```
1) #include<iostream>
 Using namespace std;
void merge(int a[], int sa , int b[], int sb)
{     int h=0;
      for (int i=(sb-sa-1); i<sb ;i++)
      {       b[i] = a[h];
              h++;
      }
}
void delcomp (int a[] ,int sa)
{      for (int i=0 ; i<sa ; i++)
       {      for (int j=2 ; j< a[i]/2 ; j++)
              {    if(a[i] % j == 0)
                   {
```

(2)
```
#include <iostream>

Class linkedlist
{   node* getHead()
    {   return head;
    }
};

int main()
{       linkedlist a;
        linkedlist b;
        linkedlist c;
        node* A = a.getHead();
        node* B = b.getHead();
      while (!(A==NULL))
       {   if(A->data%2!=0)
           {   c.add_node(A->data);
           }
           A = A->next;
       }
     while(!(B==Null))
       {  if(B->data % 2==0)
           {  c.add_node(B->data);
           }
           B = B->next;
        }
       a.deletelist();
       b.deletelist();
       c.deletelist();
 }
```

```
(3) Class linkedlist
{   int listSize( )
    {   Node* temp = head;
        int c=0;
        while (!(temp == Null))
        {
              c++;
              temp = temp->next;
         }
        return c;
    }
    Void pal( )
    {   bool a = true;
        int size = list size();
        int n = size/2;
        ~~for (int i=0 ; i<n ;;i++)~~
        ~~{        if~~
        node *temp1 = head;
        node * temp2 = tail;
       for (int i=0 ; i<n ; i++)
        {
              if (temp1->data != temp2->data)
                {  ~~return false;~~
                }  a = false;
              temp1 = temp1->next;
              temp2 = temp2->Prev;
         }
        return a;
    }
};
```

```
int main()
{
      bool p = pal();
      if(P==true)
      { cout <<" Is Palindrome ";
       }
      else{ cout<< " Is not palindrome "; }
}
```