IT161: Introduction to Programming and Problem Solving

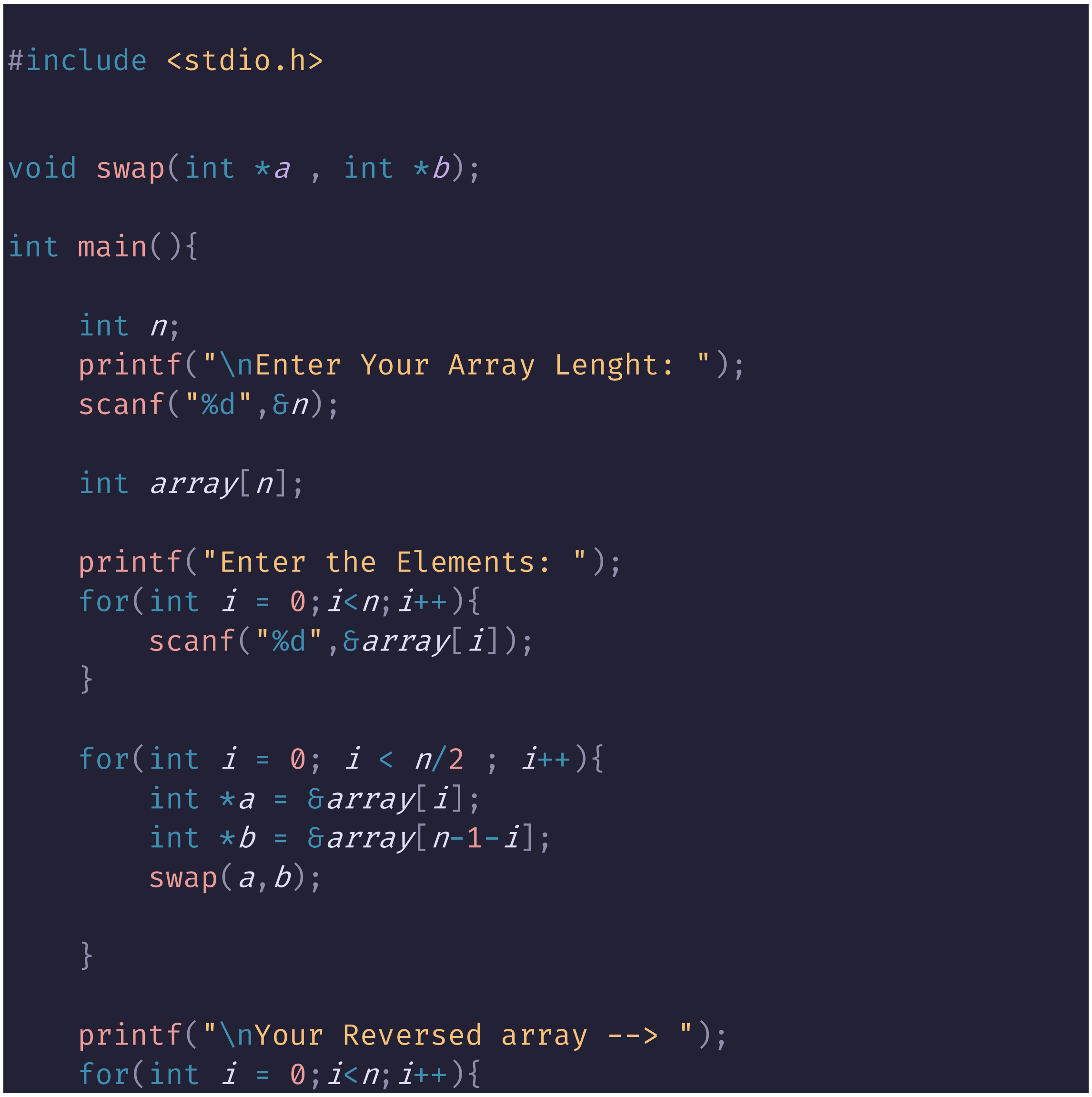
Lab 6/Assignment 6

NAME: SHIDDHARTH BHEEMARAI TASHILDAR
ROLL NUMBER: 202451151

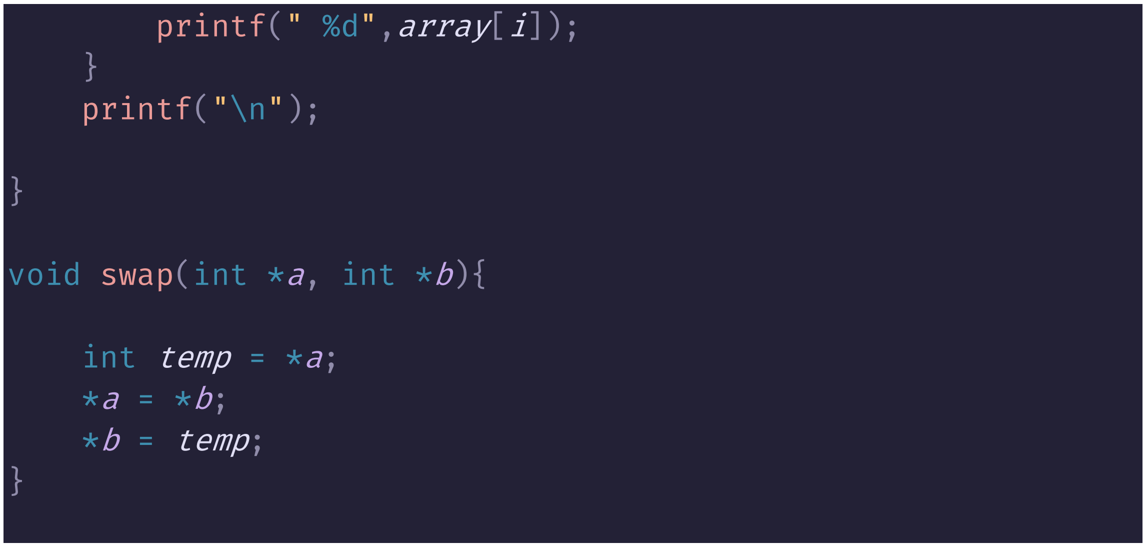
PROGRAMS

1. **C function to reverse an array using pointers**

Code:

