

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; }
```

```
    c = (r + 1) % qs;
```

```
    q[c] = item;
```

```
    r++; }
```

3

```
int delete() {
```

```
    if (c == 0) {
```

```
        return -1; }
```

```
    item = q[f];
```

```
    f = (f + 1) % qs;
```

```
    c = c - 1;
```

```
    return item; }
```

3

```
void display() {
```

```
    int i, front;
```

```
    if (c == 0) {
```

```
        printf("Queue is empty\n");
```

```
    return; }
```

```

front = f;
printf("contents of queue : %s");
for (i=0; i < n; i++) {
    printf("%d ", q[front]);
    front = (front + 1) % q_size;
}

```

3

```

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);  
3

3

- return 0;

3

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 deque.

```
# include < stdio.h >
```

```
# include < stdlib.h >
```

```
# define qs 5
```

```
int f = 0, r = -1, ch; item, q[10];
```

```
int isfull () {
```

```
    return (r == q[0] - 1) ? 1 : 0; }
```

```
int isempty () {
```

```
    return (f > r) ? 1 : 0; }
```

```
void insert_rear () {
```

```
    if (isfull ()) {
```

```
        printf ("Queue overflow");
```

```
        return;
```

3

```
r = r + 1;
```

```
q[r] = item;
```

3

```
void delete_front () {
```

```
    if (isempty ()) {
```

```
        printf ("Queue underflow");
```

```
        return; }
```

```
    printf ("Item deleted : %d", q[f++]);
```

```
    if (f > r) { f = 0;
```

```
    r = -1; }
```

3

```
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 (isempty()) {  
        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++) {
    prints ("%.d", q[i]);
}

int main () {
    for (;;) {
        printf ("1:insert or 2:ins-f 3: del or 4:del
                5: display ");
        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); } }

3

3

0/p>

1: i-n 2: i-n 3: d-f 4: d-r s: display  
play

Enter choice : 1

Enter item : 23

1: i-n 2: i-n 3: d-f 4: d-r s: display

Enter choice : 2

Enter item : 46

1: i-n 2: i-n 3: d-f 4: d-r s: display

Enter choice : 4

~~46~~ Item deleted : 46

1: i-n 2: i-n 3: d-f 4: d-r s: display

~~s: display~~ Enter choice : 5

Contents :

23

③ WAP to simulate the working of a queue of integers using an array.

```
#include <stdio.h>
#include <stdlib.h>
#define Q_SIZE 3
int item, f = 0, r = -1, q[10];
void insertrear() {
    if (r == Q_SIZE - 1) {
        printf("Queue overflow\n");
        return;
    }
    r = r + 1;
    q[r] = item;
}
int deletefront() {
    if (f > r) {
        f = 0;
        r = -1;
        return -1;
    }
    return q[f++];
}
void display() {
    int i;
    if (f > r) {
        printf("Queue is empty");
        return;
    }
    printf("Contents of queue : ");
}
```

```
for (i = 1; i <= n; i++) {  
    printf("%d\n", q[i]);}
```

3

```
int main() {  
    int choice;  
    for (;;) {  
        printf("1: insert 2: delete 3: display");  
        printf("Enter choice : ");  
        switch (choice) {  
            case 1: printf("Enter item : ");  
                scanf("%d", &item);  
                insertrear();  
                break;  
            case 2: item = deletefront();  
                if (item == -1) {  
                    printf("Queue Underflow");  
                    break; }  
                else {  
                    printf("Deleted : %d\n", item);  
                    break; }  
            case 3: display();  
                break;  
            default: exit(0); }  
    }  
}
```

3

0/1 →

1: insert 2: delete 3: display

Enter choice : 1

Enter item : 50

1: insert 2: delete 3: display

Enter choice : 3

Contents :

50

1: insert 2: delete 3: display

~~contents~~ Enter choice : 2

Item deleted : 50