Exercise 1:

#include<bits/stdc++.h>

using namespace std;

#include <stdlib.h>

int Power(int n, int p)

{

if (p<=1)return n;

else

return n\*Power(n, p-1);

}

int main()

{

int n,p,a=0;

cout<<"Enter the number:"<<endl;

cin>>n;

cout<<"Enter the power:"<<endl;

cin>>p;

a=Power(n,p);

cout<<"Output:"<<endl;

cout<<a<<endl;

}

Exercise 2:

#include<bits/stdc++.h>

using namespace std;

#include <stdlib.h>

int GCD(int x, int y)

{

while(x!=y)

{

if(x>y) return GCD(x-y,y);

else return GCD(x,y-x);

}

return x;

}

int main()

{

int x,y,a=0;

cout<<"Enter the first number:"<<endl;

cin>>x;

cout<<"Enter the second number:"<<endl;

cin>>y;

a=GCD(x,y);

cout<<"Output GCD:"<<endl;

cout<<a<<endl;

}

Exercise 3:

#include<bits/stdc++.h>

using namespace std;

#include <stdlib.h>

int m=0;

int LCM(int a, int b)

{

m+=b;

if((m%a==0) && (m%b==0))

return m;

else

LCM(a,b);

}

int main()

{

int x,y,a=0;

cout<<"Enter the first number:"<<endl;

cin>>x;

cout<<"Enter the second number:"<<endl;

cin>>y;

a=LCM(x,y);

cout<<"Output LCM:"<<endl;

cout<<a<<endl;

}

Exercise 4:

#include<bits/stdc++.h>

using namespace std;

#include <stdlib.h>

int n=0;

int Palindrome(string a)

{

int length= a.length()-(n+1);

if (a[n]==a[length])

{

if (n+1==length||n==length)

{

cout<<"Its palindrome."<<endl;

exit(0);

}

n=n+1;

Palindrome(a);

}

else

cout<<"It's not a palindrome."<<endl;

}

int main()

{

string x;

cout<<"Enter the number:"<<endl;

cin>>x;

Palindrome(x);

}

Exercise 5:

#include<bits/stdc++.h>

using namespace std;

#include <stdlib.h>

struct Node

{

int data;

struct Node\*next;

}\*head =NULL;

void CreateNodeList()

{

struct Node \*p, \*temp;

int data,n;

cout<<"Input total number of data:"<<endl;

cin>>n;

head=(struct Node\*)malloc(sizeof(struct Node));

if(head==NULL)

{cout<<"There is no data!!!!";}

else

{

cout<<"Input data for Node 1 :"<<endl;

cin>>data;

head->data=data;

head->next=NULL;

temp=head;

for(int i=2;i<=n;i++)

{

p =(struct Node\*)malloc(sizeof(struct Node));

cout<<"Input data for Node "<<i<<" :"<<endl;

cin>>data;

p->data=data;

p->next=NULL;

temp->next=p;

temp=temp->next;

}

}

cout<<endl;

}

void reverse(struct Node\* p)

{

if(p->next==NULL)

{

head=p;

return;

}

reverse(p->next);

struct Node\* rev=p->next;

rev->next =p;

p->next =NULL;

}

void PRINT()

{

struct Node\* p;

p=head;

while(p!=NULL)

{

cout<<p->data<<" ";

p=p->next;

}

cout<<endl;

}

int main()

{

CreateNodeList();

cout<<"Before Reverse:"<<endl;

PRINT();

reverse(head);

cout<<"\n\nAfter Reverse:"<<endl;

PRINT();

}