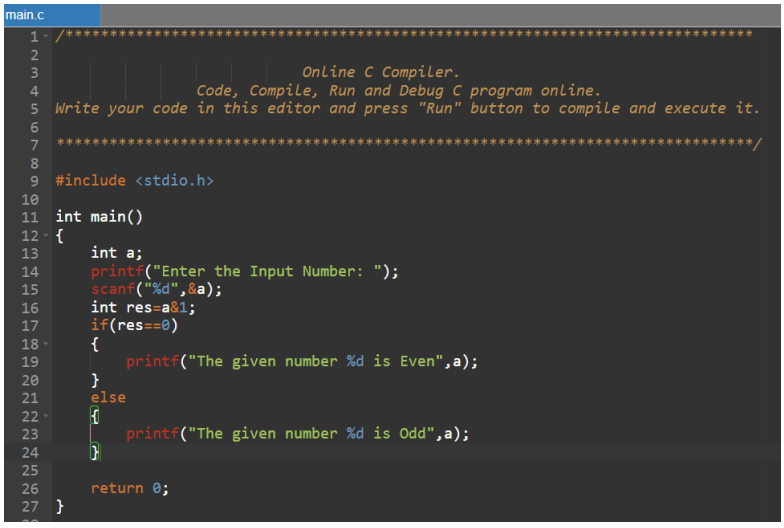
ASSIGNMENT 1

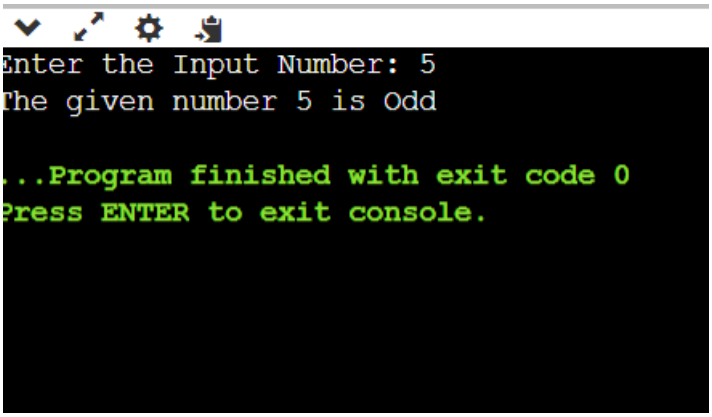
1.Write a C program to determine the given number is odd or even using Bitwise operators.

Solution:

```
#include<stdio.h>
int main()
{
int a;
printf(“Enter the Input Number: “);
scanf(“%d”,&a);
int res=a&1;
if(res==0)
{
printf(“The given number %d is Even”,a);
}
else
{
printf(“The given number %d is Odd”,a);
}
return 0;
}
```

RESULTS:





2. Write a C program to count the number of bits set in a number.

Input:

144

Output:

Count of Set bits: 2

Solution:

```
#include<stdio.h>
int main()
{
int num;
printf("Enter the input Number:");
scanf("%d",&num);
int count=0;
while(num>=1)
{
int rem=num%2;
if(rem==1)
{
count++;
}
num=num/2;
}
printf("The Number of bits set in a number us %d ",count);

return 0;
}
```

RESULTS:

main.c

```c

int main()
{
    int num;
    printf("Enter the input number:");
    scanf("%d",&num);
    int count=0;
    while(num>=1)
    {
        int rem=num%2;
        if(rem==1)
        {
            count++;

        }
        num=num/2;;
    }
    printf("The Number of bits set in a number is %d",count);
    return 0;
}

```

```
Enter the input number:15
The Number of bits set in a number is 4

...Program finished with exit code 0
Press ENTER to exit console.
```

3.Write a C program to swap two numbers. Use a function pointer to do this operation.

Input:

84 25

Output:

