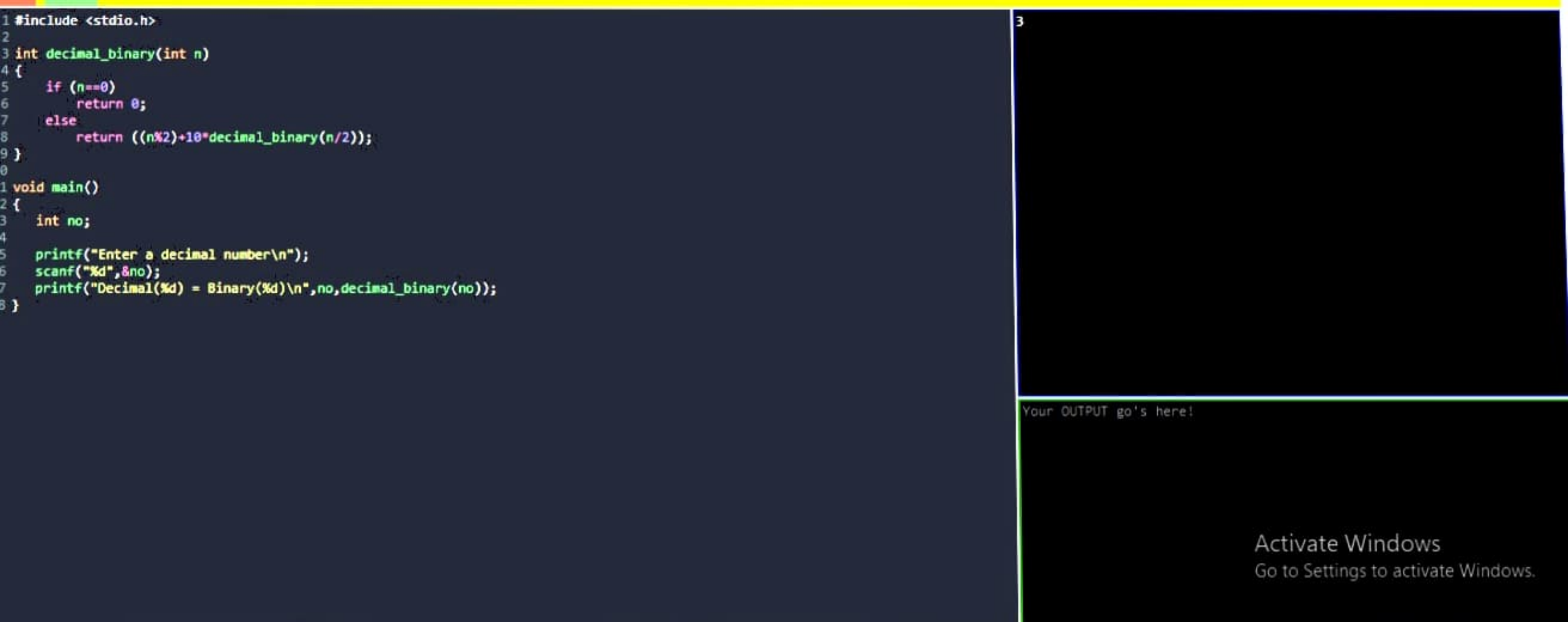
```
ABC
ACB
BAC
BCA
CBA
CAB
```



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

void check(char [], int);

int main()
{
    char word[15];

    printf("Enter a word to check if it is a palindrome\n");
    scanf("%s", word);
    check(word, 0);

    return 0;
}

void check(char word[], int index)
{
    int len = strlen(word) - (index + 1);
    if (word[index] == word[len])
    {
        if (index + 1 == len || index == len)
        {
            printf("The entered word is a palindrome\n");
            return;
        }
        check(word, index + 1);
    }
    else
    {
        printf("The entered word is not a palindrome\n");
    }
}
```

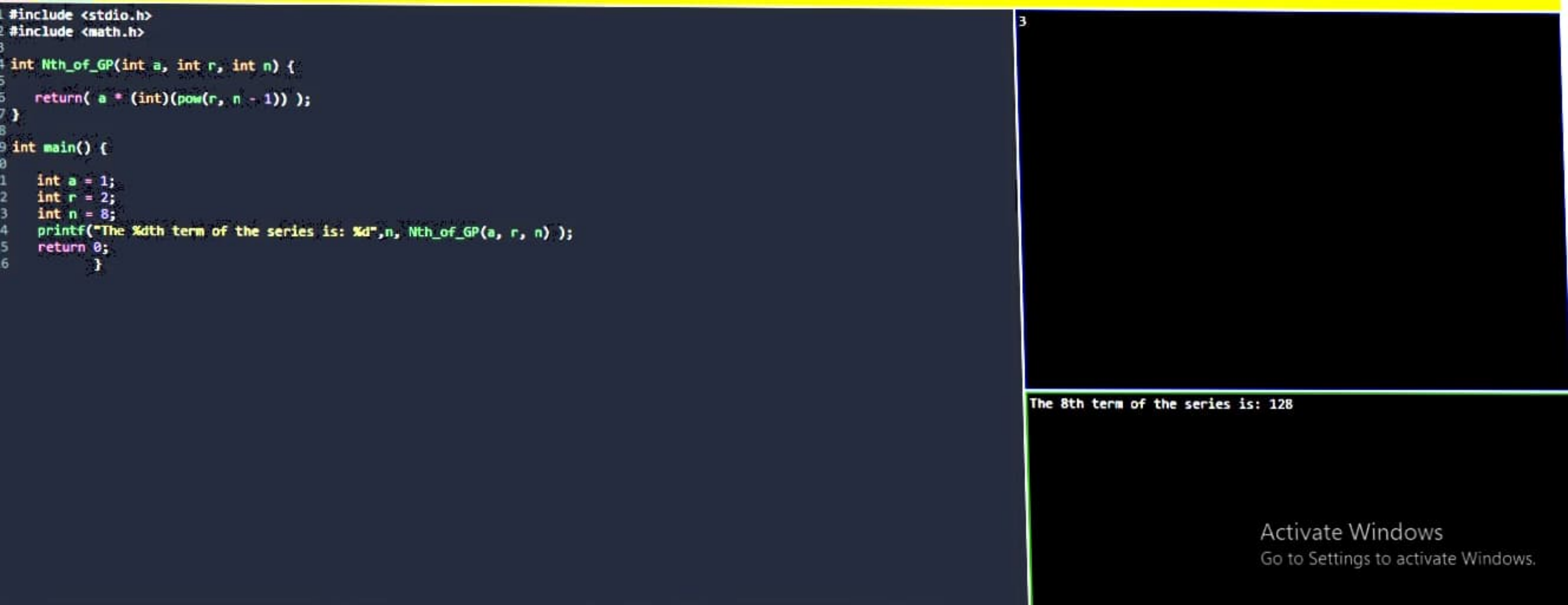
Your INPUT go's here! Give only values. do not give like a=10

```
Enter a word to check if it is a palindrome
The entered word is not a palindrome
```



```c
#include <stdio.h>

int decimal_binary(int n)
{
    if (n==0)
        return 0;
    else
        return ((n%2)+10*decimal_binary(n/2));
}

void main()
{
   int no;

