```
#include<iostream>
using namespace std;

struct node
{
      int info;
      struct node* next;
}; typedef struct node* nodeptr;

void insertbegin(nodeptr &plist)
{
      nodeptr p;
      p = new node;

      cout<<"Enter any number : ";
      cin>>p->info;

      p->next = plist;
      plist = p;
}

void insertend(nodeptr &plist)
{
      nodeptr q, p;
      q = plist;

      for( ; q->next != NULL; )
            q = q->next;

      p = new node;

      cout<<"Enter any number : ";
      cin>>p->info;

      p->next = NULL;
      q->next = p;
}

void ins_betwen(nodeptr &plist)
{
    nodeptr p, a, b;
    int i, L;

    cout << "Enter the number of location : ";
    cin >> L;

    p = new node;

    cout << "Enter any number : ";
    cin >> p->info;

    a = plist;

    for (i = 2; i < L; i++)
        a = a->next;

    b = a->next;
    a->next = p;
    p->next = b;
```

```
}

void deletebeg(nodeptr &plist)
{
      nodeptr p;

      p = plist;
      plist = plist->next;
      free(p);
}

void deleteend(nodeptr &plist)
{
      nodeptr q, p;
      q = plist;
      p = plist;

      for( ; q->next != NULL; )
      {
            p = q;
            q = q->next;
      }
      p->next = NULL;
      free(q);
}

void deletebet(nodeptr &plist)
{
      nodeptr p, q, a;
      int i, l;
      p = plist;
      q = plist;
      a = plist;

      cout << "Enter the number of location : ";
    cin >> l;

      for(i = 2; i < l; i++)
      {
            p = q;
            q = q->next;
            a = q->next;
      }

      q->next = a->next;

      free(a);
}

void displaying(nodeptr &plist)
{
      nodeptr q;

      if(plist == NULL)
            cout<<"\nList is Empty\n";
      else
      {
            q = plist;
```

```
            for( ; q != NULL; )
            {
                  cout<<q->info<<"\t";
                  q = q->next;
            }
            cout<<"\n";
      }
}

int main()
{
      nodeptr p, q, plist;
      int x;
      p = new node;

      cout<<"Enter any number : ";
      cin>>p->info;
      p->next = NULL;
      plist = p;

      cout<<endl;
      for( ; x != 8; )
      {
            cout<<"1- add from begin\n";
            cout<<"2- add from end\n";
            cout<<"3- add from between\n";
            cout<<"4- delete from begin\n";
            cout<<"5- delete from end\n";
            cout<<"6- delete from between\n";
            cout<<"7- print list\n";
            cout<<"8- exit\n";

            cout<<"Enter your choise : ";
            cin>>x;

            cout<<endl;
            switch(x)
            {
                  case 1:insertbegin(plist); break;
                  case 2:insertend(plist); break;
                  case 3:ins_betwen(plist); break;
                  case 4:deletebeg(plist); break;
                  case 5:deleteend(plist); break;
                  case 6:deletebet(plist); break;
                  case 7:displaying(plist); break;
                  default:cout<<"Error\n";
            }
      }

      return 0;
}
```

```
#include<iostream>
using namespace std;

int const size = 1;

struct node
{
      int info;
      struct node* next;
}; typedef struct node* nodeptr;

void insertbegin(nodeptr &plist)
{
      nodeptr p;
      p = new node;

      cout<<"Enter any number : ";
      cin>>p->info;

      p->next = plist;
      plist = p;
}

void insertend(nodeptr &plist)
{
      nodeptr q, p;
      q = plist;

      for( ; q->next != NULL; )
            q = q->next;

      p = new node;

      cout<<"Enter any number : ";
      cin>>p->info;

      p->next = NULL;
      q->next = p;
}

void ins_betwen(nodeptr &plist)
{
    nodeptr p, a, b;
    int i, L;

    cout << "Enter the number of location : ";
    cin >> L;

    p = new node;

    cout << "Enter any number : ";
    cin >> p->info;

    a = plist;

    for (i = 2; i < L; i++)
        a = a->next;

    b = a->next;
```

```
    a->next = p;
    p->next = b;
}

void deletebeg(nodeptr &plist)
{
      nodeptr p;

      p = plist;
      plist = plist->next;
      free(p);
}

void deleteend(nodeptr &plist)
{
      nodeptr q, p;
      q = plist;
      p = plist;

      for( ; q->next != NULL; )
      {
            p = q;
            q = q->next;
      }
      p->next = NULL;
      free(q);
}

void deletebet(nodeptr &plist)
{
      nodeptr p, q, a;
      int i, l;
      p = plist;
      q = plist;
      a = plist;

      cout << "Enter the number of location : ";
    cin >> l;

      for(i = 2; i < l; i++)
      {
            p = q;
            q = q->next;
            a = q->next;
      }

      q->next = a->next;

      free(a);
}

void prime(nodeptr &plist)
{
      nodeptr q;
      int x, i, j=1, l = 0;

      q = plist;

      for( ; q != NULL; )
```

```
	{
		if(q->info != 1)
		{
			x = 0;
			for(i = 2; i < q->info; i++)
			{
				if(q->info % 2 == 0)
					x = 1;
			}

			if(x == 0)
			{
				cout<<q->info<<"\t";
				++l;
			}
		}

		else
		{
			cout<<q->info<<"\t";
			++l;
		}

		q = q->next;
	}

	cout<<endl<<"Number numbers prime :"<<l<<endl<<endl;
}

