```
Lab5:--
WAP to Implement Singly Linked List with following operations
a) a) Create a linked list. b) Insertion of a node at first position, at any position and at end of
list. c) Display the contents of the linked list.
Program:-
#include<stdio.h>
#include<stdlib.h>
struct node
 int info;
 struct node *link;
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
 printf("mem full\n");
```

exit(0);

return x;

void freenode(NODE x)

NODE insert_front(NODE first,int item)

}

}

{

{

NODE temp; temp=getnode(); temp->info=item; temp->link=NULL; if(first==NULL) return temp;

free(x);

```
temp->link=first;
first=temp;
return first;
}
NODE delete_front(NODE first)
NODE temp;
if(first==NULL)
printf("List is empty!Can't delete an item'\n");
return first;
}
temp=first;
temp=temp->link;
printf("The Item deleted at front-end is = %d\n",first->info);
free(first);
return temp;
NODE insert_rear(NODE first,int item)
NODE temp, cur;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
cur=first;
while(cur->link!=NULL)
cur=cur->link;
cur->link=temp;
return first;
NODE delete_rear(NODE first)
{
NODE cur, prev;
if(first==NULL)
printf("List is empty cannot delete\n");
return first;
}
```

```
if(first->link==NULL)
{
printf("Item deleted is %d\n",first->info);
free(first);
return NULL;
}
prev=NULL;
cur=first;
while(cur->link!=NULL)
prev=cur;
cur=cur->link;
printf("Item deleted at rear-end is %d",cur->info);
free(cur);
prev->link=NULL;
return first;
}
NODE insert_pos(int item,int pos,NODE first)
NODE temp,cur,prev;
int count;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL&&pos==1)
{
return temp;
if(first==NULL)
printf("invalid position\n");
return first;
}
if(pos==1)
temp->link=first;
first=temp;
return temp;
}
count=1;
prev=NULL;
cur=first;
```

```
while(cur!=NULL&&count!=pos)
{
prev=cur;
cur=cur->link;
count++;
if(count==pos)
prev->link=temp;
temp->link=cur;
return first;
printf("invalid position\n");
return first;
}
void display(NODE first)
NODE temp;
if(first==NULL)
printf("List empty cannot display items\n");
for(temp=first;temp!=NULL;temp=temp->link)
 printf("%d\t",temp->info);
 }
}
int main()
int item, choice, pos;
NODE first=NULL;
for(;;)
printf("\n 1:Insert at first position\t 2:Insert at any position\t 3.Insert at last position\t
4:Display_list\t 5:Exit\n");
printf("Enter the choice\n");
scanf("%d",&choice);
switch(choice)
 case 1:printf("Enter the item at front-end\n");
      scanf("%d",&item);
      first=insert_front(first,item);
      break;
```

```
case 2:printf("enter the item to be inserted at given position\n");
        scanf("%d",&item);
        printf("enter the position\n");
        scanf("%d",&pos);
        first=insert_pos(item,pos,first);
        break;
 case 3:printf("Enter the item at rear-end\n");
        scanf("%d",&item);
        first=insert_rear(first,item);
        break;
 case 4:display(first);
        break;
default:exit(0);
        break;
}
}
Output screenshot:-
                      2:Insert at any position
1:Insert at first position
Enter the choice
 1:Insert at first position 2:Insert at any position nter the choice
 nter the position
 1:Insert at first position 2:Insert at any position
                      2:Insert at any position
```

Written Screenshot:-

■ P Type here to search

print ("list empty annut display termino)

for (temp=tirust; temp) = NULL; temp=temp=1)

pt ("id \n", temp=info);

} temp = getnodil); temp > Into=item; temp > link = NULL; 4 (tirut == NULL && pos == 1) return temp; 4 (tirut==NOUL) int main() 5 int item, choice, pos, element; Pf ("invalid pos In"); NODE HOUSE - NULL; NODE second = NULL; action first; fon(i))
I pt ("In1 Invoit@ first per In2 inscrit@ last # (pes == x) 1/13 - invert @ specific pos una duplay us
pt ("enter choice"); temp -> link = first; return temp; ce (1-td 11, Scholu); switch (choice) 9+ (count = = pes) eased: pf ("entry the item @ front end in") prev-> link = temp) st ("rd", fitum); temp > link = cus; tierest = Insert - perona friend, retwin forst; case 2: point (" enter the "tern @ meanend". void display (NODE first) ef (" id", & ritem); first = insent_ near (first, item): Nobe temp: bruak; of (forst = = NOLL)

Lab-5 #include <stdio. h> # include < stdlib h> struct mode & int into: struct node * link; typedel structuode "NODE; Nope getnode() NODE X; x = (Note) mallor (size of (structural)); 18 (X=NULL) 1 points ("mein full \n"); sutton 2; void fremode (NODE x) free (x); NODE Provid from [NOOK first, and item) NODE temp; temp= getnode()

temp -> sink = NULL; (11UN == NULL) +i return temp: temp > link = forst; firest = tempi return forst; NODE IF (NODE second, int item) NODE temp; temp = get node(); temp > info = item; temp->link=NULL
if (second == NULL) return temps temp > link = second; second = temp; return cuand; NODE POWERL pos (int item, int pos, NODE, NODE temp; NODE previcion; int count;

tump= info = item;

DATE: / Pt (" enter the position to insert ");

St (" 1. d", & pos);

Pt (" enter the item to insert); case 3: st ("%d", sitem); finst = insent_pos(tem, pos, finst); briaki; cause 4: display (tirrst); bruak? default : exit (0); break: cres trong of room with xxxxx ") any hours