Date __/__/____ Lab Program 5

Q WAP to implement Singly linked list

Sol
```
#include <stdio.h>
#include <stdlib.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 ("memory 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 delete_front (NODE first)
{
    NODE temp;
    if (first == NULL)
    {
        printf ("list is empty cannot delete \n");
        return first;
    }
    temp = first;
    temp = temp -> link;
    printf ("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;
```

Date ___/___/_____

```
    return first;
}
void display (NODE first)
{
  NODE temp;
  if (first == NULL)
  printf ("list is empty cannot display items \n");
  for (temp = first; temp != NULL; temp = temp->link
  {
    printf ("%d \n", temp -> info);
  }
}
void main ()
{
  int item, choice;
  NODE first = NULL;
  for(;;)
{
printf ("\n 1: Insert_front\n2: Delete_front\n
        3: Insert_rear\n 4: Delete_rear \n
                5: Display_ list \n 6: EXIT\n");

printf (" enter the choice \n");
scanf ("%d", &choice);
printf (" ------ \n");
switch (choice)
{
case 1: printf (" enter the item at front end \n");
   scanf ("%d", &item);
  first = insert_front (first, item);
   break;
```

Date __/__/____

```
case 2 :    first = delete_front (first);
            break;
case 3 : printf ("enter the item at rear end \n");
         scanf ("%d", &item);
         first = insert_rear (first, item);
         break;
case 4 : first = delete_rear (first);
         break;
case 5 : display (first);
         break;
default : exit (0);
break;
}
}
}
```