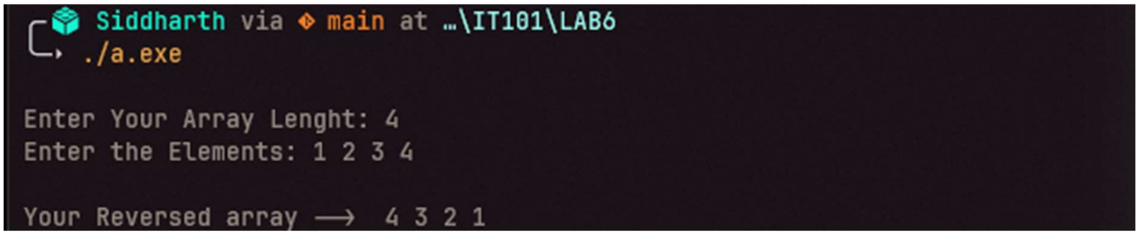


```c
#include <stdio.h>


void swap(int *a , int *b);

int main(){

    int n;
    printf("\nEnter Your Array Lenght: ");
    scanf("%d",&n);

    int array[n];

    printf("Enter the Elements: ");
    for(int i = 0;i<n;i++){
        scanf("%d",&array[i]);
    }

    for(int i = 0; i < n/2 ; i++){
        int *a = &array[i];
        int *b = &array[n-1-i];
        swap(a,b);

    }

    printf("\nYour Reversed array --> ");
    for(int i = 0;i<n;i++){
```



```c
        printf(" %d",array[i]);
    }
    printf("\n");

}

void swap(int *a, int *b){

    int temp = *a;
    *a = *b;
    *b = temp;
}
```

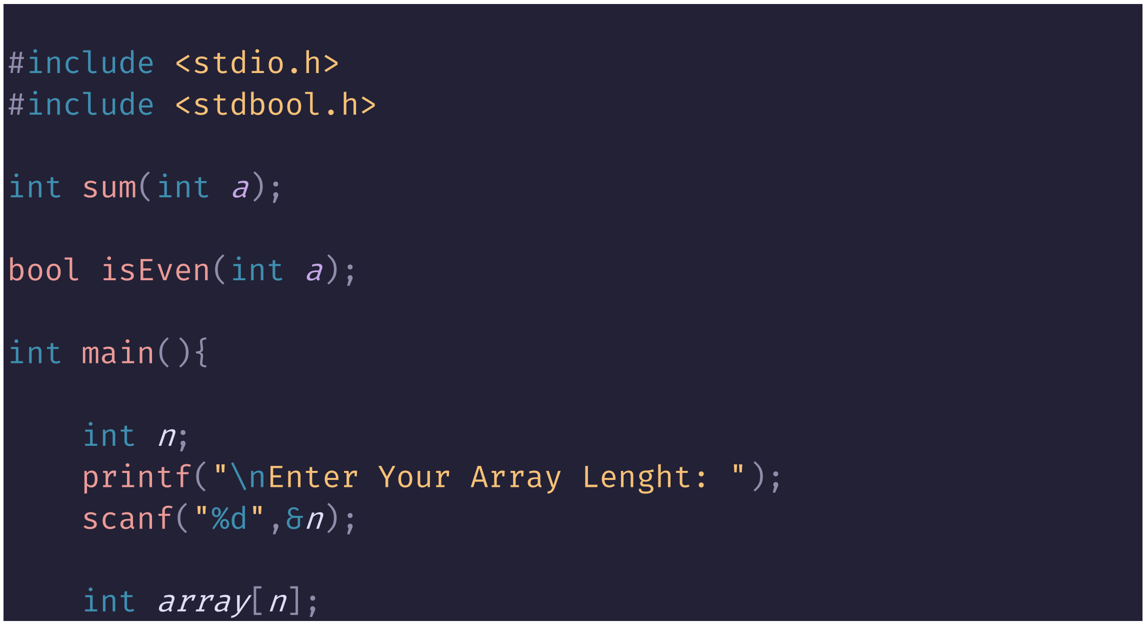
Output:



