**Assignment - 1**

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

Code:

#include <stdio.h>

int main()

{

int n;

printf("Enter a number: ");

scanf("%d", &n);

if (n & 1)

{

printf("odd");

} else

{

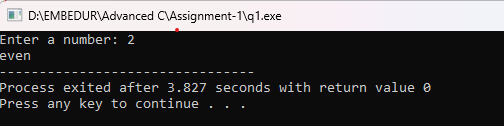
printf("even");

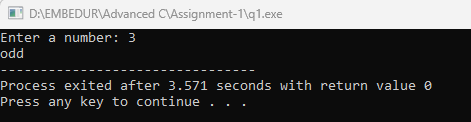
}

return 0;

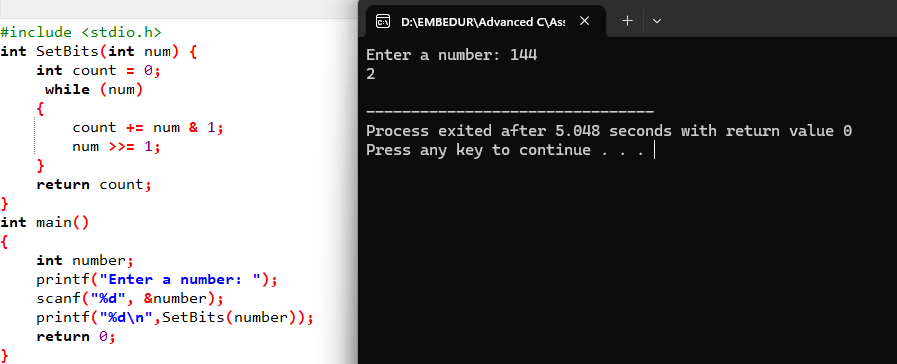
}

Output:





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



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

Code:

#include <stdio.h>

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

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main()

{

int m, n;

printf("Enter the first number: ");

scanf("%d", &m);

printf("Enter the second number: ");

scanf("%d", &n);

swap(&m, &n);

printf("After swapping:\n");

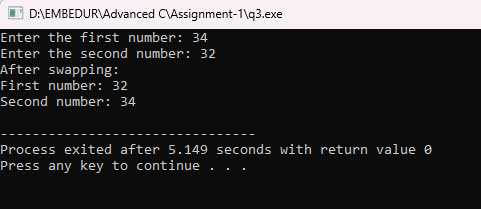
printf("First number: %d\n", m);

printf("Second number: %d\n", n);

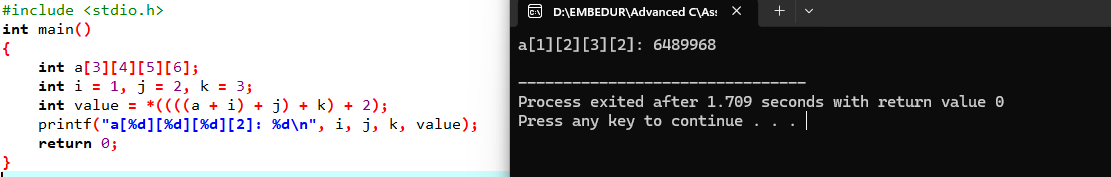
return 0;

}

Output:



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



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

#include <stdio.h>

#define s 10

void multiplyMatrices(int (\*mat1)[s], int (\*mat2)[s], int (\*result)[s], int n)

{

int i, j, k;

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

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

\*(\*(result + i) + j) = 0;

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

\*(\*(result + i) + j) += \*(\*(mat1 + i) + k) \* \*(\*(mat2 + k) + j);

}

}

}

}

void displayMatrix(int (\*mat)[s], int n)

{

int i, j;

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

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

printf("%d ", \*(\*(mat + i) + j));

}

printf("\n");

}

}

int main()

{

int size;

printf("Enter the size of the square matrices: ");

scanf("%d", &size);

int matrix1[s][s], matrix2[s][s], result[s][s];

printf("Enter elements of the first matrix:\n");

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

printf("Enter element m1[%d][%d]: ", i, j);

scanf("%d", &matrix1[i][j]);

}

}

printf("Enter elements of the second matrix:\n");

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

printf("Enter element m2[%d][%d]: ", i, j);

scanf("%d", &matrix2[i][j]);

}

}

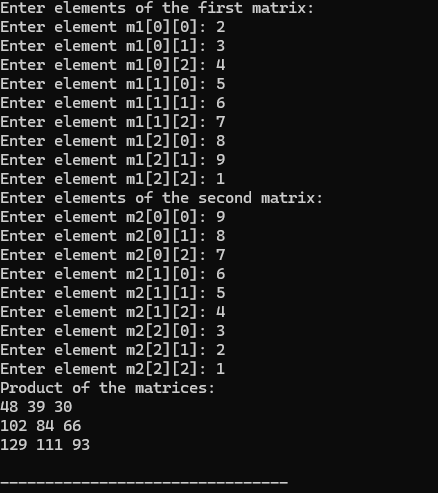
multiplyMatrices(matrix1, matrix2, result, size);

printf("Product of the matrices:\n");

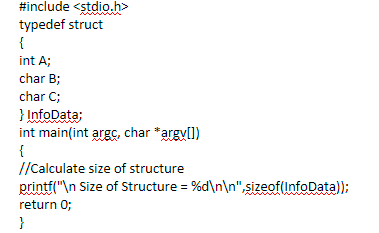
displayMatrix(result, size);

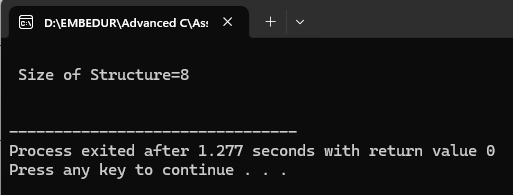
return 0;

}

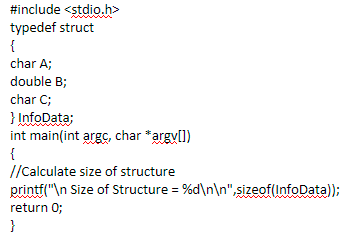


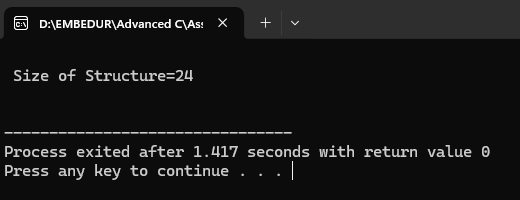
1. Find the output of the following





1. Find the output of the following





1. Find the output of the following

