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

```
#include <stdio.h>
int main() {
   int num;
   printf("Enter the number: ");
   scanf("%d",&num);
   if(num&0x001)
   { printf("It is an Odd Number");}
   else
   { printf("It is an Even Number");}

   return 0;
}
```

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

Input:

144

Output:

Count of Set bits: 2

```
#include <stdio.h>

int main() {
   int num;
   printf("Enter the number: ");
   scanf("%d",&num);
   int count=0;
   int num1=num;
   while(num)
   { count+=num&1;
     num=num>>1;}
   printf("Number of bits set in the number %d is %d",num1,count);

   return 0;
}
```

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

Input:

84 25

Output:

25 84

```
#include <stdio.h>
int swapnum(int, int);
int main() {
    int num, num1;
    printf("Enter the numbers: ");
    scanf("%d %d", &num, &num1);
    int (*fp)(int, int);
    fp= swapnum;
    fp(num, num1);
    return 0; }

int swapnum(int a,  int b)
   { int temp;
     temp= a;
     a= b;
     b= temp;
     printf("The swapped numbers are %d %d", a, b); }
```

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

```
*(*(*(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
```

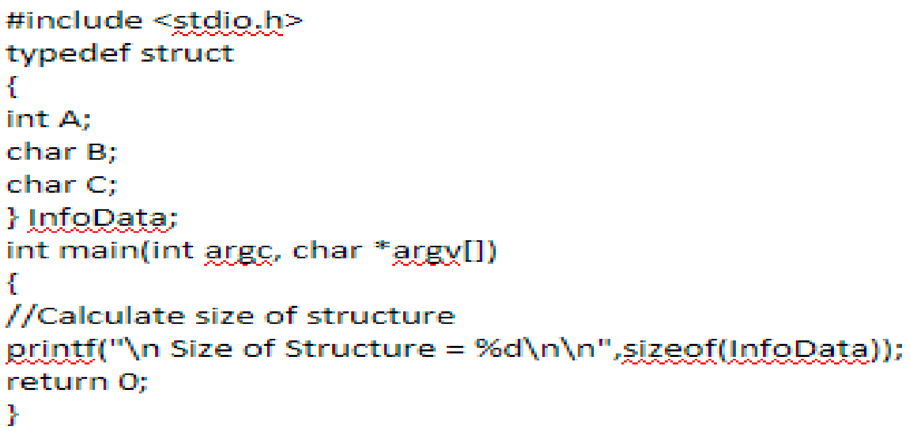
```
#include <stdio.h>
int multiply(int m1[],int m2[]);
int main() {
  int row,col;
  printf("Enter the number of rows: ");
  scanf("%d",&row);
  printf("Enter the number of columns: ");
  scanf("%d",&col);
  int m1[row][col],m2[row][col],res[row][col];
  printf("Enter m1 elements one by one");
  for(int i=0;i<row;i++)
  { for(int j=0;j<row;j++)
     {scanf("%d",(*(m1+i)+j));}}

  printf("Enter m2 elements one by one");
  for(int i=0;i<row;i++)
{ for(int j=0;j<row;j++)
     {scanf("%d",(*(m2+i)+j));}}

  //Performing multiplication
  for(int i=0;i<row;i++)
  { for(int j=0;j<col;j++)
  { int sum=0;
    for(int k=0;k<col;k++)
     {sum+=(*(*(m1+i)+k))*(*(*(m2+k)+j));}
     *(*(res + i) + j) = sum;}}

  //Printing the result matrix
  printf("The result matrix elements are");
  for(int i=0;i<row;i++)
  { for(int j=0;j<row;j++)
     {printf(" %d ",(*(*(res+i)+j)));}
     printf("\n");
  }
  return 0;}
```

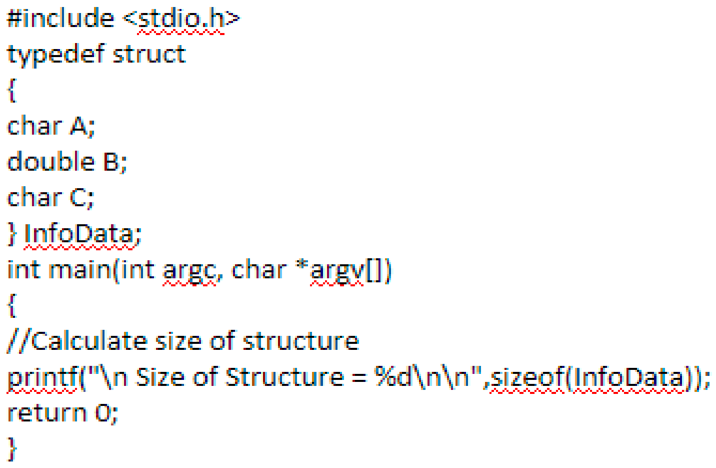
6. Find the output of the following // Consider the compiler is 32-bit machine



```
#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;
}
```

Size of Structure = 8

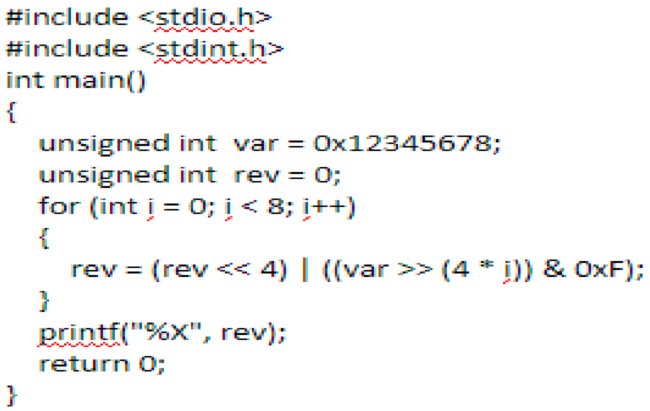
7. Find the output of the following // Consider the compiler is 32-bit machine



```
#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;
}
```

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;
}
```

Output : 87654321