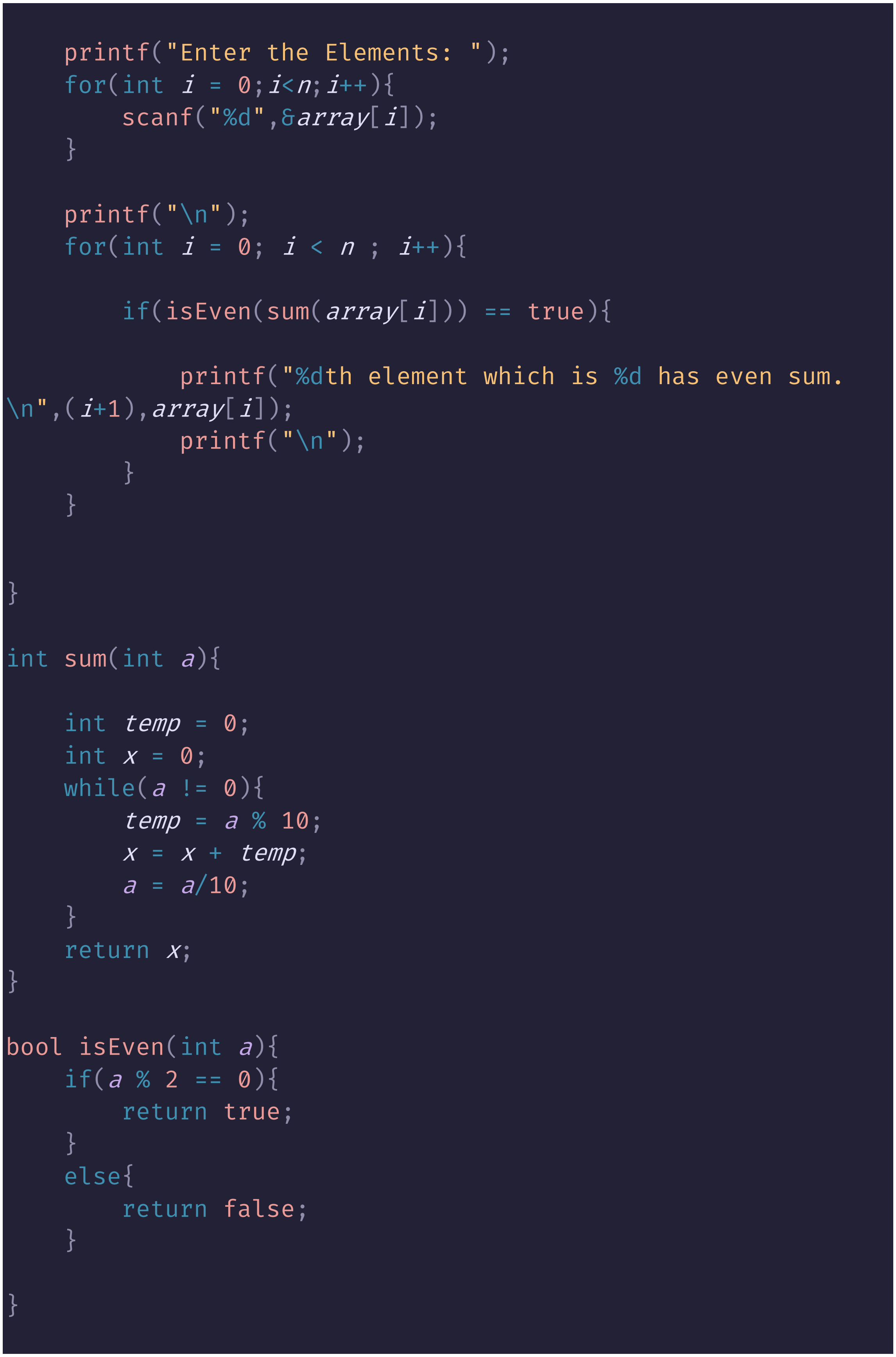
```
╭─ Siddharth via ◆ main at …\IT101\LAB6
╰─ ./a.exe

Enter Your Array Lenght: 4
Enter the Elements: 1 2 3 4

Your Reversed array ⟶  4 3 2 1
```

2. **c program to find numbers in an array whose digit sum is/are even.**

Code:

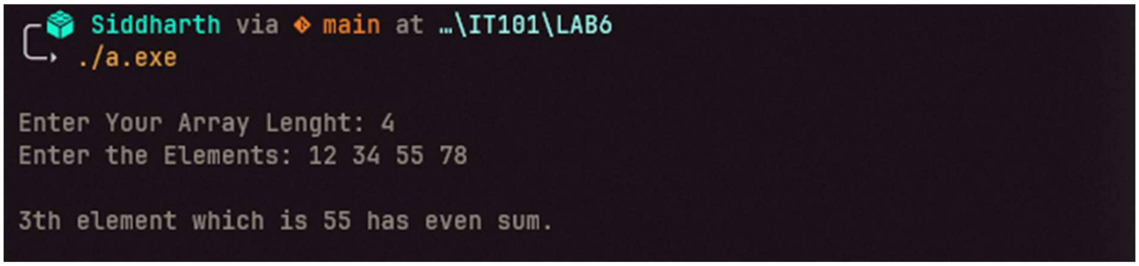


```c
#include <stdio.h>
#include <stdbool.h>

int sum(int a);

bool isEven(int a);

int main(){

    int n;
    printf("\nEnter Your Array Lenght: ");
    scanf("%d",&n);

    int array[n];
```



```
    printf("Enter the Elements: ");
    for(int i = 0;i<n;i++){
        scanf("%d",&array[i]);
    }

    printf("\n");
    for(int i = 0; i < n ; i++){

        if(isEven(sum(array[i])) == true){

            printf("%dth element which is %d has even sum.
\n",(i+1),array[i]);
            printf("\n");
        }
    }


}

int sum(int a){

    int temp = 0;
    int x = 0;
    while(a != 0){
        temp = a % 10;
        x = x + temp;
        a = a/10;
    }
    return x;
}

bool isEven(int a){
    if(a % 2 == 0){
        return true;
    }
    else{
        return false;
    }

}
```

