Name of the Experiment :
Experiment No. : OS_LAB_10
Date :
Page No. :

```
#include <stdio.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("mem full");
        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;
```

Signature.....

```
    temp->link = first;
    first = temp;
    return first; }
NODE delete_front(NODE first) {
    NODE temp;
    if (first == NULL) {
    printf("empty");
    return first; }
    temp = first;
    temp = temp->link;
    printf("item deleted: %d", 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");
        return first; }
    if(first->link == NULL{
        printf("deleted %d", first->info);
        free(first);
        return NULL; }
    prev = NULL;
    cur = first;
    while(cur->link != NULL){
        prev = cur;
        cur = cur->link; }
    printf("deleted %d", cur->info);
    free(cur);
    prev->link = NULL;
    return first;
}
NODE insert_pos(int item, int pos, NODE first){
    NODE temp, prev, cur;
    int count;
    temp = getnode();
    temp->info = item;
```

```
temp->link = NULL;
if (first == NULL && pos == 1)
    return temp;
if (first == NULL) {
    printf("invalid pos");
    return first; }
if (pos == 1) {
    temp->link = first;
    return temp; }
count = 1;
prev = NULL;
cur = first;
while (cur != NULL && count != pos) {
    prev = cur;
    cur = cur->link;
    count++; }
if (count == pos) {
    prev->link = temp;
    temp->link = cur;
    return first; }
printf("IP");
return first;
}
```


Signature.....

Name of the Experiment :
Experiment No. :
Date :
Page No. :

```
NODE delete_pos(int pos, NODE first){
    NODE prev, cur;
    int count;
    if(first==NULL || pos<0){
        printf("invalid");
        return NULL; }
    if(pos==1){
        cur = first;
        first = first->link;
        printf("deleted %d", cur->info);
        freenode(cur);
        return first; }
    prev = NULL;
    cur = first;
    count = 1;
    while(cur!=NULL){
        if(count==pos) break;
        prev = cur;
        cur = cur->link;
        count++; }
    if(count!=pos){
        printf("invalid");
        return first;
    }
```

Signature.....

```
    prev → link = cur → link;
    printf("deleted %d", cur → info);
    freenode(cur);
    return first;
}
void display(NODE first){
    NODE temp;
    if(first == NULL)
        printf("empty");
    for(temp = first; temp != NULL; temp = temp → link){
        printf("%d", temp → info);

void main(){
    int item, choice, pos;
        NODE first = NULL;
    for(;;){
printf("1: in_front 2: del_front 3: in_rear 4: del_rear
    5: in_pos 6: del_pos 7: display def: exit");
printf("enter choice");
    scanf("%d", &choice);
    switch(choice){
case 1: printf("enter item");
        scanf("%d", &item);
        first = insert_front(first, item);
        break;
```


Signature.....

```
case 2: first = delete_front (first); break;
case 3: printf ("enter item");
        scanf ("%d", &item);
        first = insert_rear (first, item);
        break;
case 4: first = delete_rear (first); break;
case 5: printf ("enter pos & item");
        scanf ("%d", &pos);
        scanf ("%d", &item);
        first = insert_pos (item, pos, first);
        break;
case 6: printf ("enter pos");
        scanf ("%d" &pos);
        first = delete_pos (pos, first);
        break;
case 7: display (first);
        break;
default: exit(0);
        }
    }
}
```


Signature.....