

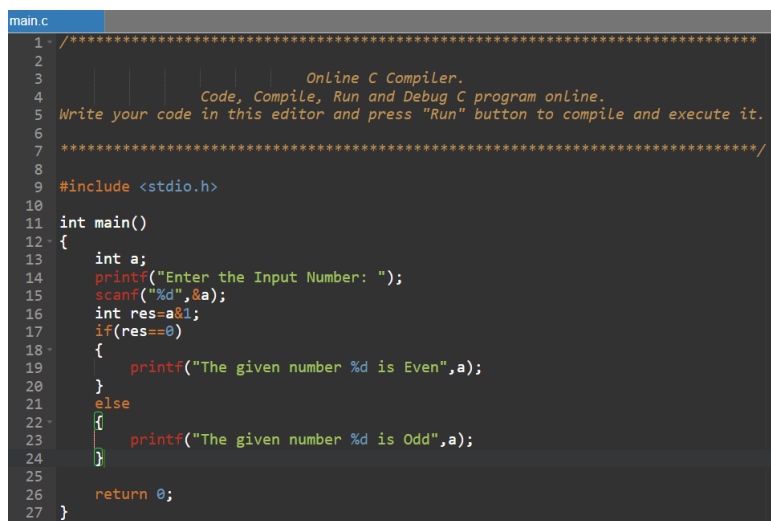
ASSIGNMENT 1

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

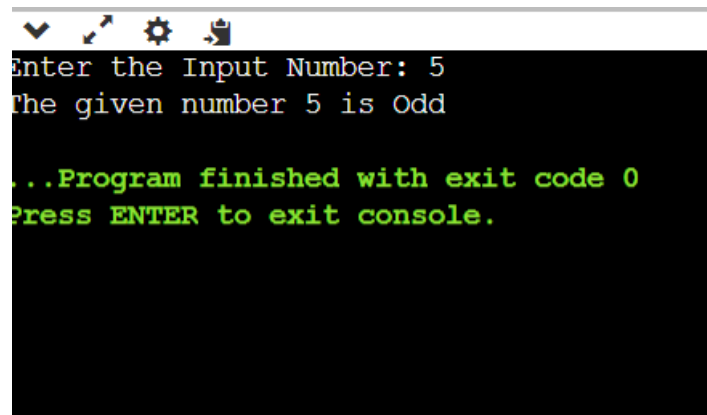
Solution:

```
#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:

A screenshot of an online C compiler interface. The top bar shows 'main.c'. The code area contains the same C program as shown in the previous block. The output area at the bottom shows the program's execution: it prompts 'Enter the Input Number: ', receives the input '1', and outputs 'The given number 1 is Odd'.

```
1- 1- *****
2- 2-
3- 3- Online C Compiler.
4- 4- Code, Compile, Run and Debug C program online.
5- 5- Write your code in this editor and press "Run" button to compile and execute it.
6- 6- *****
7- 7-
8- 8-
9- 9- #include <stdio.h>
10- 10-
11- 11- int main()
12- 12- {
13- 13-     int a;
14- 14-     printf("Enter the Input Number: ");
15- 15-     scanf("%d",&a);
16- 16-     int res=a&1;
17- 17-     if(res==0)
18- 18-     {
19- 19-         printf("The given number %d is Even",a);
20- 20-     }
21- 21-     else
22- 22-     {
23- 23-         printf("The given number %d is Odd",a);
24- 24-     }
25- 25-
26- 26-     return 0;
27- 27- }
```



```
Enter the Input Number: 5
The given number 5 is Odd

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

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
10 #include <stdio.h>
11 int main()
12 {
13     int num;
14     printf("Enter the input number:");
15     scanf("%d",&num);
16     int count=0;
17     while(num>=1)
18     {
19         int rem=num%2;
20         if(rem==1)
21         {
22             count++;
23         }
24         num=num/2;;
25     }
26     printf("The Number of bits set in a number is %d",count);
27     return 0;
28 }
29
```

```
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:

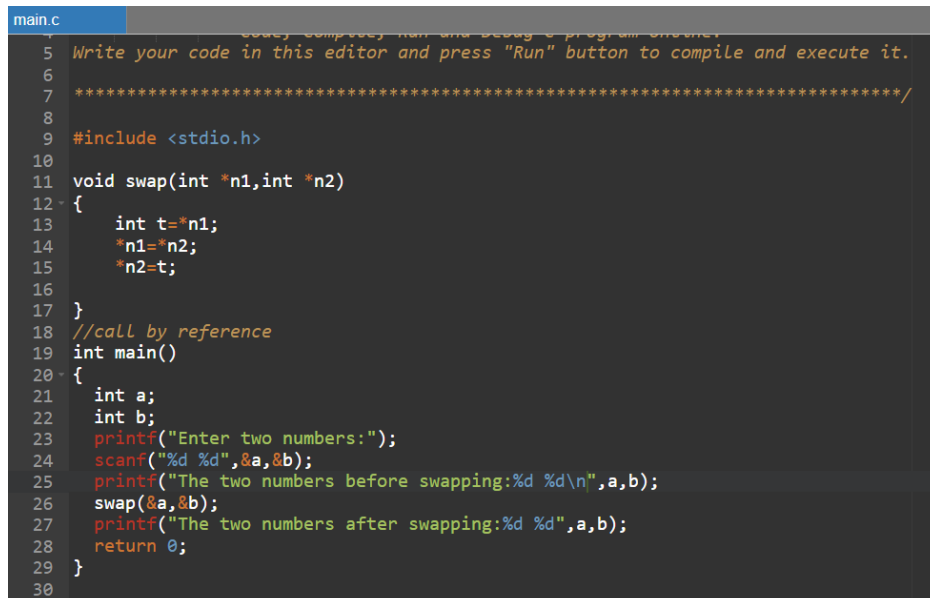
```
#include<stdio.h>
void swap(int *n1,int *n2)
{
    int t=*n1;
    *n1=*n2;
    *n2=t;
}
```

```

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;
}

```

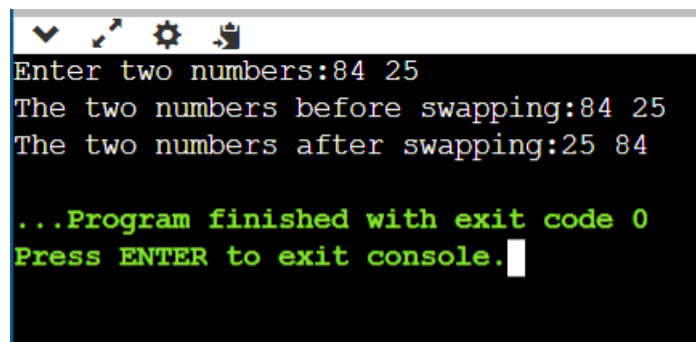
RESULTS:



```

main.c
5  Write your code in this editor and press "Run" button to compile and execute it.
6
7  *****/
8
9  #include <stdio.h>
10
11 void swap(int *n1,int *n2)
12 {
13     int t=*n1;
14     *n1=*n2;
15     *n2=t;
16 }
17
18 //call by reference
19 int main()
20 {
21     int a;
22     int b;
23     printf("Enter two numbers:");
24     scanf("%d %d",&a,&b);
25     printf("The two numbers before swapping:%d %d\n",a,b);
26     swap(&a,&b);
27     printf("The two numbers after swapping:%d %d",a,b);
28     return 0;
29 }
30

```



```

Enter two numbers:84 25
The two numbers before swapping:84 25
The two numbers after swapping:25 84

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

```

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:

Size of Row: 3

Size of Column: 3

Matrix 1:

2 3 4

5 6 7

8 9 1

Matrix 2:

9 8 7

6 5 4

3 2 1

Output:

Product:

48 39 30

102 84 66

129 111 93

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
1  /******
2
3      OnLine C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to compile and execute it.
6
7      *****/
8  #include<stdio.h>
9
10 void MatrixMultiplication(int A[][50], int r, int c, int B[][50],int C[][50])
11 {
12
13     int i, j, k, sum = 0;
14
15     if (c != r) {
16         printf("Order of Input Matricis is Invalid");
17         return;
18     }
19
20     for (i = 0; i < r; ++i) {
21         for (j = 0; j < c; ++j) {
22             sum = 0;
23             for (k = 0; k < r; ++k) {
24                 sum += *(A + i) + k) * (*(B + k) + j);
25             }
26             (*(C + i) + j) = sum;
27         }
28     }
29 }
30 }
```

```
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 i = 0; i < 8; i++)
    {
        rev = (rev << 4) | ((var >> (4 * i)) & 0xF);
    }
    printf("%X", rev);
    return 0;
}
```

Solution:

87654321