void sum_num(nodeptr &plist)
{
	nodeptr q;
	int sum = 0, i = 0;

	q = plist;

	for( ; q != NULL; )
	{
		if(q->info % 5 == 0)
		{
			sum += q->info;
			++i;
		}
		q = q->next;
	}
	cout<<"Sum number %5 == 0 : "<<sum<<"\n";
	cout<<"Number numbers %5 == 0 : "<<i<<"\n";
}

void con_arr(nodeptr &plist, int array[size], int &y)
{
	nodeptr q;
	int i = 0, z = 0;
	q = plist;


	for( ; q != NULL; )
	{
		if(q->info % 2 == 0)
```

```
            {
                  ++z;
                  if(z > 1)
                        ++y;

                  array[i] = q->info;
                  ++i;
            }
            q = q->next;
      }

      for(i = 0; i <= y; i++)
            cout<<array[i]<<"\t";
      cout<<endl;
}

void fact(nodeptr &plist)
{
      nodeptr q;
      int i, fact = 1, maxamimm;

      q = plist;

      maxamimm = q->info;

      for( ; q != NULL; )
      {
            if(maxamimm < q->info)
                  maxamimm = q->info;

            q = q->next;
      }

      for(i = 1; i <= maxamimm; i++)
      {
            fact = fact * i;
      }

      cout<<"Fact ( "<<maxamimm<<" ) : "<<fact<<"\n";
}

void number(nodeptr &plist)
{
      nodeptr q;
      int i = 0;
      q = plist;

      for( ; q != NULL; )
      {
            ++i;
            q = q->next;
      }

      cout<<"number nods : "<<i<<"\n\n";
}

void displaying(nodeptr &plist)
{
      nodeptr q;
```

```
	if(plist == NULL)
		cout<<"\nList is Empty\n";
	else
	{
		q = plist;

		for( ; q != NULL; )
		{
			cout<<q->info<<"\t";
			q = q->next;
		}
		cout<<"\n";
	}
}

int main()
{
	nodeptr p, q, plist;
	int x, y = 0;
	int array[size + y];
	p = new node;

	cout<<"Enter any number : ";
	cin>>p->info;
	p->next = NULL;
	plist = p;

	cout<<endl;
	for( ; x != 13; )
	{
		cout<<"1- add from begin\n";
		cout<<"2- add from end\n";
		cout<<"3- add from between\n";
		cout<<"4- delete from begin\n";
		cout<<"5- delete from end\n";
		cout<<"6- delete from between\n";
		cout<<"7- prime list elem\n";
		cout<<"8- Sum number %5 == 0 and number him list elem\n";
		cout<<"9- convert even to array\n";
		cout<<"10- fact maximamm list\n";
		cout<<"11- numbers node in list\n";
		cout<<"12- print list\n";
		cout<<"13- exit\n";

		cout<<"Enter your choise : ";
		cin>>x;

		cout<<endl;
		switch(x)
		{
			case 1:insertbegin(plist); break;
			case 2:insertend(plist); break;
			case 3:ins_betwen(plist); break;
			case 4:deletebeg(plist); break;
			case 5:deleteend(plist); break;
			case 6:deletebet(plist); break;
			case 7:prime(plist); break;
			case 8:sum_num(plist); break;
```

```
                case 9:con_arr(plist, array, y); break;
                case 10:fact(plist); break;
                case 11:number(plist); break;
                case 12:displaying(plist); break;
                default:cout<<"Error\n";
            }
        }

        return 0;
}
```

```
#include<iostream>
using namespace std;

int const size = 1;

struct node
{
      int info;
      struct node* next;
}; typedef struct node* nodeptr;


void insertend(nodeptr &plist)
{
      nodeptr q, p;
      int i;
      q = plist;

      for(i = 0; i <= 12; i++)
      {
            for( ; q->next != NULL; )
            q = q->next;

            p = new node;

            cout<<"Enter any number : ";
            cin>>p->info;

            p->next = NULL;
            q->next = p;
      }

}

void spilt(nodeptr &plist)
{
      nodeptr q, p, head1, head2, a, b, c;
      int i;
      a = plist;

      cout<<"List 1 : \n";
      p = new node;
      p->info = a->info;
      p->next = NULL;
      head1 = p;
      b = head1;
      cout<<p->info<<"\t";

      a = a->next;
      for(i = 0; i <= 5; i++)
      {
            for( ; b->next != NULL; )
                        b = b->next;

            c = new node;
            c->info = a->info;
            cout<<c->info<<"\t";
            c->next = NULL;
            b->next = c;
```

```
            a = a->next;
      }

      cout<<"\nList 2 : \n";
      nodeptr b1, c1;
      b1 = head2;

      q = new node;
      q->info = a->info;
      q->next = NULL;
      head2 = q;
      cout<<q->info<<"\t";

      a = a->next;

      for(i = 7; i <= 12; i++)
      {
            for( ; b1->next != NULL; )
                        b1 = b1->next;

            c1 = new node;
            c1->info = a->info;
            cout<<c1->info<<"\t";
            c1->next = NULL;
            b1->next = c1;

            a = a->next;
      }


}

void displaying(nodeptr &plist)
{
      nodeptr q;

      if(plist == NULL)
            cout<<"\nList is Empty\n";
      else
      {
            q = plist;

            for( ; q != NULL; )
            {
                  cout<<q->info<<"\t";
                  q = q->next;
            }
            cout<<"\n";
      }
}

int main()
{
      nodeptr p, q, plist;
      int x, y = 0;
      int array[size + y];
      p = new node;
```

```
	cout<<"Enter any number : ";
	cin>>p->info;
	p->next = NULL;
	plist = p;

	cout<<endl;
	insertend(plist);

	cout<<endl;
	displaying(plist);

	cout<<endl;
	spilt(plist);


	return 0;
}
```