**Q01// REVERSE BITS OF A BINARY**

#include <stdio.h>

unsigned short int reverse\_bits(unsigned short int n){

unsigned short int a=0;

for(int i=0;i<16;i++){

a = a<<1;

a= a|(n&1);

n = n>>1;

}

return a;

}

void binary(unsigned short int a){

int array[16];

for(int x=0;x<16;x++){

array[x]=a%2;

a=a/2;

}

for(int j=15;j>=0;j--){

printf("%d",array[j]);

}

}

int main() {

printf("ENTER YOUR NUMBER\n");

unsigned short int num;

scanf("%hu",&num);

printf("%hu\n",num);

binary(num);

unsigned short int c = reverse\_bits(num);

printf("\n%hu\n",c);

binary(c);

return 0;

}

**Q02//COUNT SETBITS IN A 16 BIT INTEGER**

#include <stdio.h>

#include <math.h>

unsigned short int rep=0;

int binaryRep(unsigned short int n)

{

for(int i=0;i<16;i++)

{

rep+=(n&1)\*pow(10,i);

n=n>>1;

}

printf("%hu",rep);

return 0;

}

int main() {

unsigned short int n,m;

scanf("%hu",&n);

m=n;

binaryRep(m);

printf("\n\nYOU WANT TO COUNT BINARY SETBITS OF %d\n\n",n);

int count=0;

unsigned short int a;

for(int i=0;i<16;i++)

{

a=n&1;

n= n>>1;

if(a==1)

count+=1;

}

printf("NUMBER OF SETBITS = %hu\n\n",count);

return 0;

}

**Q03 //TO CHECK WHETHER A NUMBER IS PALINDROME OR NOT……**

**//4.Write a C program to check whether the entered number is palindrome or not**

**// reverse of 141 is 141**

#include <stdio.h>

int main() {

int num,arr[20],rem;

printf("Input the number you want to know about that is palindrome or not\n\n");

scanf("%d",&num);

printf("you enterd %d\n\n",num);

int j=0;

while(num!=0){

rem=num%10;

arr[j]=rem;

num=num/10;

j++;

}

for(int i=0; i<j/2;i++){

if(arr[i]==arr[j-i-1])

printf("IT IS A PALINDROME");

else

printf ("IT IS NOT A PALINDROME");

}

return 0;

}

**Q04//ARMSTRONG NUMBER**

//Write a C program to check whether a given number is an Armstrong Number of 3 digits or not. An Armstrong number is a number which is equal to the sum of the cubes of its own digits. E.g. 153 is an Armstrong number since 153 = 1\*1\*1 + 5\*5\*5+ 3\*3\*3

#include <stdio.h>

int main() {

int num,a[100],pro;

printf ("Input a number\n\n");

scanf("%d",&num);

printf("you entered %d\n\n",num);

pro=num;

int j=0;

while(num!=0){

a[j] = num%10;

num=num/10;

j++;

}

int q=0,p;

for(int i=0;i<j;i++)

{

p=a[i]\*a[i]\*a[i];

q=q+p;

}

if(q==pro)

printf("armstrong number");

else

printf("not an armstrong number");

return 0;

}

**Q05// LCM AND GCD OF TWO NUMBERS…**

**// Write a C program to find HCF and LCM of two numbers.**

#include <stdio.h>

int main() {

int n1,n2,x,y,LCM,HCF;

scanf("%d %d",&n1,&n2);

printf ("you want lcm and hcf of %d and %d\n\n",n1,n2);

x=n1; y=n2;

while(n1!=n2){

if (n1>n2)

n1=n1-n2;

else

n2=n2-n1;

}

HCF= n1;

printf("Hcf = %d\n",HCF);

LCM=(x\*y)/HCF;

printf ("Lcm = %d\n",LCM);

return 0;

}