#include <stdio.h>

#include<stdlib.h>

int queue\_array[20];

int rear = - 1;

int front = - 1;

void insert()

{

int add\_item;

if (rear == 20-1)

printf("Queue Overflow \n");

else

{

if (front == - 1)

/\*If queue is initially empty \*/

front = 0;

printf("\nEnter element : ");

scanf("%d", &add\_item);

printf("\n%d is inserted in queue\n",add\_item);

rear = rear + 1;

queue\_array[rear] = add\_item;

}

}

void delete()

{

if (front == - 1 || front > rear)

{

printf("Queue Underflow \n");

return ;

}

else

{

printf("\nElement deleted from queue is : %d\n", queue\_array[front]);

front = front + 1;

}

}

void display()

{

int i;

if (front == - 1)

printf("Queue is empty \n");

else

{

printf("\nQueue is : ");

for (i = front; i <= rear; i++)

printf("%d ", queue\_array[i]);

printf("\n");

}

}

int main()

{

printf("Perform operations on queue\n");

printf("\tMenu");

printf("\n------------------------------\n");

printf("1. Insert element \n");

printf("2. Delete element \n");

printf("3. Display queue\n");

printf("4. Exit\n");

printf("------------------------------\n");

int ch;

while (1)

{

printf("Choose operation : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

default:

printf("Invalid operation \n");

}

}

return 0;

}

#include <stdio.h>

#include<stdlib.h>

int queue\_array[20];

int rear = - 1;

int front = - 1;

void insert()

{

int add\_item;

if (rear == 20-1)

printf("Queue Overflow \n");

else

{

if (front == - 1)

/\*If queue is initially empty \*/

front = 0;

printf("\nEnter element : ");

scanf("%d", &add\_item);

printf("\n%d is inserted in queue\n",add\_item);

rear = rear + 1;

queue\_array[rear] = add\_item;

}

}

void delete()

{

if (front == - 1 || front > rear)

{

printf("Queue Underflow \n");

return ;

}

else

{

printf("\nElement deleted from queue is : %d\n", queue\_array[front]);

front = front + 1;

}

}

void display()

{

int i;

if (front == - 1)

printf("Queue is empty \n");

else

{

printf("\nQueue is : ");

for (i = front; i <= rear; i++)

printf("%d ", queue\_array[i]);

printf("\n");

}

}

int main()

{

printf("Perform operations on queue\n");

printf("\tMenu");

printf("\n------------------------------\n");

printf("1. Insert element \n");

printf("2. Delete element \n");

printf("3. Display queue\n");

printf("4. Exit\n");

printf("------------------------------\n");

int ch;

while (1)

{

printf("Choose operation : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

default:

printf("Invalid operation \n");

}

}

return 0;

}

#include <stdio.h>

#include<stdlib.h>

int queue\_array[20];

int rear = - 1;

int front = - 1;

void insert()

{

int add\_item;

if (rear == 20-1)

printf("Queue Overflow \n");

else

{

if (front == - 1)

/\*If queue is initially empty \*/

front = 0;

printf("\nEnter element : ");

scanf("%d", &add\_item);

printf("\n%d is inserted in queue\n",add\_item);

rear = rear + 1;

queue\_array[rear] = add\_item;

}

}

void delete()

{

if (front == - 1 || front > rear)

{

printf("Queue Underflow \n");

return ;

}

else

{

printf("\nElement deleted from queue is : %d\n", queue\_array[front]);

front = front + 1;

}

}

void display()

{

int i;

if (front == - 1)

printf("Queue is empty \n");

else

{

printf("\nQueue is : ");

for (i = front; i <= rear; i++)

printf("%d ", queue\_array[i]);

printf("\n");

}

}

int main()

{

printf("Perform operations on queue\n");

printf("\tMenu");

printf("\n------------------------------\n");

printf("1. Insert element \n");

printf("2. Delete element \n");

printf("3. Display queue\n");

printf("4. Exit\n");

printf("------------------------------\n");

int ch;

while (1)

{

printf("Choose operation : ");

scanf("%d", &ch);

switch(ch)

{

case 1:

insert();

break;

case 2:

delete();

break;

case 3:

display();

break;

case 4:

exit(1);

default:

printf("Invalid operation \n");

}

}

return 0;

}

Perform operations on queue

        Menu

------------------------------

1. Insert element

2. Delete element

3. Display queue

4. Exit

------------------------------

Choose operation : 1

Enter element : 2

2 is inserted in queue

Choose operation : 2

Element deleted from queue is : 2

Choose operation :

#include <stdio.h>

int queue[6];

int front=-1;

int rear=-1;

void enqueue(int element)

{

if(front==-1 && rear==-1) // check queue is empty

{

front=0;

rear=0;

queue[rear]=element;

}

else if((rear+1)%6==front) // condition to check queue is full

{

printf("Queue is overflow..");

}

else

{

rear=(rear+1)%6; // rear is incremented

queue[rear]=element; // assigning a value to the queue at the rear position.