   printf("Enter a decimal number\n");
   scanf("%d",&no);
   printf("Decimal(%d) = Binary(%d)\n",no,decimal_binary(no));
}
```

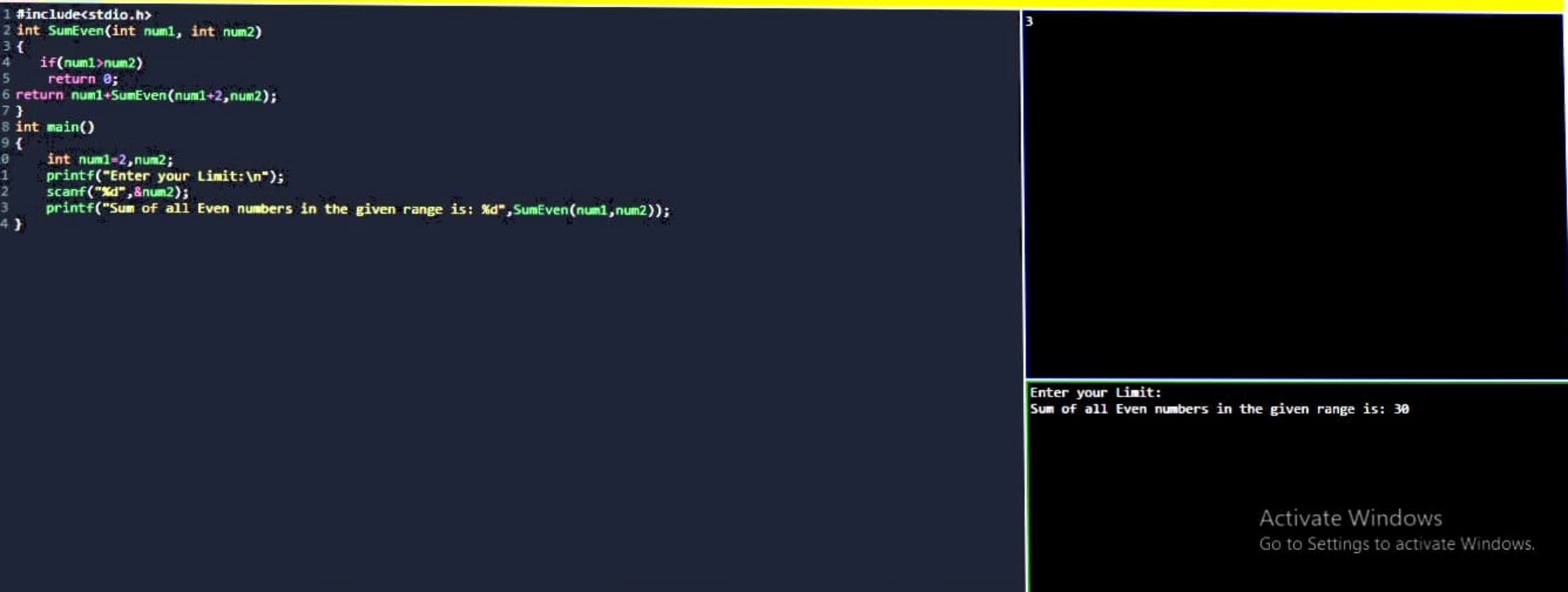
```
3
```



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

int Nth_of_GP(int a, int r, int n) {

   return( a * (int)(pow(r, n - 1)) );
}

int main() {

   int a = 1;
   int r = 2;
   int n = 8;
   printf("The %dth term of the series is: %d",n, Nth_of_GP(a, r, n) );
   return 0;
          }
```

```
3
```

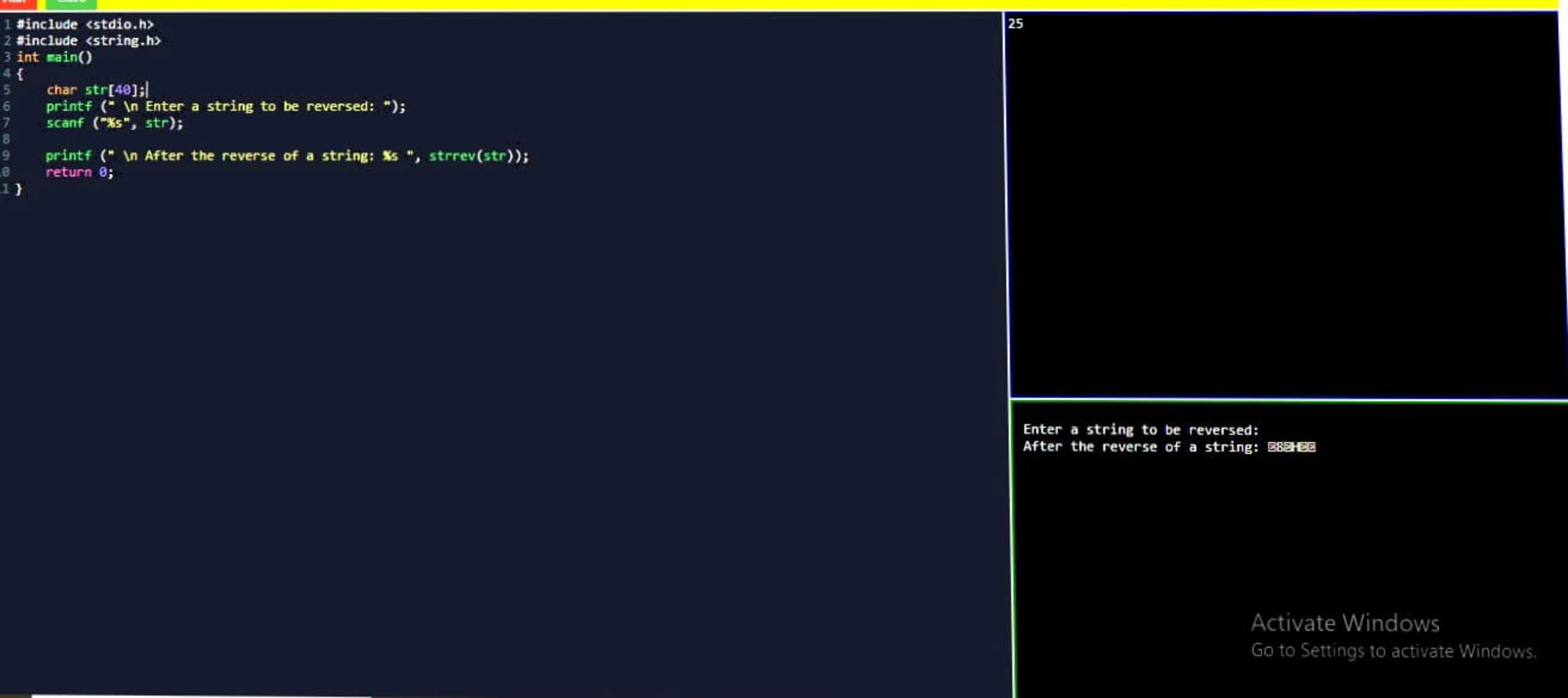
```
The 8th term of the series is: 128
```



```c
#include<stdio.h>
int SumEven(int num1, int num2)
{
   if(num1>num2)
    return 0;
return num1+SumEven(num1+2,num2);
}
int main()
{
    int num1=2,num2;
    printf("Enter your Limit:\n");
    scanf("%d",&num2);
    printf("Sum of all Even numbers in the given range is: %d",SumEven(num1,num2));
}
```

```
3
```

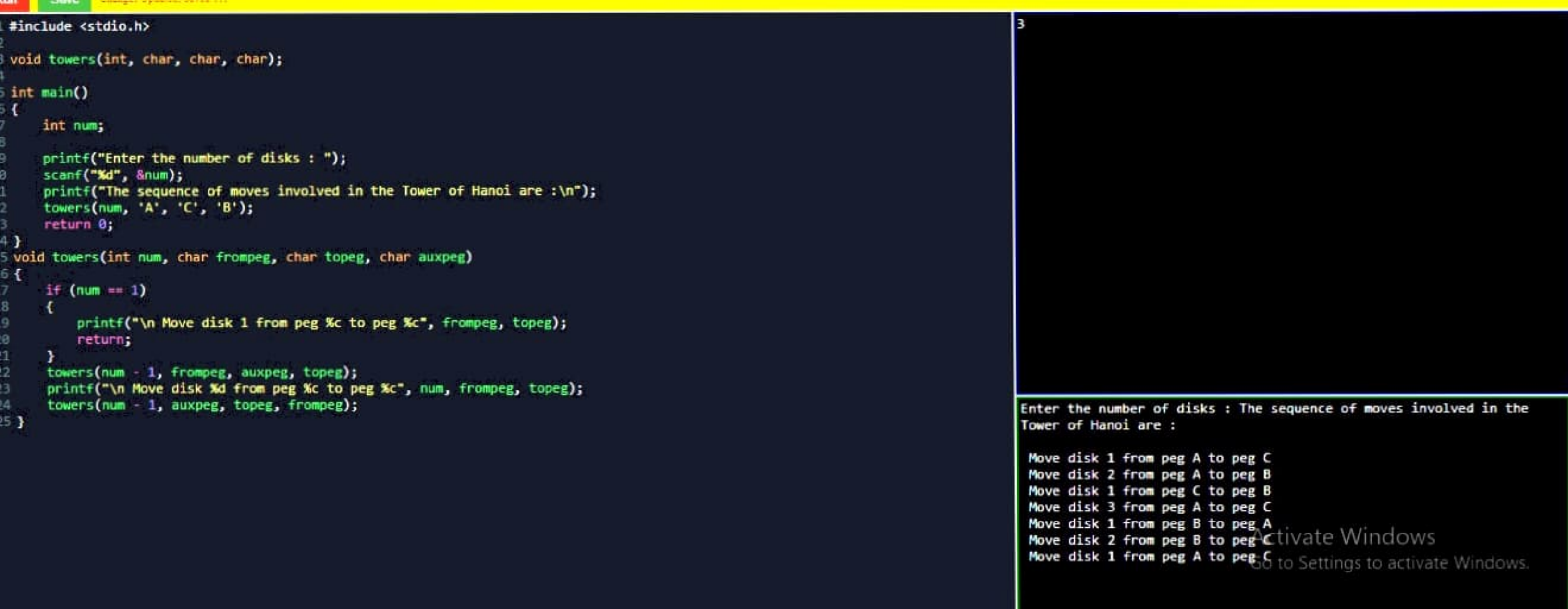
```
Enter your Limit:
Sum of all Even numbers in the given range is: 30
```



```c
#include <stdio.h>
#include <string.h>
int main()
{
    char str[40];
    printf (" \n Enter a string to be reversed: ");
    scanf ("%s", str);

    printf (" \n After the reverse of a string: %s ", strrev(str));
    return 0;
}
```

```
25
```

