Labprogram-5

WAP to implement singly linked list with operations.

a) Create a linked list
b) Insertion at first position, any position and at end of list
c) Display content of linked list

```
#include<stdio.h>
#include<conio.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)
{
free(x);
}
```

```
NODE insert_front(NODE first, int item)
{
 NODE temp;
 temp=getnode();
 temp -> info=item;
 temp->link=NULL;
 if(first == NULL)
    {
     return temp;
    }
 temp -> link = first;
  first=temp;
  return first;
}
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->link=temp
       cur=cur->link;
    cur->link =temp;
  return first;
}
```

```
NODE insert_pos(int item, int pos, NODE first)
{
 NODE temp;
  NODE prev, cur;
 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;
     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("Wrong position\n");
return first;
}
void display(NODE first)
{
NODE temp;
if(first == NULL)
printf("cannot display items");
else
printf("Content of list:\n");
for(temp = first; temp != NULL; temp = temp->link)
{
printf("%d\n", temp->info);
}
}
void main()
{ int item, choice, pos;
NODE first = NULL;
for(;;)
{
printf("\n1.Insert front\n2.Insert rear
\n3.Insert pos\n4.Display list\n5.Exit\n");
printf("Enter choice");
switch(choice
scanf("%d", &choice);
switch(choice)
```

DATE / /

```
{
case 1:
    printf("Enter the item at front end\n");
     scanf("%od", &item);
       first = insert_front(first,item);
       break;
case 2:
      printf("Enter the item at rear end
     scanf("%od", &item);
     first = insert_rear(first, item);
     break;

case 3:..
     printf("Enter the position and item:\n");
      scanf("%d%d", &pos);
      scanf("%od", &item);
      first = insert_pos(item,pos, first);
      break;
case 4:
    display(first);
     break;
case 5:
    exit(0);
     break;
    default: printf("Invalid choice\n");
  }
 }
}
```