WAP to Implement Singly Linked List with the following operations:

a) Create a linked list
b) Insertion of a node at first position, at end of list
c) Display the contents of the linked list
d) Deletion of first element and last element in the 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 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
    ~~printf("List is empty cannot delete \n");~~
    ~~return first;~~ 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;
}
```

```
void display (NODE first)
{
  NODE temp;
  if (first == NULL)
  printf("List empty, cannot display items\n");
  printf("Contents of the 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("\n 1: Insert_front\n 2: 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);
  switch (choice)
  {
   case 1: printf("Enter the item at front-end\n");
   scanf("%d", &item);
   first = insert-front (first, item);
    break;
   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;
   case 6: exit(0);
   default: printf("Invalid choice!\n"); break;
                    }}}
```