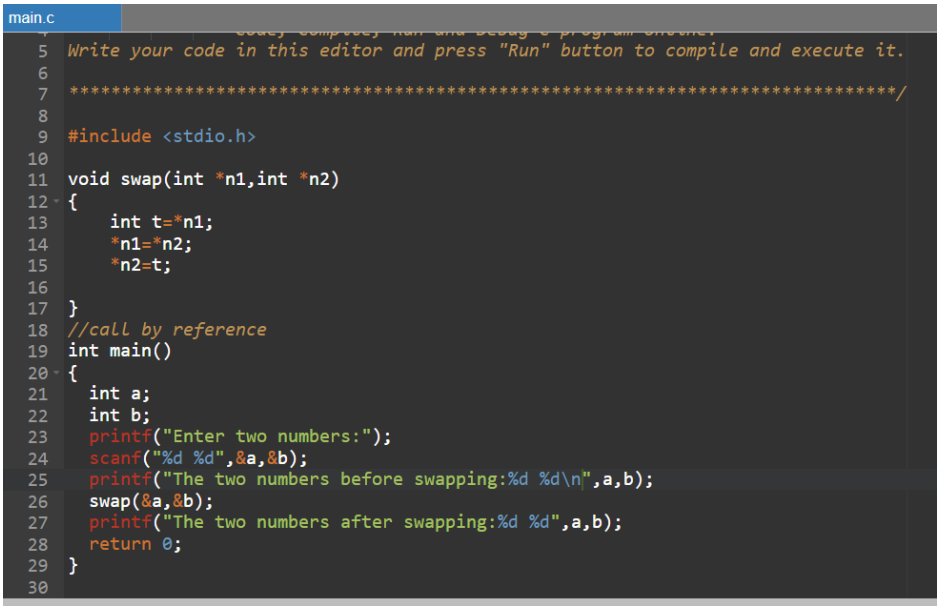
25 84

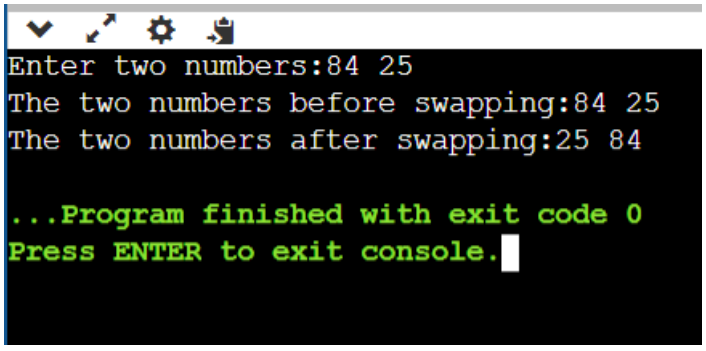
Solution:

```
#include<stdio.h>
void swap(int *n1,int *n2)
{
int t=*n1;
*n1=*n2;
*n2=t;
}
```

```
int main()
{
int a;
int b;
printf("Enter two numbers:");
scanf("%d %d",&a,&b);
printf("The two numbers before swapping:%d %d\n",a,b);
swap(&a,&b);
printf("The two numbers after swapping:%d %d",a,b);
return 0;
}
```

RESULTS:





4.Write an equivalent pointer expression for fetching the value of array element a[i][j][k][2]

Solution:

((*(*(a + i) + j) + k) +2)

5.Write a C program to Multiply two matrix (n*n) using pointers.

Input:	**Output:**
Size of Row: 3	Product:
Size of Column: 3	48 39 30
Matrix 1:	102 84 66
2 3 4	129 111 93
5 6 7	
8 9 1	
Matrix 2:	
9 8 7	
6 5 4	
3 2 1	

Solution:

```
#include<stdio.h>

void MatrixMultiplication(int A[][50], int r, int c, int B[][50],int C[][50])
{

 int i, j, k, sum = 0;

 if (c != r) {
        printf("Order of Input Matricis is Invalid");
        return;
 }

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

```
                sum = 0;
                for (k = 0; k < r; ++k) {
                        sum += *(*(A + i) + k) * *(*(B + k) + j);
                }
                *(*(C + i) + j) = sum;
        }
 }

}

int main() {

 int i, j;
 int r,c;
 int A[50][50],B[50][50],C[50][50];
 printf("Enter Number of Rows: ");
 scanf("%d", &r);
 printf("Enter Number of columns: ");
 scanf("%d", &c);
 printf("Enter A: \n");
 for (i = 0; i < r; ++i) {
        for (j = 0; j < c; ++j) {
                scanf("%d", &A[i][j]);
        }
 }


 printf("Enter B: \n");
 for (i = 0; i < r; ++i) {
        for (j = 0; j < c; ++j) {
                scanf("%d", &B[i][j]);
        }
 }
 MatrixMultiplication(A, r, c,B,C);
 printf("\nThe Matrix Multiplication of A and B: \n");
 for (i = 0; i < r; ++i) {
        for (j = 0; j < c; ++j) {
                printf("%d ", C[i][j]);
        }
        printf("\n");
 }

}
```

RESULTS:

main.c

```c
/******************************************************************************

                              Online C Compiler.
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/
#include<stdio.h>

void MatrixMultiplication(int A[][50], int r, int c, int B[][50],int C[][50])
{

    int i, j, k, sum = 0;

    if (c != r) {
        printf("Order of Input Matricis is Invalid");
        return;
    }

    for (i = 0; i < r; ++i) {
        for (j = 0; j < c; ++j) {
            sum = 0;
            for (k = 0; k < r; ++k) {
                sum += *(*(A + i) + k) * *(*(B + k) + j);
            }
            *(*(C + i) + j) = sum;
        }
    }

}
```

```
Enter Number of Rows: 3
Enter Number of columns: 3
Enter A:
2 3 4
5 6 7
8 9 1
Enter B:
9 8 7
6 5 4
3 2 1

The Matrix Multiplication of A and B:
48 39 30
102 84 66
129 111 93
```

6. Find the output of the following // Consider the compiler is 32-bit machine

```
#include <stdio.h>
typedef struct
{
int A;
char B;
char C;
} InfoData;
int main(int argc, char *argv[])
{
//Calculate size of structure
printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
return 0;
}
```

Solution:

Size of Structure=8

7.Find the output of the following // Consider the compiler is 32-bit machine

```
#include <stdio.h>
typedef struct
{
char A;
double B;
char C;
} InfoData;
int main(int argc, char *argv[])
{
//Calculate size of structure
printf("\n Size of Structure = %d\n\n",sizeof(InfoData));
return 0;
}
```

Solution:

Size of Structure=24

8.Find the output of the following // Consider the compiler is 32-bit machine

```
#include <stdio.h>
#include <stdint.h>
int main()
{
   unsigned int  var = 0x12345678;
   unsigned int  rev = 0;
   for (int j = 0; j < 8; j++)
   {
      rev = (rev << 4) | ((var >> (4 * j)) & 0xF);
   }
   printf("%X", rev);
   return 0;
}
```

Solution:

87654321