#include<bits/stdc++.h>

using namespace std;

struct Node{

int data;

struct Node \*next;

struct Node \*prev;

};

struct Node \*head = NULL;

void Insert(int position,int x)

{

struct Node \*newNode=new Node();

if(!newNode)

{

cout<<"Memory Limit Exceeded!!\n";

return;

}

newNode->data=x;

if(position==1)

{

newNode->prev=NULL;

newNode->next=head;

if(head)

head->prev=newNode;

head=newNode;

return;

}

else

{

struct Node \*temp=head;

int k=1;

while(k<(position-1) && (temp->next)!=NULL) //IMPORTANT!!!!

{

k++;

temp=temp->next;

}

if(k!=(position-1)) //IMPORTANT!!!!

{

cout<<"Desired position does not exist!!\n";

return;

}

newNode->next=temp->next;

newNode->prev=temp;

if(newNode->next)

newNode->next->prev=newNode;

temp->next=newNode;

}

}

void Delete(int position)

{

if(head==NULL)

{

cout << "List is empty!!\n";

return;

}

if(position==1)

{

struct Node \*temp=head;

head=head->next;

if(head->next)

head->prev=NULL;

delete(temp);

return;

}

else

{

struct Node \*temp=head;

int k=1;

while(k<(position) && temp!=NULL)

{

k++;

temp=temp->next;

}

if(temp==NULL)

{

cout<<"Desired position does not exist!!\n";

return;

}

struct Node \*temp2=temp->prev;

temp2->next=temp->next;

if(temp->next)

temp->next->prev=temp2;

free(temp);

}

}

void Print()

{

struct Node \*p=head;

while(p)

{

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

p=p->next;

}

cout<<"\n";

}

int main()

{

Insert(1,5);

Insert(1,4);

Insert(3,3);

Insert(2,2);

Insert(1,1);

Print();

Delete(1);

Print();

Delete(4);

Print();

Delete(5);

Print();

Delete(2);

Print();

}