WAP to Implement Singly Linked List with following operations.

a) Create a linked list

b) Insertion of a node at first position, at any position and at end of list

c) Display the content of the linked list.

```
#include <stdio.h>
#include <conio.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 ("~~LIST IS F~~ 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 = cur->link;
  cur->link = temp;
  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\n", temp->info);
}
```

```
void main()
{
    int item, choice, pos;
    NODE first = NULL;
    printf("SINGLY LINKED LIST");
    for(;;)
    {
        printf("\n1: Insert_front \n2: Insert_rear \n3: display_list
                \n4: Exit\n");
        printf("enter the choice\n");
        scanf("%d", &choice);
        switch(choice)
        {
        case 1: printf("enter the item to be inserted at front end:
                scanf("%d", &item);
                first = Insert_front(first, item);
                break;
        case 2: printf("enter the item to be inserted at rear end
                        \n");
                scanf("%d", &item);
                first = Insert_rear(first, item);
                break;
        case 3: display(first);
                break;
        default: exit(0);
                 break;
        }
    }
}
```