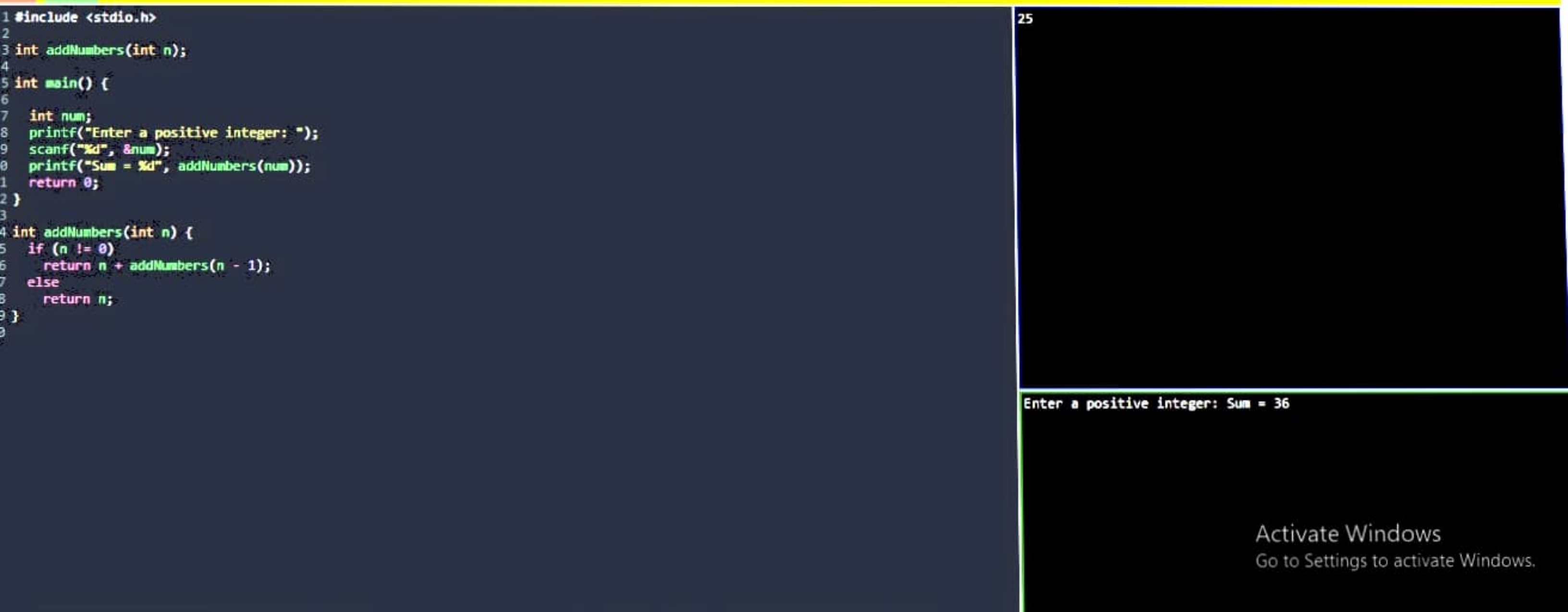
```
Enter a string to be reversed:
After the reverse of a string: □8□H□□
```



```c
#include <stdio.h>

void towers(int, char, char, char);

int main()
{
    int num;

    printf("Enter the number of disks : ");
    scanf("%d", &num);
    printf("The sequence of moves involved in the Tower of Hanoi are :\n");
    towers(num, 'A', 'C', 'B');
    return 0;
}
void towers(int num, char frompeg, char topeg, char auxpeg)
{
    if (num == 1)
    {
        printf("\n Move disk 1 from peg %c to peg %c", frompeg, topeg);
        return;
    }
    towers(num - 1, frompeg, auxpeg, topeg);
    printf("\n Move disk %d from peg %c to peg %c", num, frompeg, topeg);
    towers(num - 1, auxpeg, topeg, frompeg);
}
```

```
3
```

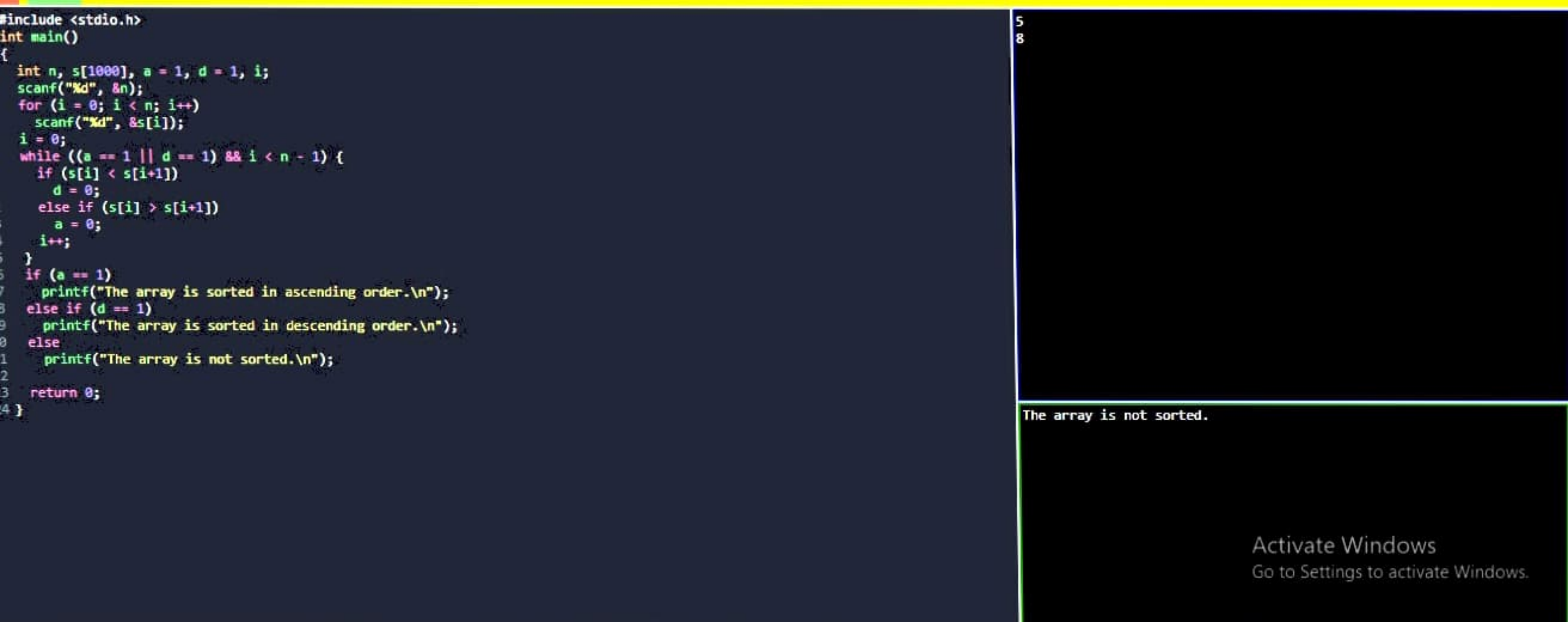
```
Enter the number of disks : The sequence of moves involved in the
Tower of Hanoi are :

 Move disk 1 from peg A to peg C
 Move disk 2 from peg A to peg B
 Move disk 1 from peg C to peg B
 Move disk 3 from peg A to peg C
 Move disk 1 from peg B to peg A
 Move disk 2 from peg B to peg C
 Move disk 1 from peg A to peg C
```



```c
#include <stdio.h>

int addNumbers(int n);

int main() {

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum = %d", addNumbers(num));
  return 0;
}

int addNumbers(int n) {
  if (n != 0)
    return n + addNumbers(n - 1);
  else
    return n;
}
```

```
25
```

