Week 8 :

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("Memory is 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 \n");
    return first;
  }
  temp = first;
  temp = temp -> link;
  printf(" Item deleted : %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 (" Empty list \n");
      return first;
  }
```

```
    if (first -> link == NULL)
    {
      printf(" Item deleted : %d \n", first -> info);
      free (first);
      return NULL;
    }
    prev = NULL;
    cur = first;
    while (cur -> link != NULL)
    {  prev = cur;
       cur = cur -> link;
    }
    printf (" Item deleted : %d \n", cur -> info);
    free (cur);
    prev -> link = NULL;
    return first;

}
void display (NODE first)
{
    NODE temp;
    if (first == NULL)
        printf(" Empty 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 (" 1: Insert front \n 2: Delete front \n 3: insert rear \n 4: delete rear
              \n 5: Display \n 6: Exit \n");
      printf (" Enter choice \n");
      scanf ("%d", &choice);
```

```
switch(choice)
{
    case 1 : printf("Enter item to insert at front end \n");
             scanf("%d", &item);
             first = insert_front(first, item);
             break;
    case 2 : first = delete_front(first);
             break;
    case 3 : printf("Enter item at rear - end \n");
             scanf("%d", &item);
             first = insert_rear(first, item);
             break;
    case 4 : first = delete_rear(first);
             break;
    case 5 : printf("List: \n");
             display(first);
             break;
    case 6 : exit(0); break;
    default : printf("Invalid choice! \n");
              break;
}
}
```