**3. a) Write a C program using dynamic variables and pointers, to construct a singly linked list. The operations to be supported are:**

**i) Insert at the front of a list.**

**ii) Insert at any position in the list.**

**iii) Deleting a node based on specific value.**

**iv) searching a node based on specified value.**

**iv) Displaying all the nodes in the list.**

#include<stdio.h>

#include<stdlib.h>

typedef struct node

{

int data;

struct node \*next;

}NODE;

NODE \*insertfront(NODE \*start,int val)

{

NODE \*newnode=(NODE \*)malloc(sizeof(NODE));

newnode->data=val;

newnode->next=start;

start=newnode;

return start;

}

NODE \*insertpos(NODE \*start,int val,int pos)

{

NODE \*ptr=start,\*temp;

int count=1;

NODE \*newnode=(NODE \*)malloc(sizeof(NODE));

newnode->data=val;

if(start==NULL && pos==1)

{

newnode->next=NULL;

start=newnode;

}

else if(start==NULL && pos>1)

printf("Invalid Position");

else

{

while(ptr!=NULL && count!=pos)

{

temp=ptr;

ptr=ptr->next;

count++;

}

if(ptr==NULL && count>pos)

printf("Invalid position\n");

else

{

temp->next=newnode;

newnode->next=ptr;

}

}

return start;

}

NODE \*deletesv(NODE \*start,int key)

{

NODE \*ptr=start,\*temp;

if(start==NULL)

{

printf("List is empty\n");

}

if(start->data==key)

{

start=start->next;

printf("%d is deleted\n",key);

free(ptr);

}

else

{

while(ptr!=NULL && ptr->data!=key)

{

temp=ptr;

ptr=ptr->next;

}

if(ptr==NULL)

printf("Invalid value\n");

else

{

temp->next=ptr->next;

printf("%d Value deleted from the list",key);

free(ptr);

}

}

return start;

}

NODE \*search(NODE \*start, int key)

{

if(start==NULL)

printf("List s empty\n");

else

{

NODE \*ptr=start;

while(ptr!=NULL && ptr->data!=key)

{

ptr=ptr->next;

}

if(ptr==NULL)

printf("Key not in list\n");

else

printf("%d is present the list",key);

}

return start;

}

void display(NODE \*start)

{

if(start==NULL)

printf("List is empty");

else

{

NODE \*ptr=start;

printf("List data are\n:");

while(ptr!=NULL)

{

printf("->%d",ptr->data);

ptr=ptr->next;

}

}

}

int main()

{

NODE \*start=NULL;

int key,val,pos,choice;

while(1)

{

printf("\nEnter your choice\n");

printf("1.Insert at front\n");

printf("2.Insert at position\n");

printf("3.Delete a node of specified value\n");

printf("4.Search a key\n");

printf("5.Display\n");

printf("6.Exit\n");

scanf("%d",&choice);

switch(choice)

{

case 1:printf("Enter the value to insert:\n");

scanf("%d",&val);

start=insertfront(start,val);

break;

case 2:printf("Enter the value to insert:\n");

scanf("%d",&val);

printf("Enter the position to insert:\n");

scanf("%d",&pos);

start=insertpos(start,val,pos);

break;

case 3:printf("Enter the key to delete:\n");

scanf("%d",&key);

start=deletesv(start,key);

break;

case 4:printf("Enter key to be search:\n");

scanf("%d",&key);

start=search(start,key);

break;

case 5:display(start);

break;

case 6:exit(0);

}

}

}

**OUTPUT**

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

1

Enter the value to insert:

10

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

1

Enter the value to insert:

20

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

1

Enter the value to insert:

30

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

5

List data are

:->30->20->10

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

2

Enter the value to insert:

50

Enter the position to insert:

2

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

5

List data are

:->30->50->20->10

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

3

Enter the key to delete:

50

50 Value deleted from the list

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit

5

List data are

:->30->20->10

Enter your choice

1.Insert at front

2.Insert at position

3.Delete a node of specified value

4.Search a key

5.Display

6.Exit