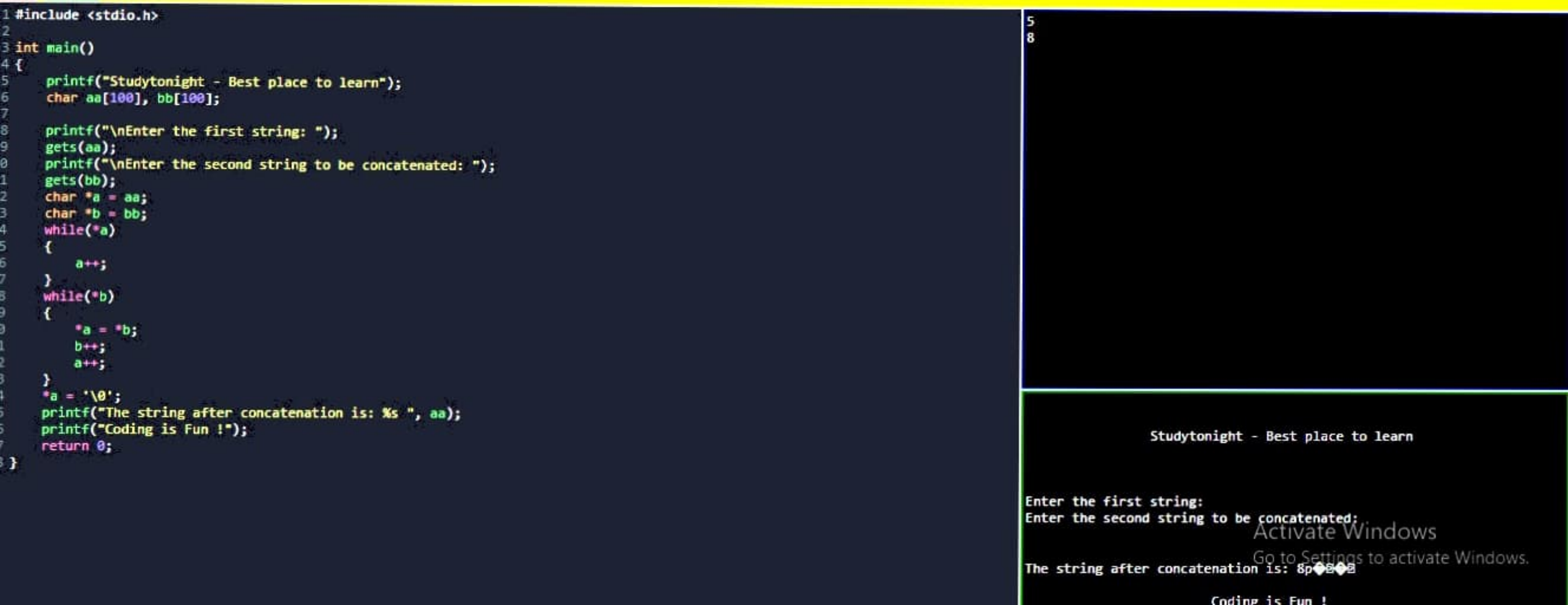
```
Enter a positive integer: Sum = 36
```



```c
#include <stdio.h>
int main()
{
  int n, s[1000], a = 1, d = 1, i;
  scanf("%d", &n);
  for (i = 0; i < n; i++)
    scanf("%d", &s[i]);
  i = 0;
  while ((a == 1 || d == 1) && i < n - 1) {
    if (s[i] < s[i+1])
      d = 0;
    else if (s[i] > s[i+1])
      a = 0;
    i++;
  }
  if (a == 1)
    printf("The array is sorted in ascending order.\n");
  else if (d == 1)
    printf("The array is sorted in descending order.\n");
  else
    printf("The array is not sorted.\n");

  return 0;
}
```

```
5
8
```

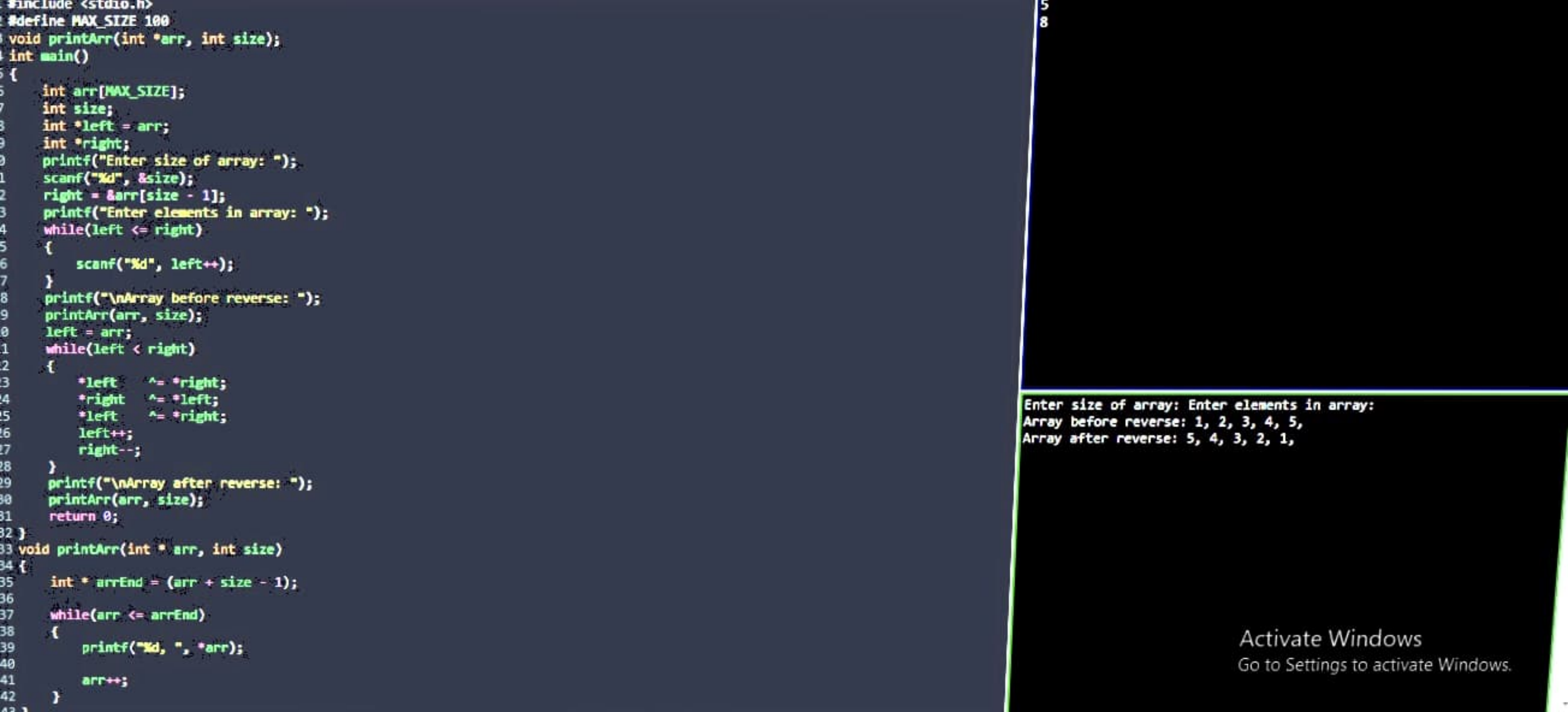
```
The array is not sorted.
```



```c
#include <stdio.h>

int main()
{
    printf("Studytonight - Best place to learn");
    char aa[100], bb[100];

    printf("\nEnter the first string: ");
    gets(aa);
    printf("\nEnter the second string to be concatenated: ");
    gets(bb);
    char *a = aa;
    char *b = bb;
    while(*a)
    {
        a++;
    }
    while(*b)
    {
        *a = *b;
        b++;
        a++;
    }
    *a = '\0';
    printf("The string after concatenation is: %s ", aa);
    printf("Coding is Fun !");
    return 0;
}
```

```
5
8
```

