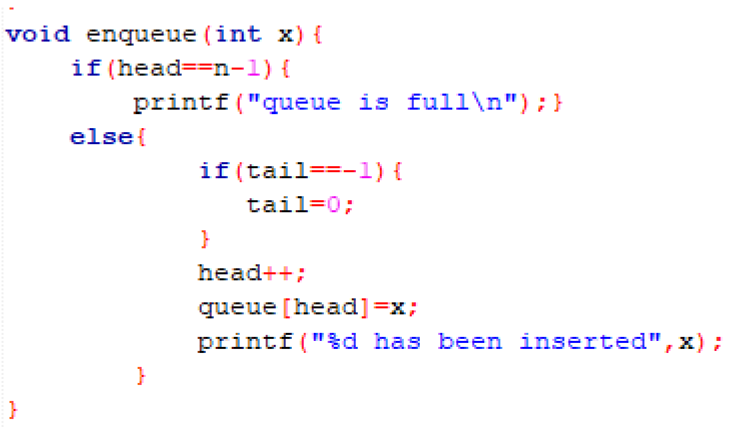
QUEUES

Queues are data structure in which its operations can be carried out on both ends of the queue. Two variables **head** and **tail** are declared globally at the beginning of the program.

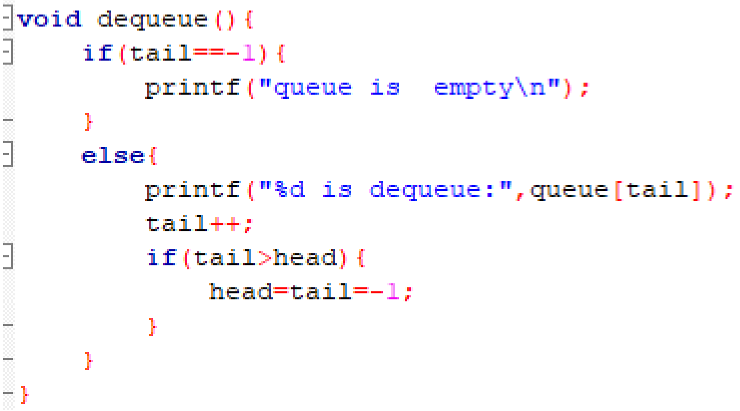
QUEUE OPERATIONS

1. **Enqueue**: This functions insert elements in the queue from the top or head of the queue. Elements can't be inserted on an empty. If queue is empty a message queue is full else increment head and set queue of head to the number that want to be inserted.



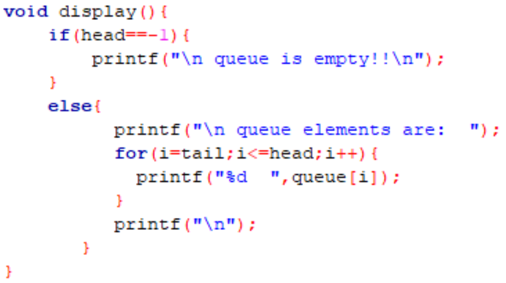
```
void enqueue(int x){
     if(head==n-1){
         printf("queue is full\n");}
     else{
               if(tail==-1){
                   tail=0;
               }
               head++;
               queue[head]=x;
               printf("%d has been inserted",x);
          }
}
```

2**. Dequeue:** This function delete elements from the tail of the queue. Element can't be deleted from an empty queue if queue is empty a message" **queue is empty**" is displayed else the first element in the queue is deleted and displayed.