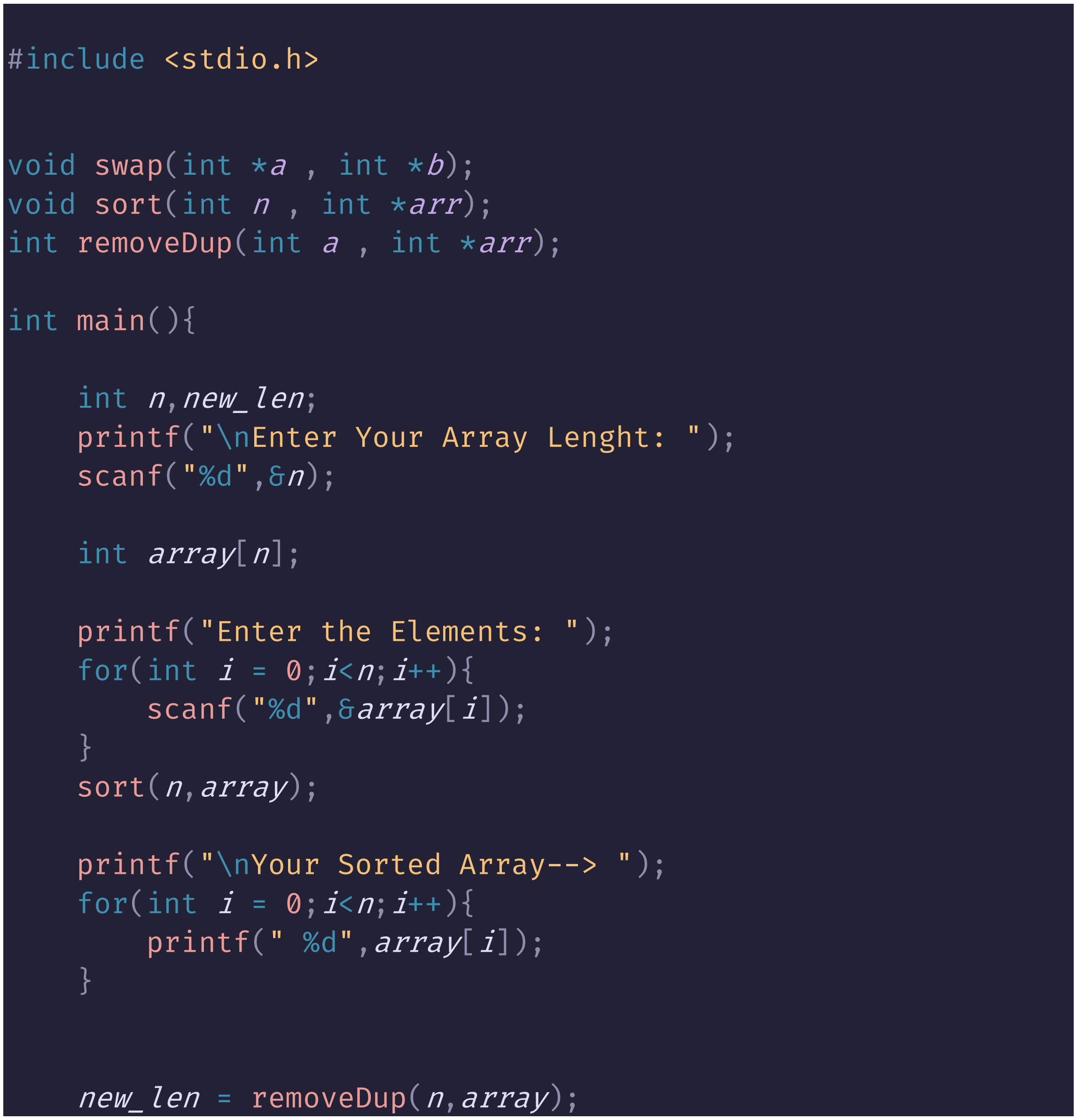
Output:



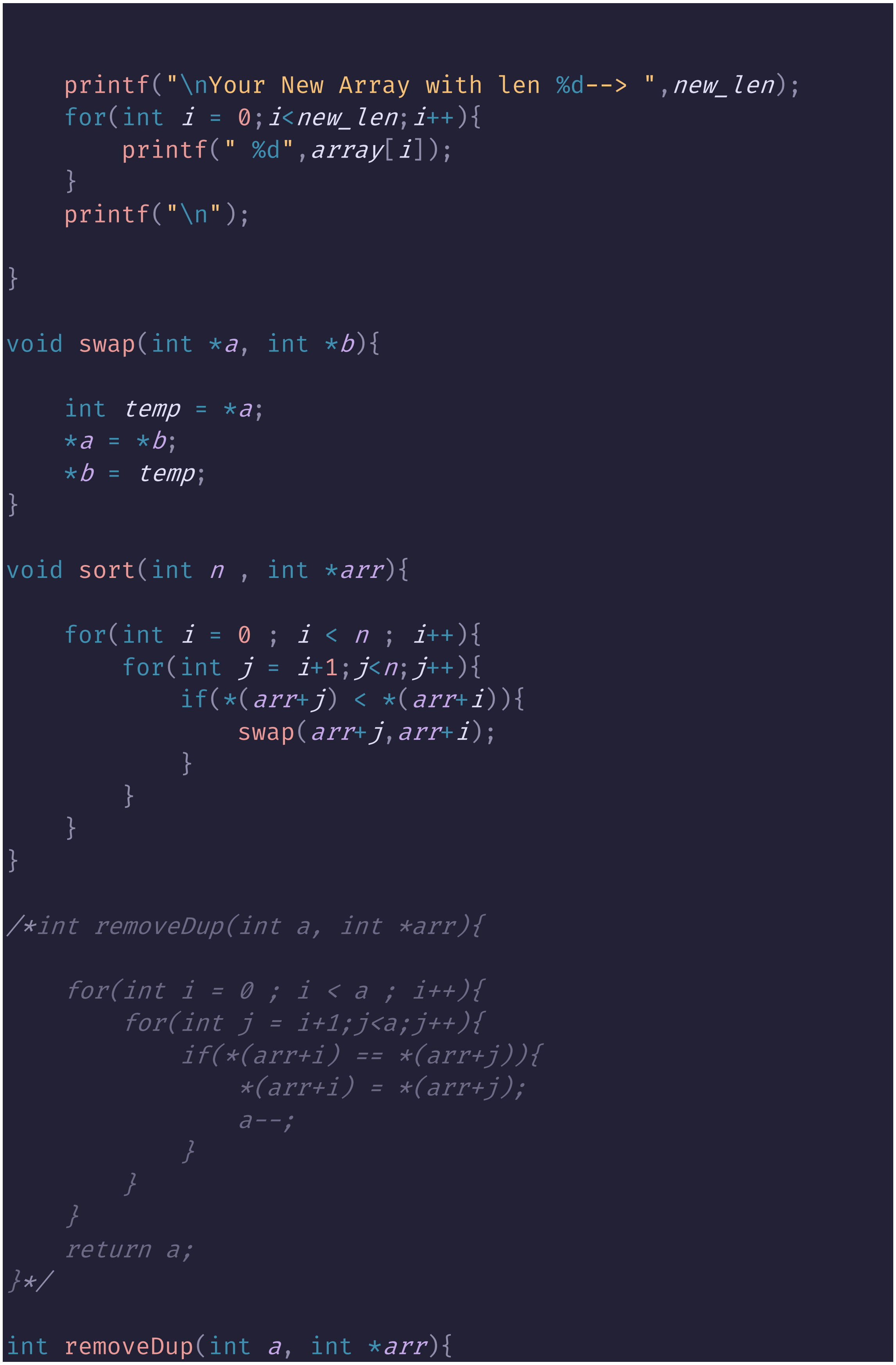
```
Siddharth via main at …\IT101\LAB6
./a.exe

Enter Your Array Lenght: 4
Enter the Elements: 12 34 55 78

3th element which is 55 has even sum.
```

