

DEFQ

#include <stdio.h>

#include <stdlib.h>

#define qsize 5

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

int item[10];

int isfull()

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

}

int isempty()

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

}

void insert_rear()

{ if (isfull())

{ printf("== Queue Overflow ==\n"); }

return;

4

5

r++; q[r] = item;

6

void delete-front()

{

if (isempty())

{ printf("Queue empty == %n",

return;

}

printf(" item deleted is %.d\n", q[f+1]);

if (f > r)

{ f = 0; r = -1; }

}

void insert-front()

{ if (f1 == 0)

{ ~~f~~ f = f - 1; q[f] = item;

return; }

else if ((f == 0) && (r == -1))

{ q[++r] = item; return; }

else

printf("== insertion not possible ==%

}

void delete-rear()

{ if (isempty())

{ printf(" == Queue is empty ==%n",

return;

}

printf(" item deleted is %.d\n", q[r-1]);

#include <queue.h>

{ f = 0; r = -1; }

y

void display()

{ int i;

if (is empty())

{ printf("Queue empty\n");
return;

y

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

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

y

void main()

{ for(;;)

{ printf("1. insert_rear\n 2. insert_front\n

3. delete_rear\n 4. delete_front\n 5. display\n\n 6. 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);
```

y
y
y.

Input restricted Q:-

```
#include <stdio.h>  
#include <stdlib.h>  
#define qsize 5  
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 delete_rear()
{
    if (isempty())
        printf("== Queue is empty ==\n");
    return;
}

```

```

printf("item deleted is %.d\n", q[r-1]);
if (f > r)
    f = 0; r = -1;
}

```

```
void display()
{
    int i;
    if (is empty())
        printf(" == Queue empty == \n");
    return;
}

printf(" = = = The elements of Queue = = = \n");
for (i = f; i <= r; i++)
    printf("%d \n", Q[i]);
```

```
void main()
{
    for (;;)
    {
        printf(" = = = = = \n");
        printf(" 1. insert-rear \n 2. delete-rear \n");
        printf(" 3. delete-front \n 4. display \n 5. 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: delete_rear();
                break;
        }
    }
}
```

case 3: delete - front();
break;

case 4: display();
break;

default: exit(6);

}

y

3

Output noticed. Q.

#include <stdio.h>

#include <stdlib.h>

#define qsize 5

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

int item, q[10];

int isfull()

{

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

y

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 Underflow = = = \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;
```

3

else

```
printf (" == insertion not possible
        == = \n");
```

3

void display()

{ int i;

if (isempty())

else printf (" == Queue empty == \n")

return;

3

printf (" == Content of Queue == \n");

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

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

3

void main()

q for();

else printf (" == \n");

printf (" 1.insert_rear In 2.insert_front
in 3.delete_front In 4.display In 5.exit()");

printf ("Enter choice \n");

```
scanf ("%1.d", &ch);  
switch (ch)
```

```
{ case 1: printf ("Enter the item\n");  
    scanf ("%1.d", &item);  
    insert_rear();  
    break;
```

```
case 2: printf ("Enter the item\n");  
    scanf ("%1.d", &item);  
    insert_front();  
    break;
```

```
case 3: delete_front();  
    break;
```

```
case 4: display();
```

```
default: init(0);
```

4

4

4.