

- Q5) WAP to implement Singly linked list
 a) Create linked list
 b) Insert rear & front, display the contents of 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 finsert (NODE x)
{
    free(x);
}
NODE insert_front (NODE first, int item)
{
    NODE temp;
    temp = getnode();
    temp->info = item;
    temp->link = first;
    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 is empty");
    for (temp = first; temp != NULL; temp = temp->link)
        printf("%d\n", temp->info);
}

```

```

int main()
{
    int item, choice, pos;
    NODE first = NULL;
    printf("Enter the choice");
    scanf("%d", &choice);
    printf("1. Insert front 2. Insert rear 3. display\n 4. Exit\n");
    switch (choice)
    {
        case 1: printf("Enter the item to be added at front");
                scanf("%d", &item);
                first = insert_front(first, item);
                break;
    }
}

```

```

case 2: printf ("Enter the item to insert rear");
scanf ("%d", &item);
first = insert_rear (first, item);
break;

```

```

case 3: display (first);

```

```

case 4: break;

```

```

default: exit (0);
break;

```

```

}

```

```

}

```

```

return 0;

```

```

}

```