3. **C function to remove duplicates from an array using pointers.**

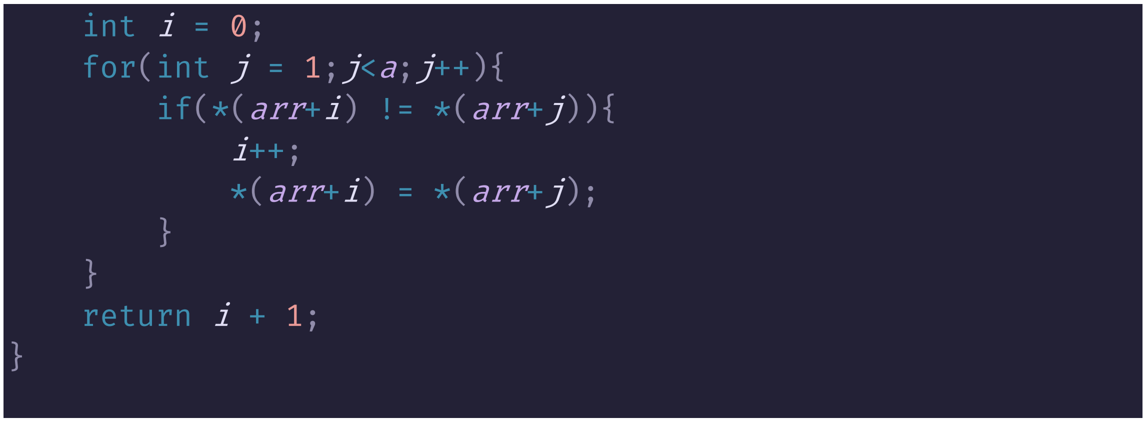
Code:



```c
#include <stdio.h>


void swap(int *a , int *b);
void sort(int n , int *arr);
int removeDup(int a , int *arr);

int main(){

    int n,new_len;
    printf("\nEnter Your Array Lenght: ");
    scanf("%d",&n);

    int array[n];

    printf("Enter the Elements: ");
    for(int i = 0;i<n;i++){
        scanf("%d",&array[i]);
    }
    sort(n,array);

    printf("\nYour Sorted Array--> ");
    for(int i = 0;i<n;i++){
        printf(" %d",array[i]);
    }


    new_len = removeDup(n,array);
```

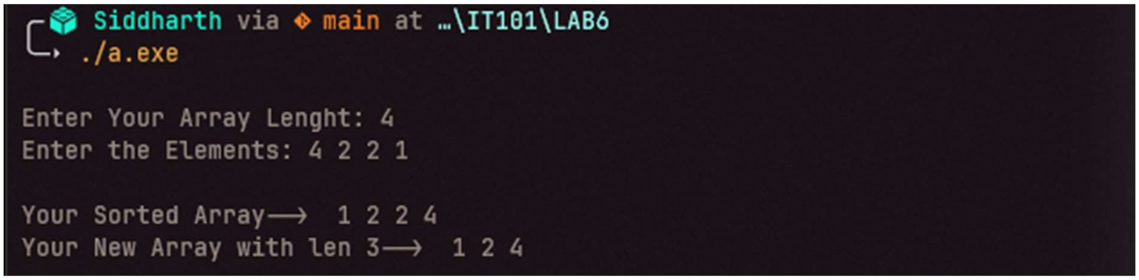


```
    printf("\nYour New Array with len %d--> ",new_len);
    for(int i = 0;i<new_len;i++){
        printf(" %d",array[i]);
    }
    printf("\n");

}

void swap(int *a, int *b){

    int temp = *a;
    *a = *b;
    *b = temp;
}

void sort(int n , int *arr){

    for(int i = 0 ; i < n ; i++){
        for(int j = i+1;j<n;j++){
            if(*(arr+j) < *(arr+i)){
                swap(arr+j,arr+i);
            }
        }
    }
}

/*int removeDup(int a, int *arr){

    for(int i = 0 ; i < a ; i++){
        for(int j = i+1;j<a;j++){
            if(*(arr+i) == *(arr+j)){
                *(arr+i) = *(arr+j);
                a--;
            }
        }
    }
    return a;
}*/

int removeDup(int a, int *arr){
```



```c
    int i = 0;
    for(int j = 1;j<a;j++){
        if(*(arr+i) != *(arr+j)){
            i++;
            *(arr+i) = *(arr+j);
        }
    }
    return i + 1;
}
```

