

EXTRA PROGRAMS:

DEQUEUE:

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

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

```
#define qsize 3
```

```
int f=0,r=-1,ch;
```

```
int item,q[10];
```

```
int isfull()
```

```
{  
    return(r==qsize-1)?1:0;  
}
```

```
int isempty()
```

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

```
void insert_rear()
```

```
{  
    if(isfull())  
    {  
        printf("queue overflow\n");  
        return;  
    }  
    r=r+1;  
    q[r]=item;  
}
```

```

void delete_front()
{
    if(isempty())
    {
        printf("queue empty\n");
        return;
    }
    printf("item deleted is %d\n",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(isempty())

        {

            printf("queue is empty\n");

            return;

        }

    printf("item deleted is %d\n",q[(r)--]);

    if(f>r)

        {

            f=0;

            r=-1;

        }

}

void display()

{

    int i;

    if(isempty())

        {

            printf("queue empty\n");

            return;

        }

    for(i=f;i<=r;i++)

        printf("%d\n",q[i]);

```

```

}

int main()

{

    for(;;)

    {

        printf("1.insert_rear\n2.insert_front\n3.delete_rear\n4.delete_front\n5.display\n6.exit\n");

        printf("enter choice\n");

        scanf("%d",&ch);

        switch(ch)

        {

            case 1:printf("enter the item\n");

                        scanf("%d",&item);

                        insert_rear();

                        break;

            case 2:printf("enter the item\n");

                        scanf("%d",&item);

                        insert_front();

                        break;

            case 3:delete_rear();

                        break;

            case 4:delete_front();

                        break;

            case 5:display();

                        break;

            default:exit(0);
        }
    }
}

```

```

    }

}

}

```

screenshots:

```

main.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  #define qsize 3
4  int f=0,r=-1,ch;
5  int item,q[10];
6
7  int isfull()
8  {
9      return(r==qsize-1)?1:0;
10 }
11 int isempty()
12 {
13     return(f>r)?1:0;
14 }
15 void insert_rear()
16 {
17     if(isfull())
18     {
19         printf("queue overflow\n");
20         return;
21     }
22     r=r+1;
23     q[r]=item;
24 }
25 void delete_rear()
26 {
27     if(isempty())
28     {
29         printf("queue empty\n");
30         return;
31     }
32     printf("item deleted is %d\n",q[(f)++]);
33     if(f>r)
34     {
35         f=0;
36         r=-1;
37     }
38 }
39 void insert_front()
40 {
41     if(f==0)
42     {
43         f=f+1;
44         q[f]=item;
45 }
46 }

```

```

> clang-7 -pthread -lm -o main main.c
> ./main
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice

```

```

main.c
23     q[r]=item;
24 }
25 void delete_rear()
26 {
27     if(isempty())
28     {
29         printf("queue empty\n");
30         return;
31     }
32     printf("item deleted is %d\n",q[(f)++]);
33     if(f>r)
34     {
35         f=0;
36         r=-1;
37     }
38 }
39 void insert_front()
40 {
41     if(f==0)
42     {
43         f=f+1;
44         q[f]=item;
45 }
46 }

```

```

1
enter the item
10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
1
enter the item
30
1.insert_rear
2.insert_front

```

Repl.it - SoupySlushyPattern x Repl.it - cnhe x (2) WhatsApp x +

repl.it/@PremaPrema/cnhe#main.c

Apps Gmail YouTube Maps News Translate SEM HALL TICKET

PremaPrema / cnhe Stop Upgrade Share +

Files main.c

```
43 f=f-1;
44 q[f]=item;
45 return;
46 }
47 else if((f==0)&&(r==1))
48 {
49 q[++r]=item;
50 return;
51 }
52 else
53 printf("insertion not possible\n");
54 }
55 void delete_rear()
56 {
57 if(isempty())
58 {
59 printf("queue is empty\n");
60 return;
61 }
62 printf("item deleted is %d\n",q[(r)--]);
63 if(f>r)
64 {
65 f=f-1;
66 }
67 }
68 }
69 void display()
70 {
71 int i;
72 if(isempty())
73 {
74 printf("queue empty\n");
75 return;
76 }
77 for(i=f;i<=r;i++)
78 printf("%d\n",q[i]);
79 }
80 int main()
81 {
82 for(;;)
83 {
84 printf("1.insert_rear\n");
85 printf("2.insert_front\n");
86 printf("3.delete_rear\n");
87 printf("4.delete_front\n");
88 printf("5.display\n");
89 printf("6.exit\n");
90 printf("enter choice\n");
91 int choice;
92 scanf("%d",&choice);
93 switch(choice)
94 {
95 case 1:
96 insert_rear();
97 break;
98 case 2:
99 insert_front();
100 break;
101 case 3:
102 delete_rear();
103 break;
104 case 4:
105 delete_front();
106 break;
107 case 5:
108 display();
109 break;
110 case 6:
111 exit(0);
112 }
113 }
```

```
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
3
item deleted is 30
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
20
```