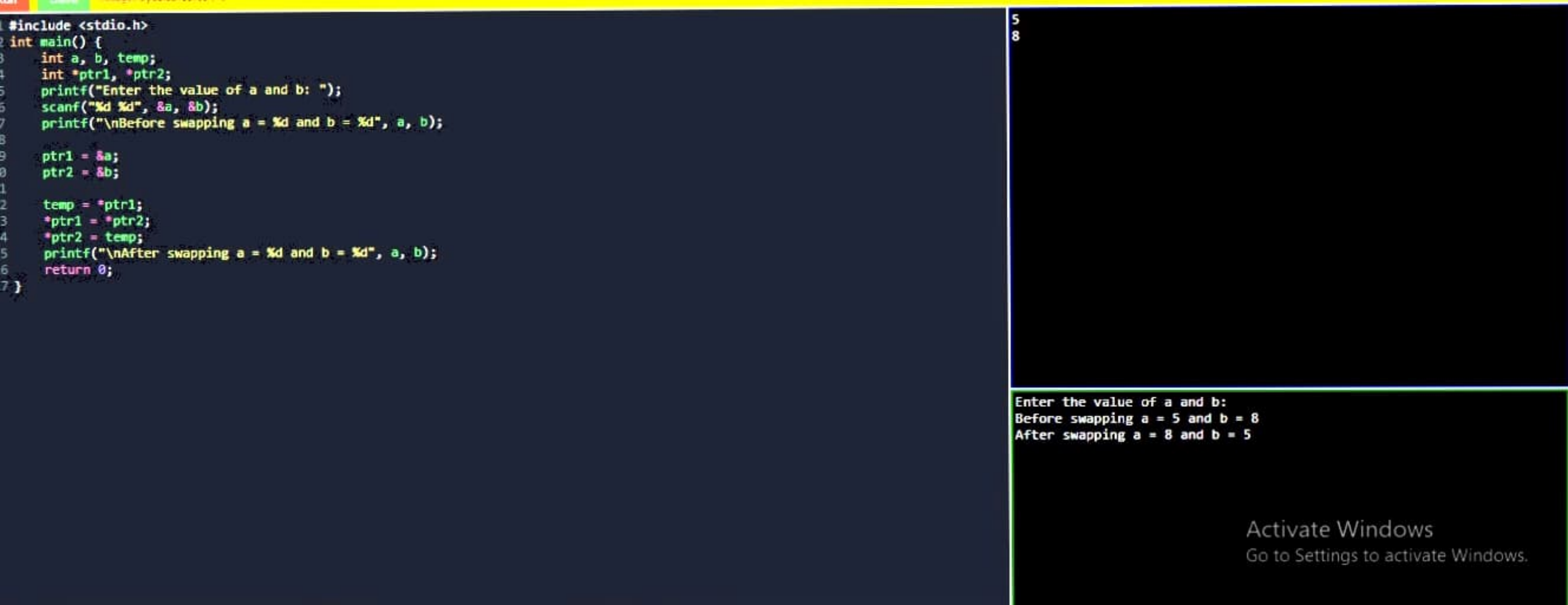
```
                    Studytonight - Best place to learn


Enter the first string:
Enter the second string to be concatenated:


The string after concatenation is: 8p�▯�▯

                    Coding is Fun !
```



```c
#include <stdio.h>
#define MAX_SIZE 100
void printArr(int *arr, int size);
int main()
{
    int arr[MAX_SIZE];
    int size;
    int *left = arr;
    int *right;
    printf("Enter size of array: ");
    scanf("%d", &size);
    right = &arr[size - 1];
    printf("Enter elements in array: ");
    while(left <= right)
    {
        scanf("%d", left++);
    }
    printf("\nArray before reverse: ");
    printArr(arr, size);
    left = arr;
    while(left < right)
    {
        *left   ^= *right;
        *right  ^= *left;
        *left   ^= *right;
        left++;
        right--;
    }
    printf("\nArray after reverse: ");
    printArr(arr, size);
    return 0;
}
void printArr(int * arr, int size)
{
    int * arrEnd = (arr + size - 1);

    while(arr <= arrEnd)
    {
        printf("%d, ", *arr);

        arr++;
    }
}
```

```
5
8
```

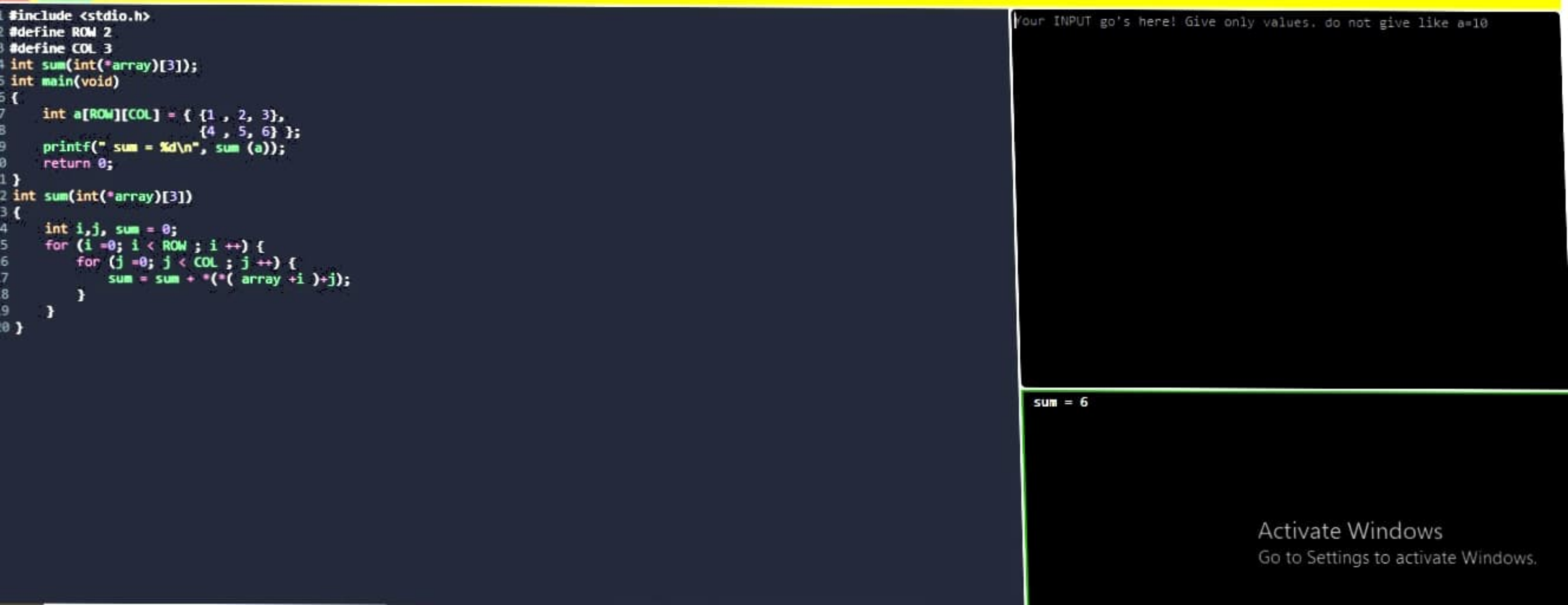
```
Enter size of array: Enter elements in array:
Array before reverse: 1, 2, 3, 4, 5,
Array after reverse: 5, 4, 3, 2, 1,
```



```c
#include <stdio.h>
int main() {
    int a, b, temp;
    int *ptr1, *ptr2;
    printf("Enter the value of a and b: ");
    scanf("%d %d", &a, &b);
    printf("\nBefore swapping a = %d and b = %d", a, b);

    ptr1 = &a;
    ptr2 = &b;

    temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
    printf("\nAfter swapping a = %d and b = %d", a, b);
    return 0;
}
```

```
5
8
```

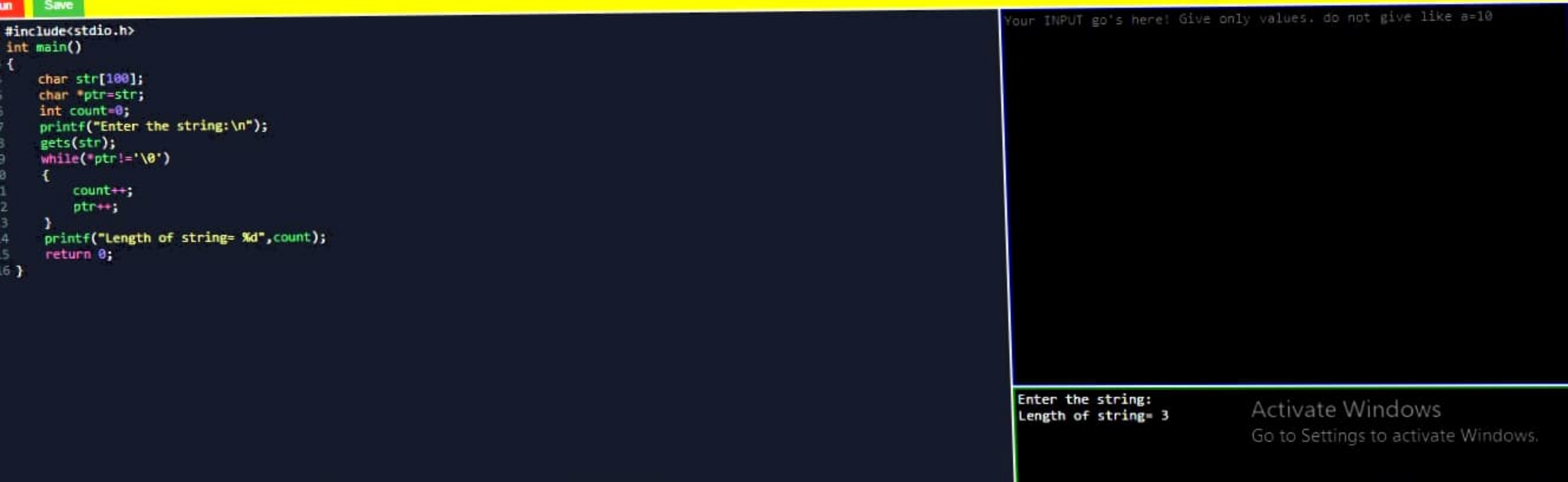
```
Enter the value of a and b:
Before swapping a = 5 and b = 8
After swapping a = 8 and b = 5
```



```c
#include <stdio.h>
#define ROW 2
#define COL 3
int sum(int(*array)[3]);
int main(void)
{
    int a[ROW][COL] = { {1 , 2, 3},
                        {4 , 5, 6} };
    printf(" sum = %d\n", sum (a));
    return 0;
}
int sum(int(*array)[3])
{
    int i,j, sum = 0;
    for (i =0; i < ROW ; i ++) {
        for (j =0; j < COL ; j ++) {
            sum = sum + *(*( array +i )+j);
        }
    }
}
```

