DATA STRUCTURES AND ALOGRITHM

ASSIGNMENT NO 3

SARAVANAKARUPPU. K

Question 1:

Write a function "insert\_any()" for inserting a node at any given position of the linked list. Assume position starts at 0.

#include <studio.h>

typedef struct node

{

Int data -> (data part)

Struct node \* next

Node InsertNth(Node head, int data, int position)

{

Node

newNode = new Node();

newNode.data = data;

newNode.next = null;

if (head == null)

{

return newNode;

}

If (position == 0)

{

newNode.next = head;

head = newNode;

return head;

}

Node prev = null;

Node current = head;

Int I = 0;

While (current !=null && I < position)

{

Prev = current;

Current = current.next;

I++;

}

newNode.next = prev.next;

prev.next = newNode;

return head;

}

Question 2:

Write a function “delete\_beg()” for deleting a node from the beginning of the linked list.

Deletion Operation both done in same program using C++ language. So for easy access at all operation use of switch case is done in program. You can select one operation at a time. Total six operation is describe of insert & delete that is:-

1. Ins/Del at beg 2.Ins/Del at end 3.Ins/Del at specific location.

#include<iostream.h>  
#include<conio.h>  
#include<malloc.h>  
struct node  
{int info;  
struct node \*next;  
}\*start, \*temp;  
  
void insert\_beg(int);  
void insert\_end(int);  
void insert\_spe(int,int);  
void delete\_beg();  
void delete\_end();  
void delete\_spe(int);  
void display();  
  
void main()  
{clrscr();  
 start=NULL;  
 int item,choice,location,element,position;  
 cout<<endl<<"Insert & Delete operation in linked list by-Tarun Rawat\n";  
 again:  
 cout<<"\n1.Insert at beg of linked list\t";  
 cout<<"2.Insert at end of linked list\n";  
 cout<<"3.Insert at specific location\t";  
 cout<<"4.Delete from beginning of List \n5.delete from end of the list \t6.Delete from specific location of List\n7.Display linked list\t8.Exit\n";  
 cout<<"Enter choice : ";  
 cin>>choice;  
 switch(choice)  
 {case 1:cout<<"Enter item to insert : ";  
       cin>>item;  
       insert\_beg(item);  
       goto again;  
 case 2:cout<<"Enter item to insert : ";  
       cin>>item;  
       insert\_end(item);  
       goto again;  
 case 3:cout<<"Enter location to insert : ";  
       cin>>location;  
       cout<<"Enter item to insert : ";  
       cin>>item;  
       insert\_spe(item,location);  
       goto again;  
 case 4:delete\_beg();  
       goto again;  
 case 5:delete\_end();  
       goto again;  
 case 6:cout<<"Enter the location to delete : ";  
       cin>>location;  
       delete\_spe(location);  
       goto again;  
 case 7:cout<<"\nInserted item = ";  
       display();  
       goto again;  
 case 8 :cout<<"\nTHANK YOU";  
 default:break;  
 }  
 getch();  
}  
  
void insert\_beg(int item)  
{temp=(node\*)malloc(sizeof(node));  
temp->info=item;  
temp->next=start;  
start=temp;  
}  
  
void insert\_end(int item)  
{temp=(node\*)malloc(sizeof(node));  
temp->info=item;  
temp->next=NULL;  
node\* current=start;  
while(current->next!=NULL)  
    {current=current->next;  
    }  
current->next=temp;  
}  
  
void insert\_spe(int item,int location)  
{temp=(node\*)malloc(sizeof(node));  
temp->info=item;  
node\* current=start;  
int count=1;  
while (count <location-1)  
    {current=current->next;  
     count=count+1;  
    }  
temp->next=current->next;  
current->next=temp;  
}  
  
void delete\_beg()  
{temp=start;  
start=start->next;  
free(temp);  
}  
  
void delete\_end()  
{temp=start;  
node\* current;  
while(temp->next!=NULL)  
{  
current=temp;  
temp=temp->next;  
}  
current->next=NULL;  
free(temp);  
}  
  
void delete\_spe(int location)  
{node\* current;  
temp=start;  
int count=1;  
while(count<=location-1)  
{current=temp;  
temp=temp->next;  
count=count+1;  
}  
current->next=temp->next;  
free(temp);  
}  
  
void display()  
{temp=start;  
while(temp!=NULL)  
{cout<<temp->info<<" ";  
temp=temp->next;  
}  
cout<<"\n";  
  
}

Question 3:

Write a function “delete\_end()” for deleting a node from the end of the linked list.

Void delete\_pos()

{

Int I,pos;

Struct node \*temp,\*ptr;

If(start==NULL)

{

Printf(“nThe List is Empty:n”);

Exit(0);

}

Else

{

Printf(“nEnter the position of the node to be deleted:t”);

Scanf(“%d”,&pos);

If(pos==0)

{

Ptr=start;

Start=start->next ;

Printf(“nThe deleted element is:%dt”,ptr->info );

Free(ptr);

}

Else

{

Ptr=start;

For(i=0;i<pos;i++) { temp=ptr; ptr=ptr->next ;

If(ptr==NULL)

{

Printf(“nPosition not Found:n”);

Return;

}

}

Temp->next =ptr->next ;

Printf(“nThe deleted element is:%dt”,ptr->info );

Free(ptr);

}

}