**Q. WAP to dynamically implement a singly linked list.**

**Source Code:**

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

void insert(int, int);

void delete(int);

void display();

struct node

{

int data;

struct node \*next\_data;

};

struct node \*head = NULL;

struct node \*ptr = NULL;

struct node \*temp = NULL;

struct node \*newNode;

int main()

{

char choice, position\_char;

int data, loop = 1;

while (loop == 1)

{

system("cls");

printf("\n1 -> insert. \n2 -> delete. \n3 -> display. \nESC -> Exit.");

choice = getch();

switch (choice)

{

case 27:

exit(13);

break;

case '1':

system("cls");

printf("insert at : \n1-> Beginning \n2-> End \n3-> Custom Position");

position\_char = getch();

printf("\n\nData to insert = ");

scanf("%d", &data);

insert(data, position\_char-48);

break;

case '2':

system("cls");

if (head == NULL){

printf("\nNO LIST\n");

getch();

break;

}

printf("delete at : \n1-> Beginning \n2-> End \n3-> Custom Position");

position\_char = getch();

delete(position\_char-48);

break;

case '3':

display();

getch();

break;

default:

loop = 1;

}

}

return 0;

}

void display()

{

system("cls");

ptr = head;

while(ptr != NULL)

{

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

ptr = ptr->next\_data;

}

}

void insert(int n, int pos)

{

newNode = (struct node\*) malloc(sizeof(struct node)); // dynamically create a new node

newNode->data = n; // add the data to the new node.

int i = 1, position;

//placing the node in the singly linked list.

switch (pos)

{

case 1:

newNode->next\_data = head; // previously 1st node is now pointed to by the newly made first node.

head = newNode; // head points to the newly made node.

break;

case 2:

newNode->next\_data = NULL; // previously 1st node is now pointed to by the newly made last node.

if (head == NULL){

head = newNode;

}

else{

ptr = head;

while(ptr->next\_data != NULL)

{

ptr = ptr->next\_data;

}

ptr->next\_data = newNode; // make the previously last node point to the newly made last node

}

break;

default:

system("cls");

printf("\nInsert at position : ");

scanf("%d", &position);

ptr = head;

while(i < position-1)

{

ptr = ptr->next\_data;

i++;

}

temp = ptr->next\_data; // holds the address of current data at pos.

ptr->next\_data = newNode; // makes the newly made data, data number pos.

newNode->next\_data = temp; // newly made data points to the prev pos data.

}

}

void delete(int pos)

{

int position, i = 1;

switch (pos)

{

case 1:

temp = head;

head = head->next\_data;

free(temp);

break;

case 2:

ptr = head;

if(head->next\_data == NULL){

free(ptr);

}

while(ptr->next\_data != NULL)

{

temp = ptr;

ptr = ptr->next\_data;

}

temp->next\_data = NULL;

free(ptr);

break;

default:

system("cls");

printf("\nDelete position : ");

scanf("%d", &position);

ptr = head;

while(i < position)

{

temp = ptr;

ptr = ptr->next\_data;

i++;

}

if(ptr == NULL){

printf("\nNOT FOUND\n");

getch();

break;

}

temp->next\_data = ptr->next\_data;

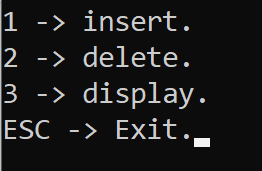
free(ptr);

}

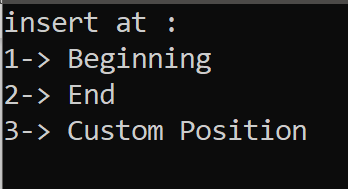
}

**Output:**

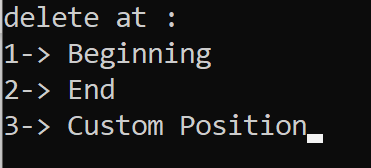
First Prompt Screen



Insert Options

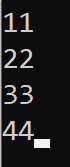


Delete Options



Inserting 11, 22, 33, 44 in order.

Display



Deleting 22 i.e., 2nd element, then inserting 999 as the 2nd element.