Your INPUT go's here! Give only values. do not give like a=10

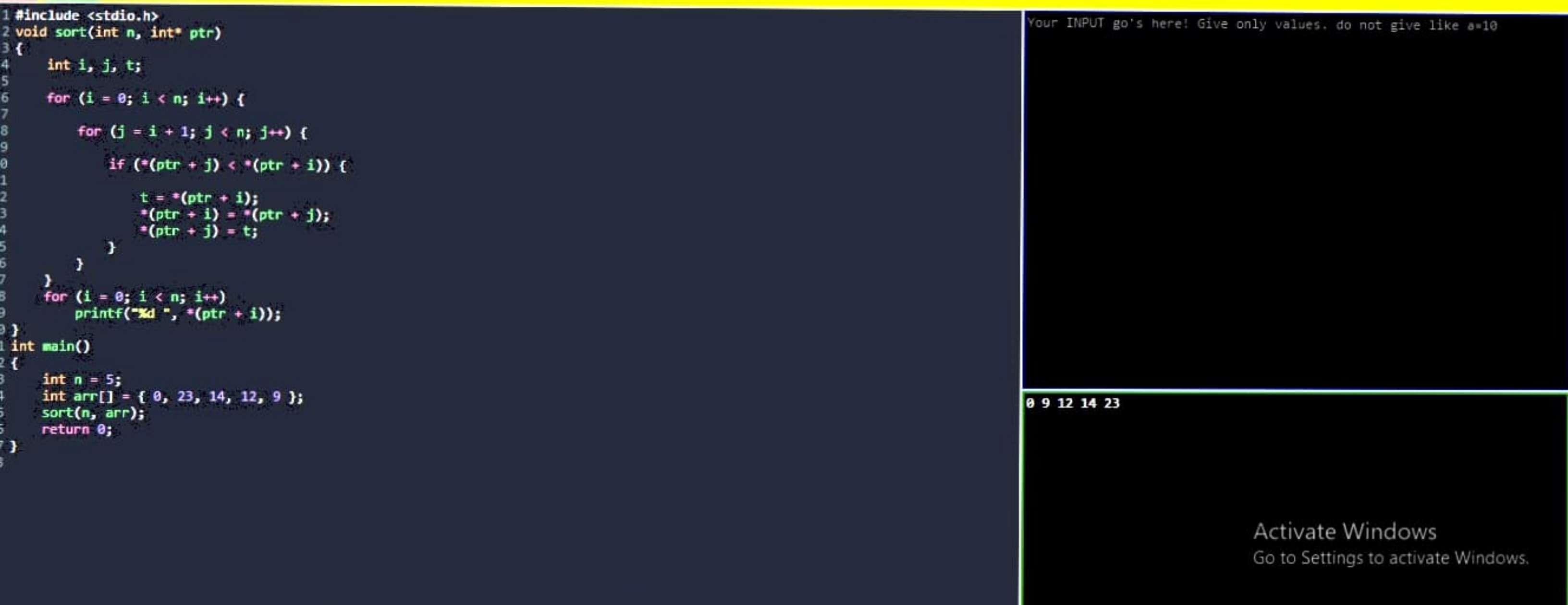
```
sum = 6
```



```c
#include<stdio.h>
int main()
{
    char str[100];
    char *ptr=str;
    int count=0;
    printf("Enter the string:\n");
    gets(str);
    while(*ptr!='\0')
    {
        count++;
        ptr++;
    }
    printf("Length of string= %d",count);
    return 0;
}
```

```
Your INPUT go's here! Give only values. do not give like a=10
```

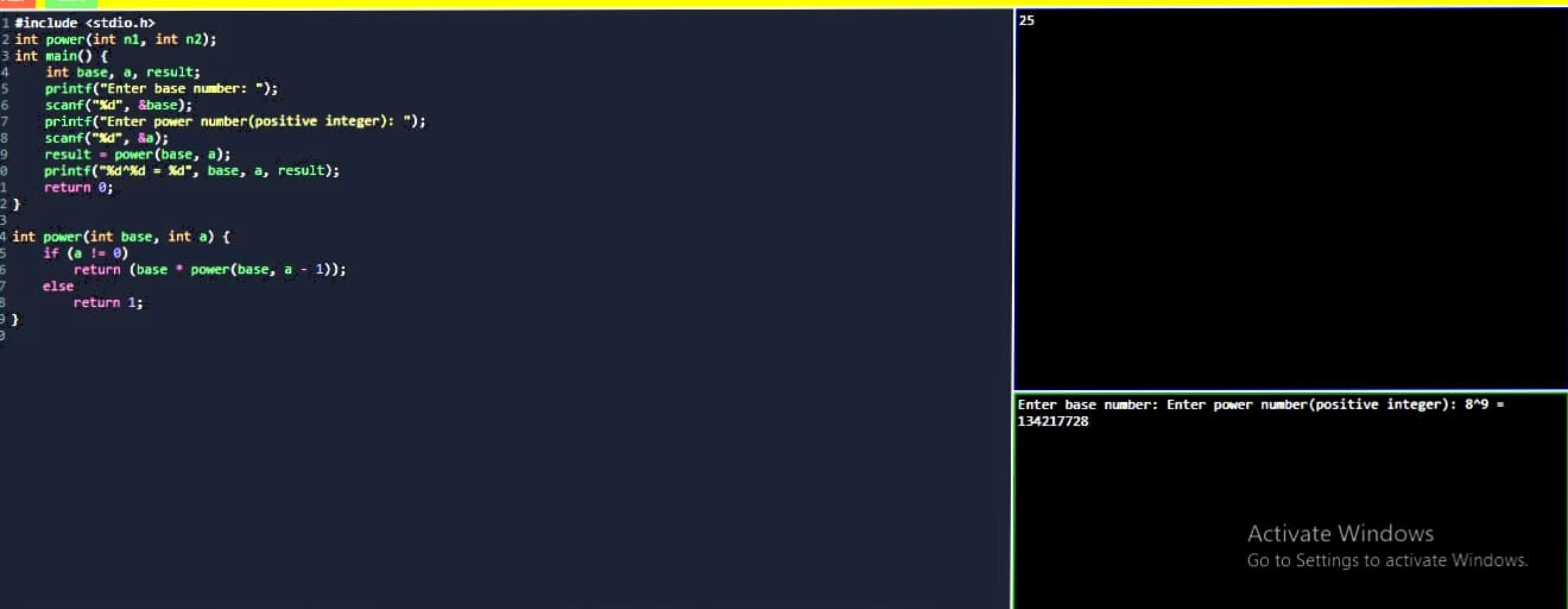
```
Enter the string:
Length of string= 3
```



```c
#include <stdio.h>
void sort(int n, int* ptr)
{
    int i, j, t;

    for (i = 0; i < n; i++) {

        for (j = i + 1; j < n; j++) {

            if (*(ptr + j) < *(ptr + i)) {

                t = *(ptr + i);
                *(ptr + i) = *(ptr + j);
                *(ptr + j) = t;
            }
        }
    }
    for (i = 0; i < n; i++)
        printf("%d ", *(ptr + i));
}
int main()
{
    int n = 5;
    int arr[] = { 0, 23, 14, 12, 9 };
    sort(n, arr);
    return 0;
}
```

