**1. Write a C++ program that uses functions to perform the following:**

**a) Create a singly linked list of integers.**

**b) Delete a given integer from the above linked list.**

**c) Display the contents of the above list after deletion.**

#include<iostream>

using namespace std;

class node

{

public:

int data;

node \*next;

};

class sll

{

node \*head,\*tail;

public:

sll();

void insertAtStart(int);

void insertAtEnd(int);

void insertAtPos(int,int);

void deleteNode(int);

void deleteList();

void display();

void isListExist();

};

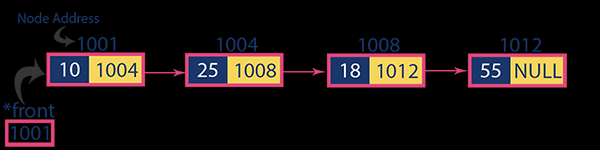
sll::sll()

{

head=NULL;

tail=NULL;

}



void sll::insertAtStart(int data)

{

node \*ptr;

ptr =new node;//1001 1002 1003

ptr->data = data; //10 20 30

if (head == NULL)//NULL==NULL 1001==NULL 1002==NULL

{

head = ptr;

30 NULL

20 NULL

10 NULL

tail = ptr;

ptr->next=NULL;// 10||NULL

return;

PTR, TAIL

,

PTR, HEAD

}

if (pos == i + 1)//(2==1+1🡺T)

{

ptr->next = temp->next;1001

temp->next = ptr;

if (ptr->next == NULL)

tail = ptr;

ptr->next = head; 20||1001 10||NULL

head = ptr;

}

void sll::insertAtEnd(int data)//30

{

node \*ptr = new node;//1003

ptr->data = data;//30

ptr->next = NULL;//30||NULL

if (tail == NULL)//NULL==NULL 1001==NULL

{

head = ptr;

tail = ptr;

}

tail->next = ptr;

tail = ptr;

}

void sll::insertAtPos(int pos, int data)//2 40

{

node \*ptr,\*temp;

ptr = new node;//1004

int i=1;

ptr->data = data;//40

ptr->next = NULL;//40||NULL

if (head == NULL || pos == 1) //1001==NULL||2==1(F||F=F)

{

if(!head)

{

head = ptr;

tail = ptr;

return;

}

ptr->next = head;

head = ptr;

return;

}

temp = head;1001

while (temp)

{

if (pos == i + 1)//(2==1+1🡺T)

{

ptr->next = temp->next;1001

temp->next = ptr;

if (ptr->next == NULL)

tail = ptr;

break;

}

i++;

temp = temp->next;

}

}

void sll::deleteNode(int data)30

{

node \*ptr, \*temp;

int res = 0;

ptr = head;

if (ptr->data == data)10==30

{

if (ptr->next == NULL)

{

delete ptr;

head = tail = NULL;

}

head = ptr->next; 1004

delete ptr;//1001

return;

}

while (ptr != NULL && ptr->next != NULL)

{

if (ptr->next->data == data)40==40

{

temp = ptr->next;

ptr->next = temp->next;

if (ptr->next == NULL)

tail = ptr;

delete temp;

res = 1;

}

ptr = ptr->next;

}

if (!res)

cout<<"Operation failed - Give data unavailable in list\n";

}

void sll::deleteList()

{

node \*ptr;

ptr = head;

while (ptr)

{

head = ptr->next;

delete ptr;

ptr = head;

}

}

void sll::display()

{

node \*ptr;

ptr = head;

while (ptr)

{

cout<<ptr->data<<endl;

ptr = ptr->next;

}

}

void sll::isListExist()

{

if (head)

cout<<"List is available\n";

else

cout<<"List is unavailable\n";

}

Int main()

{

sll ob;//object for class sll

int ch, data, pos, result;

cout<<"1. Insertion at the start of List\n";

cout<<"2. Insert at the end of list\n";

cout<<"3. Insert at node at the given position\n";

cout<<"4. Delete node\n";

cout<<"5. Delete list\n";

cout<<"6. display\n";

cout<<"7. Is list exists\n";

cout<<"8. Exit\n";

while(1)

{

cout<<"Enter ur choice:";

cin>>ch;

switch (ch)

{

case 1:

cout<<"Enter data to insert into list\n";

cin>>data;//10

ob.insertAtStart(data);//10

break;

case 2:

cout<<"Enter data to insert into list\n";

cin>>data;

ob.insertAtEnd(data);

break;

case 3:

cout<<"Enter value for position and data\n";

cin>>pos>>data;

ob.insertAtPos(pos, data);

break;

case 4:

cout<<"Enter value to delete node\n";

cin>>data;

ob.deleteNode(data);

break;

case 5:

ob.deleteList();

break;

case 6:

ob.display();

break;

case 7:

ob.isListExist();

break;

case 8:

exit(0);

default:

cout<<"Please retry once again\n";

break;

}

}

}