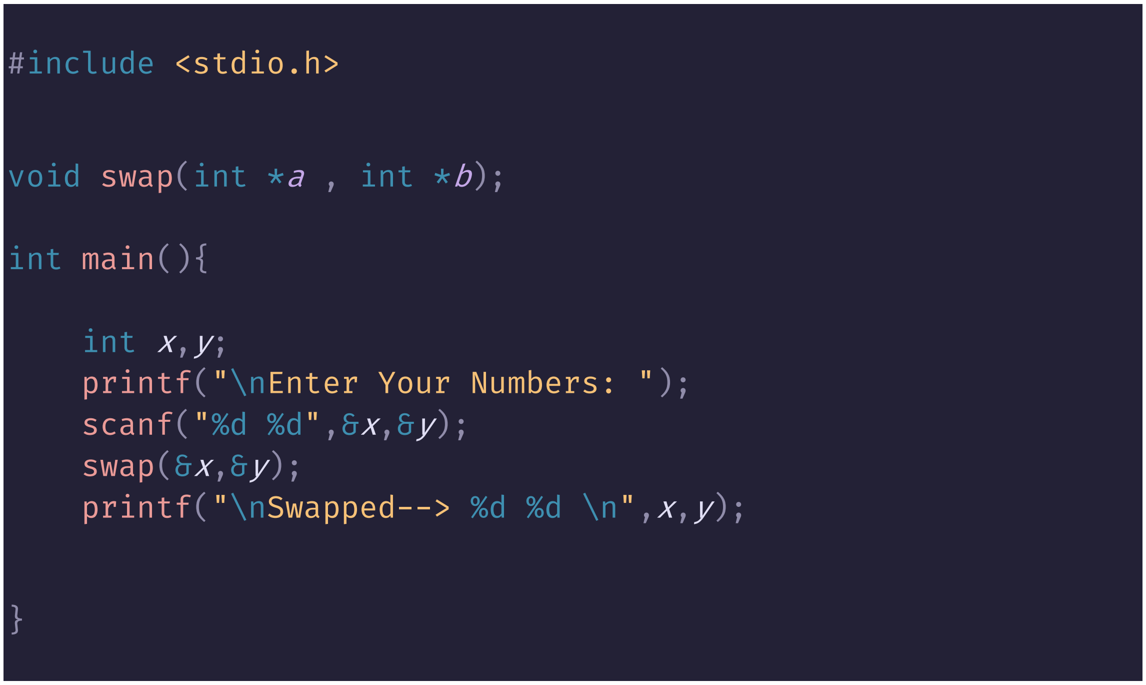
Output:



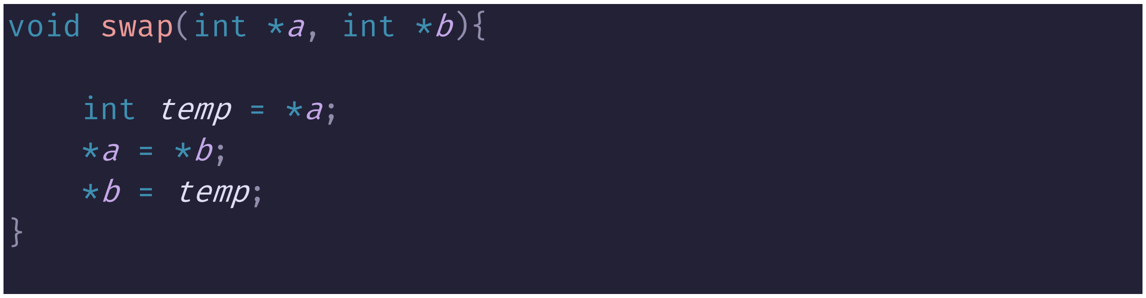
```
╭ Siddharth via ◆ main at …\IT101\LAB6
╰› ./a.exe

Enter Your Array Lenght: 4
Enter the Elements: 4 2 2 1

Your Sorted Array⟶  1 2 2 4
Your New Array with len 3⟶  1 2 4
```

4. **C program to swap two values using call by reference.**

Code:

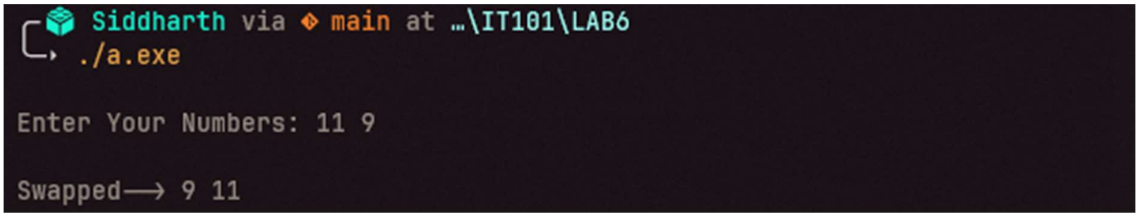


```c
#include <stdio.h>


void swap(int *a , int *b);

int main(){

    int x,y;
    printf("\nEnter Your Numbers: ");
    scanf("%d %d",&x,&y);
    swap(&x,&y);
    printf("\nSwapped--> %d %d \n",x,y);


}
```



```c
void swap(int *a, int *b){

    int temp = *a;
    *a = *b;
    *b = temp;
}
```

Output:

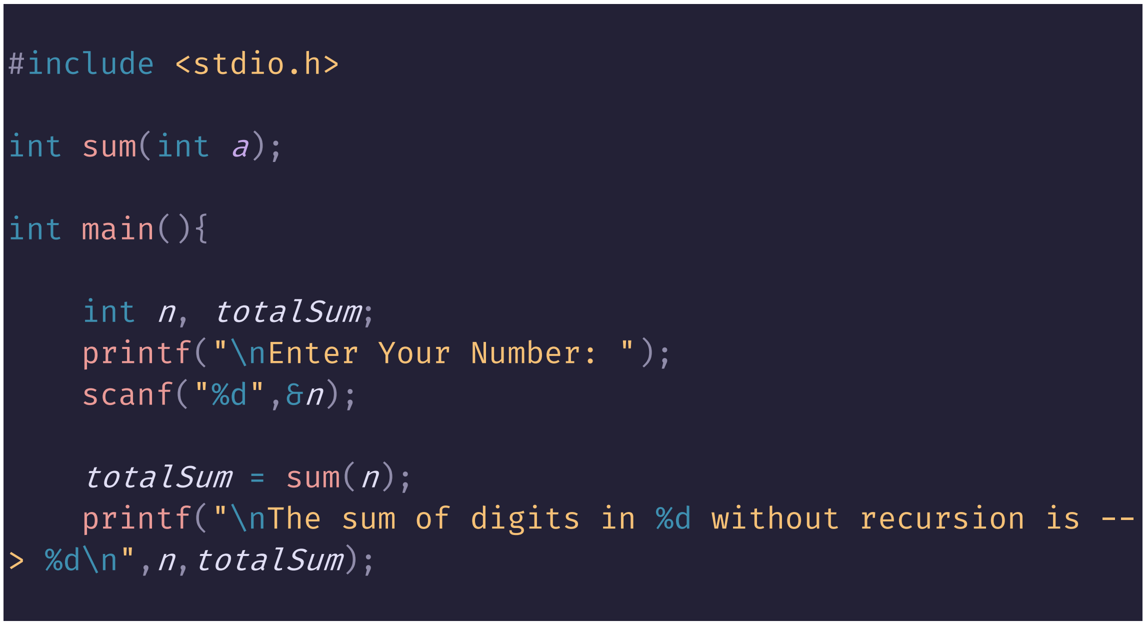


```
╭─ Siddharth via ◆ main at …\IT101\LAB6
╰─› ./a.exe

Enter Your Numbers: 11 9

Swapped--> 9 11
```

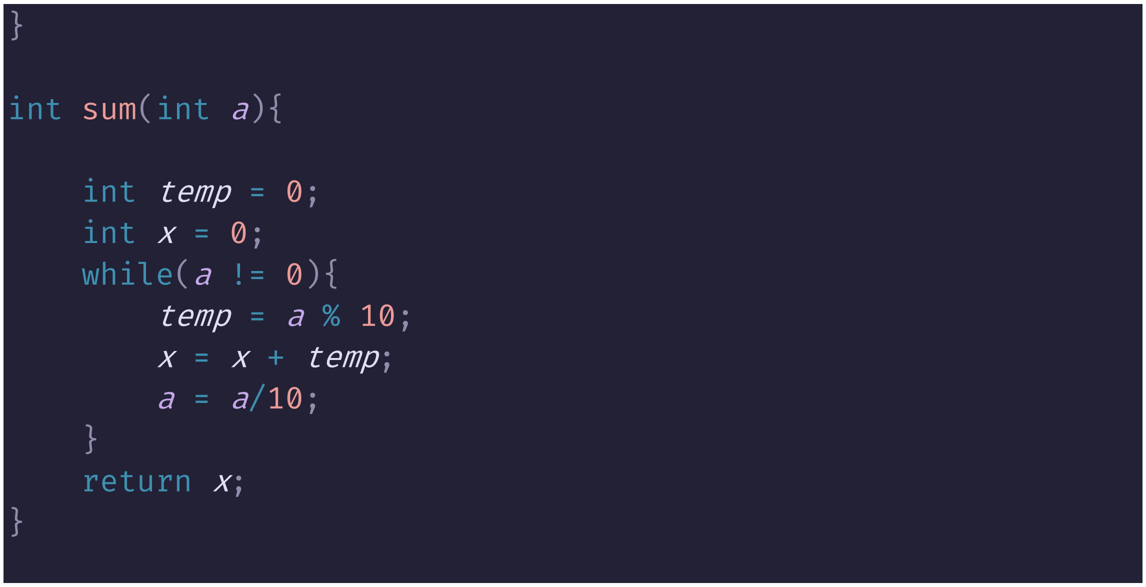
5. **A 5-digit positive integer is entered through the keyboard, write a function to calculate sum of digits of the 5-digit number:**

a) **Without Recursion**

Code:

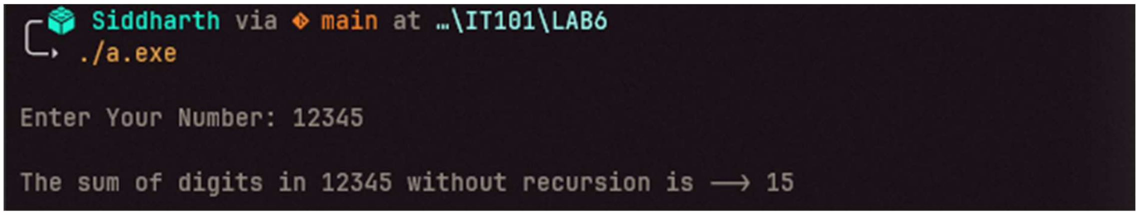


```c
#include <stdio.h>

int sum(int a);

int main(){

    int n, totalSum;
    printf("\nEnter Your Number: ");
    scanf("%d",&n);

    totalSum = sum(n);
    printf("\nThe sum of digits in %d without recursion is --
> %d\n",n,totalSum);
```



```c
}

int sum(int a){

    int temp = 0;
    int x = 0;
    while(a != 0){
        temp = a % 10;
        x = x + temp;
        a = a/10;
    }
    return x;
}
```

