Aruna Ravi K R
1BM19EC010

06) WAP to implement singly linked list
a) create list
b) delete front, rear
c) display the contents of the linked list.

```
#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) {
free (x); }
NODE insert_front (NODE first, int item)
{
NODE delete_front (NODE first) {
NODE temp;
if (first == NULL) {
printf ("list is empty");
return first; }
temp = first;
temp = temp -> link;
```

```
printf("item deleted at end is - %d\n", first->info);
free(first);
return temp;}
NODE delete_beg(NODE first){
NODE cur, prev;
if(first==NULL)
{printf("list is empty");
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 beg end %d", cur->info);
free(cur);
prev->link=NULL;
return first;}
void display(NODE first){
NODE temp;
if(first==NULL)
printf("list is empty");
for(temp=first; temp!=NULL; temp=temp->link){
printf("%d\n", temp->info);
}
}
```

```
void main ()
{
int item, choice, pos;
NODE first = NULL;
for (;;) {
printf ("1: Delete_front \n 2: Delete_rear \n 3: Display \n
        4: Exit");
printf ("Enter the choice");
scanf ("%d", &choice);

Switch (choice) {
case 1: first = delete_front (first);
        break;
case 2: first = delete_rear (first);
        break;
case 3: display (first);
        break;
Case 4: exit (0);
        break;
}
}
}
```