19-10-20 Lab 5

Date
Page 16

Q → (1) WAP for circular queue.

```
# include < stdio.h >
# include < stdlib.h >
# define qs 5
int item, f = 0, r = -1, q[qs]; c = 0;
void insert() {
    if (c == qs) {
        printf("Queue Overflow \n");
        return; }
    r = (r+1) % qs;
    q[r] = item;
    c++;
}
int delete() {
    if (c == 0) {
        return -1; }
    item = q[f];
    f = (f+1) % qs;
    c = c - 1;
    return item;
}
void display() {
    int i, front;
    if (c == 0) {
        printf("Queue is empty \n");
        return; }
```

```
    front = f;
    printf("Contents of queue : \n");
    for(i=0; i<c; i++){
        printf("%d \n", q[front]);
        front = (front+1)%qs; }
}
int main(){
    int choice;
    for(;;){
        printf("1: insert \n2: delete \n3: display");
        printf("Enter the choice : ");
        scanf("%d", &choice);
        switch(choice){
            case 1: printf("Enter item : ");
                    scanf("%d", &item);
                    insert();
                    break;
            case 2: item = delete;
                    if(item == -1)
                        printf("Queue is empty \n");
                    else
                        printf("Item deleted : %d \n",
                                item);
                    break;
            case 3: display();
                    break;
```

```
        default : exit(0);
    }
  }
  return 0;
}
```

O/P →

1: insert
2: delete
3: display
Enter choice = 1
Enter item : 23

1: insert
2: delete
3: display
Enter choice : 3
Contents :
23

1: insert
2: delete
3: display
Enter choice : 2
Item deleted : 23

(2) WAP for dequeue.

```
# include <stdio.h>
# include <stdlib.h>
# define qs 5
int f=0, r=-1, ch; item, q[10];
int isfull() {
        return (r == qs-1) ? 1:0; }
int isempty() {
        return (f>r) ? 1:0; }
void insert_rear() {
        if (isfull()) {
            printf("Queue overflow");
            return;
        }
        r = r+1;
        q[r] = item;
}
void delete_front() {
        if (isempty()) {
            printf("Queue underflow;");
            return; }
        printf("Item deleted : %d", q[f++]);
        if (f>r) { f=0;
                   r=-1; }
}
```

```
void insert_front() {
    if (f != 0) {
        f = f - 1;
        q[f] = item;
        return; }
    else if (f == 0 && r == -1) {
        q[++r] = item;
        return; }
    else {
        printf("Insertion Not Possible \n"); }
}
void delete_rear() {
    if (is_empty()) {
        printf("Queue underflow");
        return; }
    printf("Item deleted : %d", q[r--]);
    if (f > r) {
        f = 0;
        r = -1; }
}
void display() {
    int i;
    if (isempty()) {
        printf("Queue is empty");
        return; }
```

```
        printf("Contents of queue :");
        for(i=f; i<=r; i++){
            printf("%d", q[i]); }
}
int main(){
    for(;;){
    printf("1:insert_r 2:ins_f 3:del_r 4:del_f 5:display");
    printf("Enter choice :");
    scanf("%d", &ch);
    switch(ch){
        case 1: printf("Enter item");
                scanf("%d", &item);
                insert_rear();
                break;
        case 2: printf("Enter item");
                scanf("%d", &item);
                insert_front();
                break;
        case 3: delete_rear();
                break;
        case 4: delete_front();
                break;
        case 5: display;
                break;
```

```
        default : exit (0); }
    }
}
```

O/P →

```
1: i_f  2: i_r  3: d_f  4: d_r  5: display
Enter choice : 1
Enter item : 23

1: i_f  2: i_r  3: d_f  4: d_r  5: display
Enter choice : 2
Enter item : 46

1: i_f  2: i_r  3: d_f  4: d_r  5: display
Enter choice : 4
[redacted] Item deleted : 46

1: i_f  2: i_r  3: d_f  4: d_r  5: display
[redacted] Enter choice : 5
Contents :
23
```