Your INPUT go's here! Give only values. do not give like a=10

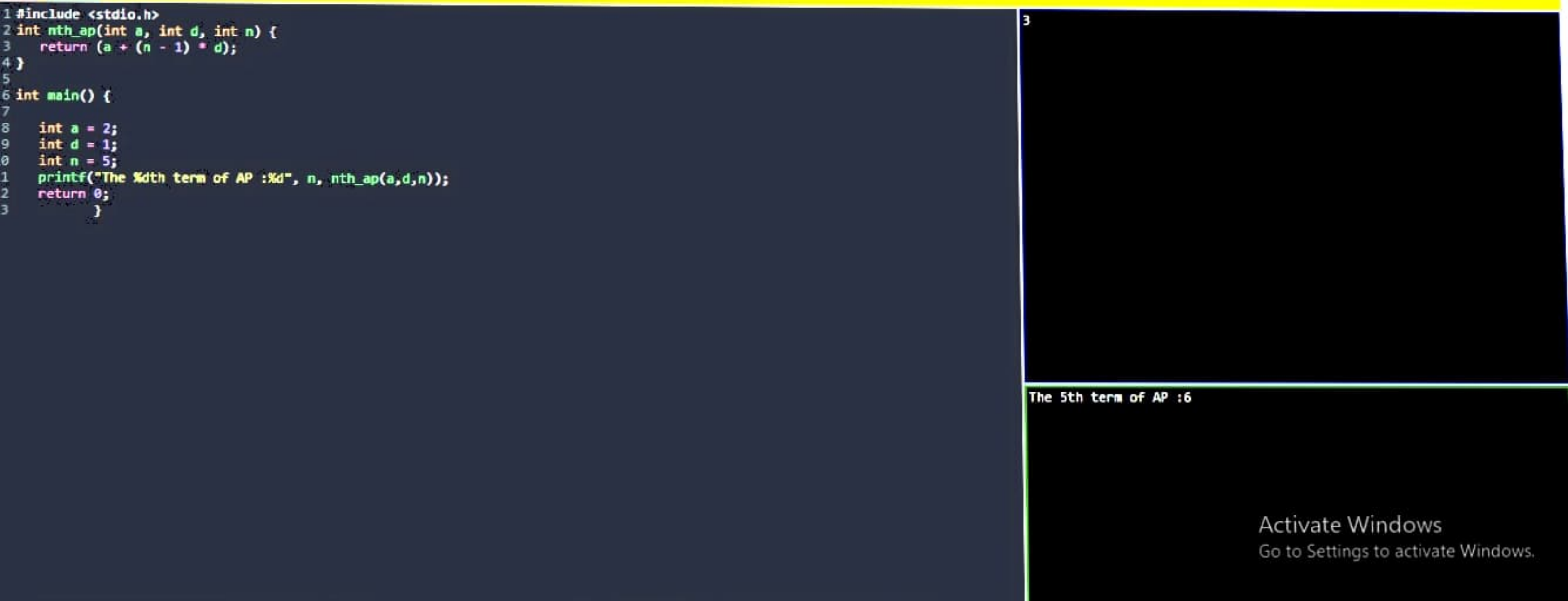
```
0 9 12 14 23
```



```c
#include <stdio.h>
int power(int n1, int n2);
int main() {
    int base, a, result;
    printf("Enter base number: ");
    scanf("%d", &base);
    printf("Enter power number(positive integer): ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d^%d = %d", base, a, result);
    return 0;
}

int power(int base, int a) {
    if (a != 0)
        return (base * power(base, a - 1));
    else
        return 1;
}
```

```
25
```

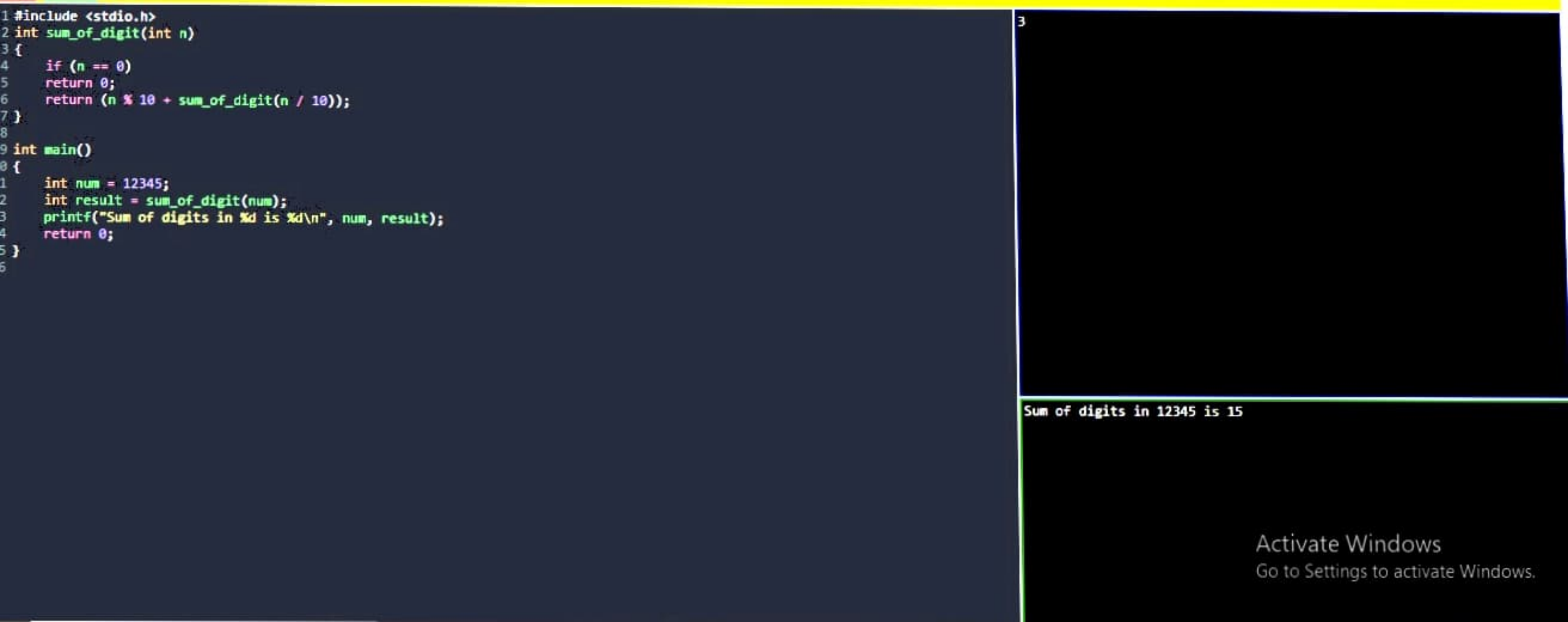
```
Enter base number: Enter power number(positive integer): 8^9 =
134217728
```



```c
#include <stdio.h>
int nth_ap(int a, int d, int n) {
   return (a + (n - 1) * d);
}

int main() {

   int a = 2;
   int d = 1;
   int n = 5;
   printf("The %dth term of AP :%d", n, nth_ap(a,d,n));
   return 0;
           }
```

```
3
```

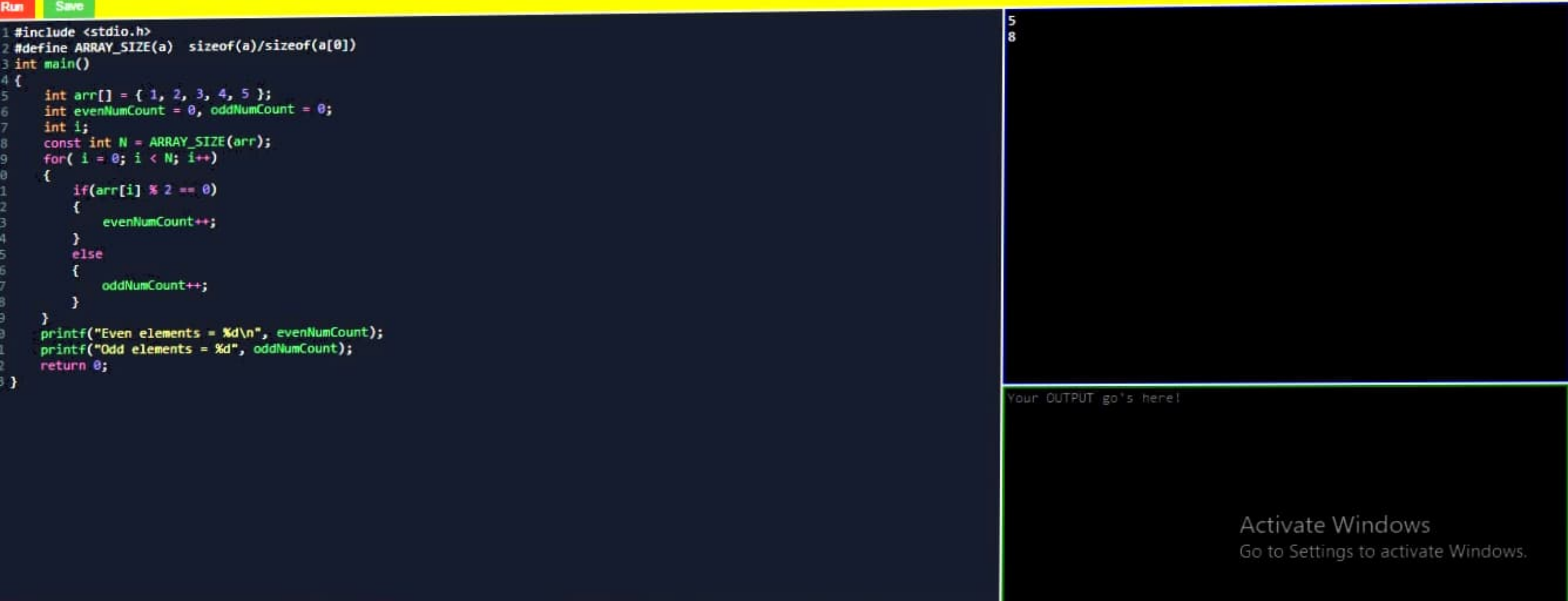
```
The 5th term of AP :6
```



```c
#include <stdio.h>
int sum_of_digit(int n)
{
    if (n == 0)
    return 0;
    return (n % 10 + sum_of_digit(n / 10));
}

int main()
{
    int num = 12345;
    int result = sum_of_digit(num);
    printf("Sum of digits in %d is %d\n", num, result);
    return 0;
}
```

```
3
```