Type here to search

22:14 08-11-2020

Repl.it - SoupySlushyPattern x Repl.it - cnhe x (2) WhatsApp x +

repl.it/@PremaPrema/cnhe#main.c

Apps Gmail YouTube Maps News Translate SEM HALL TICKET

PremaPrema / cnhe Stop Upgrade Share +

Files main.c

```
64 {
65 f=0;
66 r=-1;
67 }
68 }
69 void display()
70 {
71 int i;
72 if(isempty())
73 {
74 printf("queue empty\n");
75 return;
76 }
77 for(i=f;i<=r;i++)
78 printf("%d\n",q[i]);
79 }
80 int main()
81 {
82 for(;;)
83 {
84 printf("1.insert_rear\n");
85 printf("2.insert_front\n");
86 printf("3.delete_rear\n");
87 printf("4.delete_front\n");
88 printf("5.display\n");
89 printf("6.exit\n");
90 printf("enter choice\n");
91 int choice;
92 scanf("%d",&choice);
93 switch(choice)
94 {
95 case 1:
96 insert_rear();
97 break;
98 case 2:
99 insert_front();
100 break;
101 case 3:
102 delete_rear();
103 break;
104 case 4:
105 delete_front();
106 break;
107 case 5:
108 display();
109 break;
110 case 6:
111 exit(0);
112 }
113 }
```

```
2.insert front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
```

Type here to search

22:14 08-11-2020

The screenshot shows a Replit IDE interface with the following components:

- Browser Tabs:** Replit - SoupySlushyPattern, Replit - cnhe, (2) WhatsApp.
- Address Bar:** repl.it/@PremaPrema/cnhe#main.c
- Navigation Bar:** Includes a menu icon, a profile icon for 'PremaPrema / cnhe', a 'Stop' button, and 'Upgrade' and 'Share' buttons.
- Files Panel:** Shows a file named 'main.c'.
- Code Editor:** Displays the following C code in 'main.c':

```
84 for(;;)
85 {
86     printf
87     ("1.insert_rear\n2.insert_front\n3.delete_rear\n4
88     .delete_front\n5.display\n6.exit\n");
89     printf("enter choice\n");
90     scanf("%d",&ch);
91     switch(ch)
92     {
93     case 1:printf("enter the item\n");
94             scanf("%d",&item);
95             insert_rear();
96             break;
97     case 2:printf("enter the item\n");
98             scanf("%d",&item);
99             insert_front();
100            break;
101     case 3:delete_rear();
102            break;
103     case 4:delete_front();
104            break;
105     case 5:display();
106            break;
```
- Terminal:** Shows the program's execution output:

```
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
4
item deleted is 10
1.insert_rear
2.insert_front
3.delete_rear
4.delete_front
5.display
6.exit
enter choice
5
20
1.insert_rear
2.insert_front
3.delete_rear
4.delete_rear
5.display
6.exit
enter choice
[]
```
- Taskbar:** Includes a search bar and various application icons. The system clock shows 22:14 on 08-11-2020.

written:

Queues.

→ Dequeue. (input restricted and out restricted)

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

```
#define qsize 3
```

```
int f=0, r=-1, ch;
```

```
int item, q[10];
```

```
int isfull()
```

```
{
```

```
return (r == qsize - 1) ? 1 : 0;
```

```
}
```

```
int isempty()
```

```
{
```

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

```
}
```

```
void insert_rear()
```

```
{
```

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

```
{ printf("queue overflow\n");
```

```
return;
```

```
}
```

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

```
}
```



```
void delete-front()
{
    if (isEmpty())
    {
        pf("underflow")
    }
    else
    {
        item pf("item deleted is %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
    {
        pf("insertion not possible in");
    }
}

// comment insert front to input restriction
*/
```

```
/* void delete-rear()
{
    if (isEmpty())
    {
        printf("queue is empty\n");
        return;
    }
    printf("item deleted is %d\n", q[r--]);
    if (f > r)
    {
        f = 0;
        r = -1;
    }
}
*/ // comment delete rear to restrict output

void display()
{
    int i;
    if (isEmpty())
    {
        printf("queue empty\n");
        return;
    }
    for (i = f; i <= r; i++)
        printf("%d\n", q[i]);
}

void main()
{
    for(;;)
```

```
{
    pf("1. insert rear\n2. insert front\n3. delete rear\n4. delete front\n5. display\n6. exit\n");
    pf("enter choice\n");
    st("%d", &ch);
    switch(ch)
    {
        case 1: pf("enter the item\n");
                st("%d", &item);
                insert_rear();
                break;
        case 2: pf("enter the item\n");
                st("%d", &item);
                insert_front();
                break;
        case 3: delete_rear();
                break;
        case 4: delete_front();
                break;
        case 5: display();
                break;
        default: exit(0);
    }
}
```