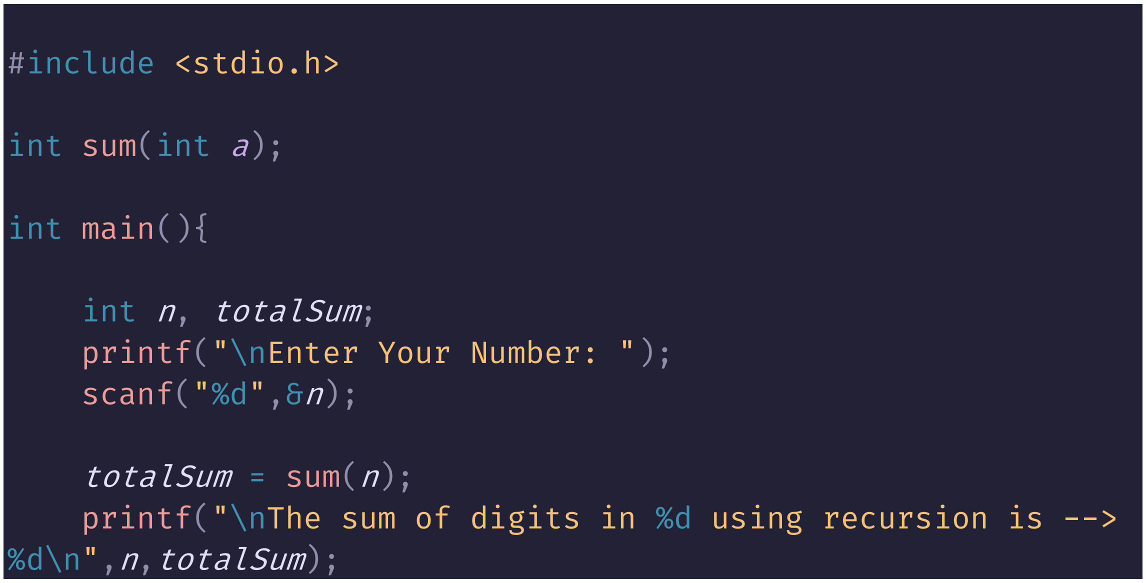
Output:



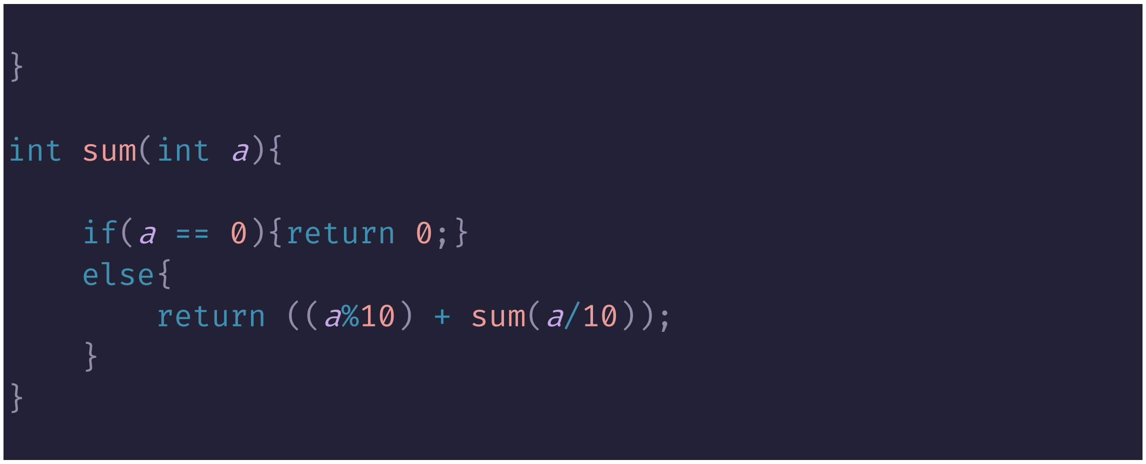
```
╭ Siddharth via ◆ main at …\IT101\LAB6
╰→ ./a.exe

Enter Your Number: 12345

The sum of digits in 12345 without recursion is ⟶ 15
```

b) With Recursion

Code:

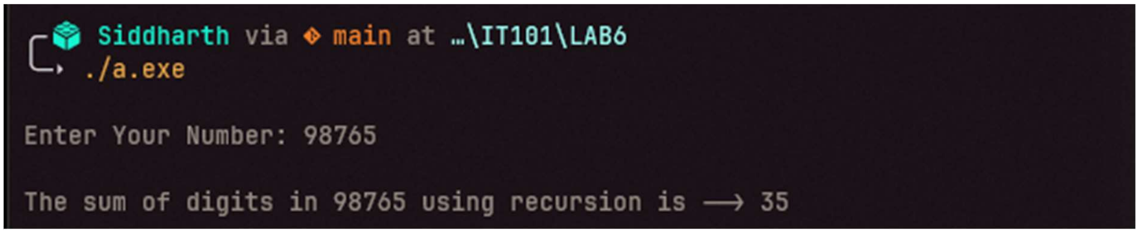


```c
#include <stdio.h>

int sum(int a);

int main(){

    int n, totalSum;
    printf("\nEnter Your Number: ");
    scanf("%d",&n);

    totalSum = sum(n);
    printf("\nThe sum of digits in %d using recursion is -->
%d\n",n,totalSum);
```



```c
}

int sum(int a){

    if(a == 0){return 0;}
    else{
        return ((a%10) + sum(a/10));
    }
}
```

Output:



```
Siddharth via ◆ main at …\IT101\LAB6
./a.exe

Enter Your Number: 98765

The sum of digits in 98765 using recursion is ⟶ 35
```