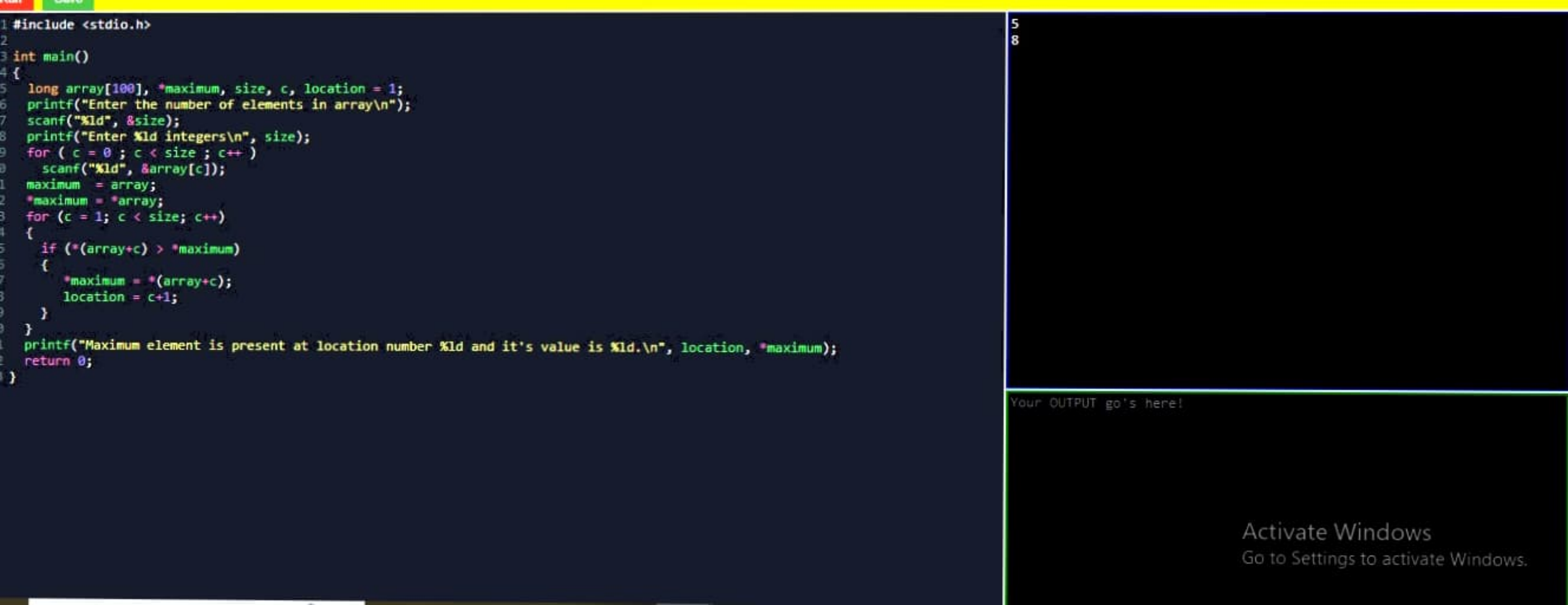
```
Sum of digits in 12345 is 15
```



```
#include <stdio.h>
#define ARRAY_SIZE(a)  sizeof(a)/sizeof(a[0])
int main()
{
    int arr[] = { 1, 2, 3, 4, 5 };
    int evenNumCount = 0, oddNumCount = 0;
    int i;
    const int N = ARRAY_SIZE(arr);
    for( i = 0; i < N; i++)
    {
        if(arr[i] % 2 == 0)
        {
            evenNumCount++;
        }
        else
        {
            oddNumCount++;
        }
    }
    printf("Even elements = %d\n", evenNumCount);
    printf("Odd elements = %d", oddNumCount);
    return 0;
}
```

```
5
8
```

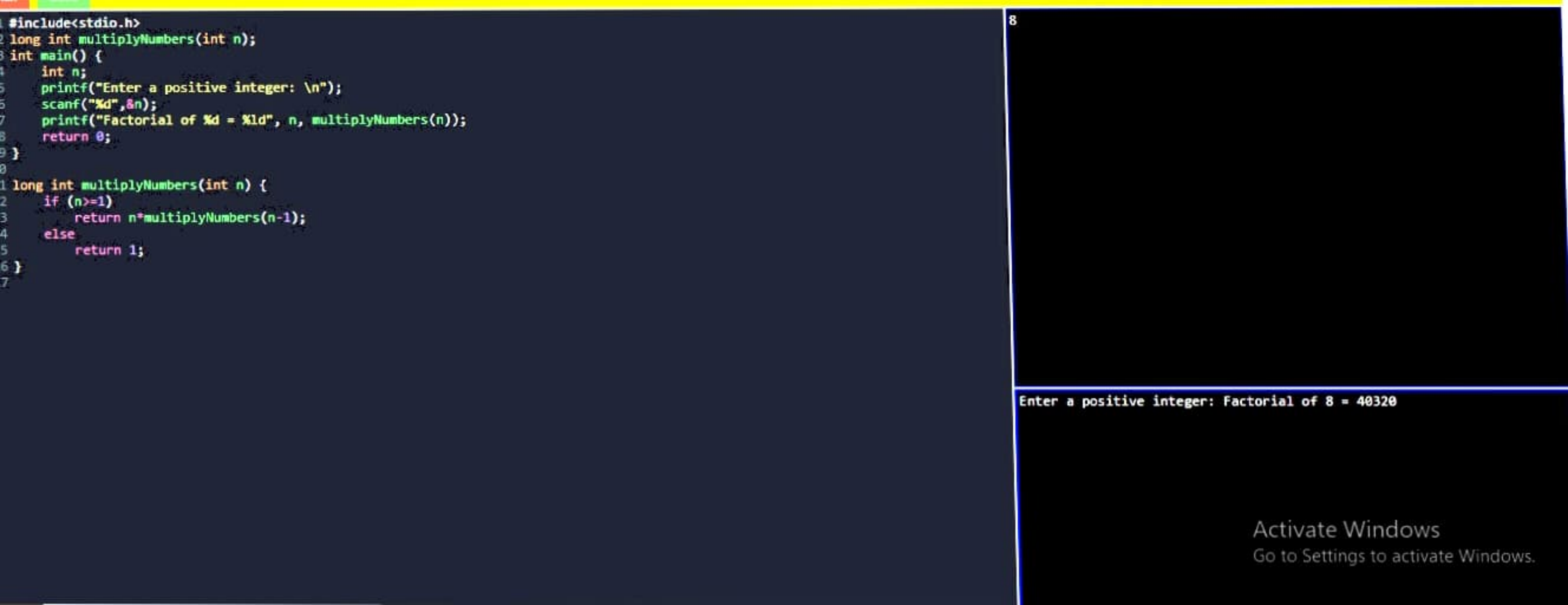
```
Your OUTPUT go's here!
```



```
#include <stdio.h>

int main()
{
  long array[100], *maximum, size, c, location = 1;
  printf("Enter the number of elements in array\n");
  scanf("%ld", &size);
  printf("Enter %ld integers\n", size);
  for ( c = 0 ; c < size ; c++ )
    scanf("%ld", &array[c]);
  maximum  = array;
  *maximum = *array;
  for (c = 1; c < size; c++)
  {
    if (*(array+c) > *maximum)
    {
       *maximum = *(array+c);
       location = c+1;
    }
  }
 printf("Maximum element is present at location number %ld and it's value is %ld.\n", location, *maximum);
 return 0;
}
```

```
5
8
```

```
Your OUTPUT go's here!
```



```
#include<stdio.h>
long int multiplyNumbers(int n);
int main() {
    int n;
    printf("Enter a positive integer: \n");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, multiplyNumbers(n));
    return 0;
}

long int multiplyNumbers(int n) {
    if (n>=1)
        return n*multiplyNumbers(n-1);
    else
        return 1;
}
```

```
8
```

```
Enter a positive integer: Factorial of 8 = 40320
```