**ADVANCED C PROGRAMMING – MODULE 1**

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

#include <stdio.h>

int main()

{

int n;

printf("enter a number=");

scanf("%d",&n);

if ((n&1)==0){

printf("even");

}

else{

printf("odd");

}

return 0;

}

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

#include <stdio.h>

int main()

{

int n,count=0;

printf("enter a number=");

scanf("%d",&n);

while(n){

count+= n&1;

n>>=1;

}

printf("Set bits = %d",count);

}

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

#include <stdio.h>

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

int main()

{

int a,b;

printf("enter value for a and b =");

scanf("%d %d",&a,&b);

printf("Before swapping\na=%d\nb=%d\n",a,b);

swap(&a,&b);

printf("After swapping\na=%d\nb=%d\n",a,b);

}

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

int temp;

temp=\*a;

\*a=\*b;

\*b=temp;

}

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

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

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

#include <stdio.h>

int main()

{

int n;

int A[100][100],B[100][100],C[100][100];

printf("Enter no of rows & coulmns = ");

scanf("%d",&n);

printf("\nEnter the elements for A matrix:\n");

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

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

printf("Enter A[%d][%d] = ",i,j);

scanf("%d",(\*(A+i)+j));

}

}

printf("\nEnter the elements for B matrix:\n");

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

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

printf("Enter B[%d][%d] = ",i,j);

scanf("%d",(\*(B+i)+j));

}

}

printf("\nMultiplication of Matrix:\n");

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

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

C[i][j]=0;

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

(\*(\*(C+i)+j)) = (\*(\*(C+i)+j)) + (\*(\*(A+i)+k)) \* (\*(\*(B+k)+j));

}

}

printf("\n");

}

printf("\nResultant Matrix:\n");

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

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

printf("%d ",C[i][j]);

}

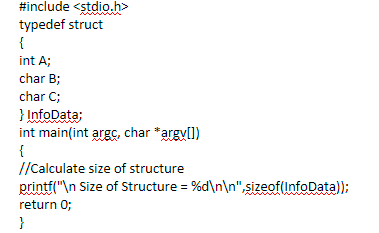
printf("\n");

}

return 0;

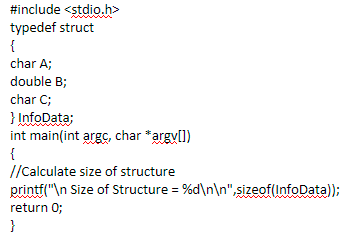
}

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

****

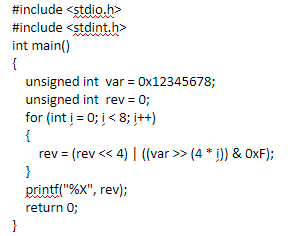
Output: 8

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

****

Output: 16

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

****

Output: 87654321