}

}

int dequeue()

{

if((front==-1) && (rear==-1)) // check queue is empty

{

printf("\nQueue is underflow..");

}

else if(front==rear)

{

printf("\nThe dequeued element is %d", queue[front]);

front=-1;

rear=-1;

}

else

{

printf("\nThe dequeued element is %d", queue[front]);

front=(front+1)%6;

}

}

// function to display the elements of a queue

void display()

{

int i=front;

if(front==-1 && rear==-1)

{

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

}

else

{

printf("\nElements in a Queue are :");

while(i<=rear)

{

printf("%d,", queue[i]);

i=(i+1)%6;

}

}

}

int main()

{

int choice=1,x; // variables declaration

while(choice<4 && choice!=0) // while loop

{

printf("\n Press 1: Insert an element");

printf("\nPress 2: Delete an element");

printf("\nPress 3: Display the element");

printf("\nEnter your choice");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the element which is to be inserted");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

}}

return 0;

}

#include <stdio.h>

int queue[6];

int front=-1;

int rear=-1;

void enqueue(int element)

{

if(front==-1 && rear==-1) // check queue is empty

{

front=0;

rear=0;

queue[rear]=element;

}

else if((rear+1)%6==front) // condition to check queue is full

{

printf("Queue is overflow..");

}

else

{

rear=(rear+1)%6; // rear is incremented

queue[rear]=element; // assigning a value to the queue at the rear position.

}

}

int dequeue()

{

if((front==-1) && (rear==-1)) // check queue is empty

{

printf("\nQueue is underflow..");

}

else if(front==rear)

{

printf("\nThe dequeued element is %d", queue[front]);

front=-1;

rear=-1;

}

else

{

printf("\nThe dequeued element is %d", queue[front]);

front=(front+1)%6;

}

}

// function to display the elements of a queue

void display()

{

int i=front;

if(front==-1 && rear==-1)

{

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

}

else

{

printf("\nElements in a Queue are :");

while(i<=rear)

{

printf("%d,", queue[i]);

i=(i+1)%6;

}

}

}

int main()

{

int choice=1,x; // variables declaration

while(choice<4 && choice!=0) // while loop

{

printf("\n Press 1: Insert an element");

printf("\nPress 2: Delete an element");

printf("\nPress 3: Display the element");

printf("\nEnter your choice");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the element which is to be inserted");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

}}

return 0;

}#include <stdio.h>

int queue[6];

int front=-1;

int rear=-1;

void enqueue(int element)

{

if(front==-1 && rear==-1) // check queue is empty

{

front=0;

rear=0;

queue[rear]=element;

}

else if((rear+1)%6==front) // condition to check queue is full

{

printf("Queue is overflow..");

}

else

{

rear=(rear+1)%6; // rear is incremented

queue[rear]=element; // assigning a value to the queue at the rear position.

}

}

int dequeue()

{

if((front==-1) && (rear==-1)) // check queue is empty

{

printf("\nQueue is underflow..");

}

else if(front==rear)

{

printf("\nThe dequeued element is %d", queue[front]);

front=-1;

rear=-1;

}

else

{

printf("\nThe dequeued element is %d", queue[front]);

front=(front+1)%6;

}

}

// function to display the elements of a queue

void display()

{

int i=front;

if(front==-1 && rear==-1)

{

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

}

else

{

printf("\nElements in a Queue are :");

while(i<=rear)

{

printf("%d,", queue[i]);

i=(i+1)%6;

}

}

}

int main()

{

int choice=1,x; // variables declaration

while(choice<4 && choice!=0) // while loop

{

printf("\n Press 1: Insert an element");

printf("\nPress 2: Delete an element");

printf("\nPress 3: Display the element");

printf("\nEnter your choice");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Enter the element which is to be inserted");

scanf("%d", &x);

enqueue(x);

break;

case 2:

dequeue();

break;

case 3:

display();

}}

return 0;

}

Press 1: Insert an element

Press 2: Delete an element

Press 3: Display the element

Enter your choice1

Enter the element which is to be inserted4

 Press 1: Insert an element

Press 2: Delete an element

Press 3: Display the element

Enter your choice2

The dequeued element is 4

 Press 1: Insert an element

Press 2: Delete an element

Press 3: Display the element

Enter your choice^C