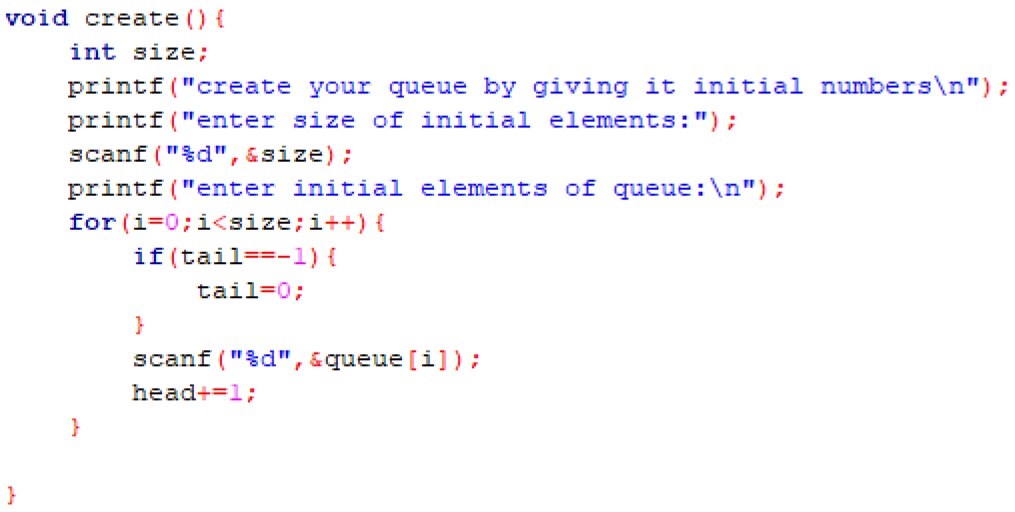
```
void dequeue(){
     if(tail==-1){
         printf("queue is  empty\n");
     }
     else{
         printf("%d is dequeue:",queue[tail]);
         tail++;
         if(tail>head){
             head=tail=-1;
         }
     }
}
```

3. **Display**: This function display all the element on the queue that is from head to tail. But it checks if the queue is empty first is queue is empty a message "**queue is empty**" is displayed because we can't display if there is nothing in the queue.



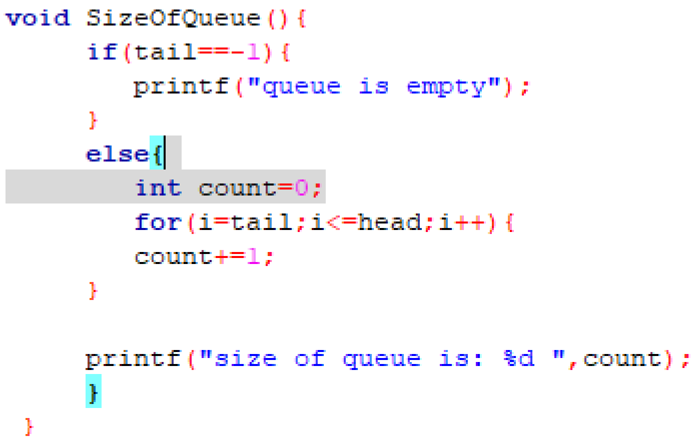
```
void display(){
    if(head==-1){
        printf("\n queue is empty!!\n");
    }
    else{
          printf("\n queue elements are:  ");
          for(i=tail;i<=head;i++){
            printf("%d  ",queue[i]);
          }
          printf("\n");
       }
}
```

4. **Create**: This function create the queue and put some initial element in the queue if desired.it ask user for the size of the number of element the queue should initially have .elements can still be enqueued in rather than initializing them.



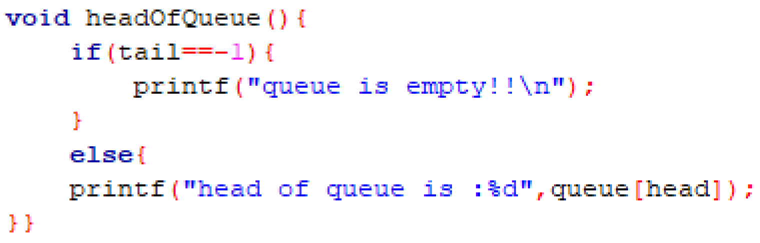
```
void create(){
    int size;
    printf("create your queue by giving it initial numbers\n");
    printf("enter size of initial elements:");
    scanf("%d",&size);
    printf("enter initial elements of queue:\n");
    for(i=0;i<size;i++){
        if(tail==-1){
            tail=0;
        }
        scanf("%d",&queue[i]);
        head+=1;
    }

}
```

5.**SizeOfQueue**: This function calculate the number of elements in the queue. If queue is empty it displays a message queue is empty and the number of element in the queue is displayed.



```
void SizeOfQueue(){
     if(tail==-1){
        printf("queue is empty");
     }
     else{
        int count=0;
        for(i=tail;i<=head;i++){
        count+=1;
     }

     printf("size of queue is: %d ",count);
     }
 }
```

6. **headOfQueue**: This function prints the top element of the queue that is the last element that was inserted in the queue.



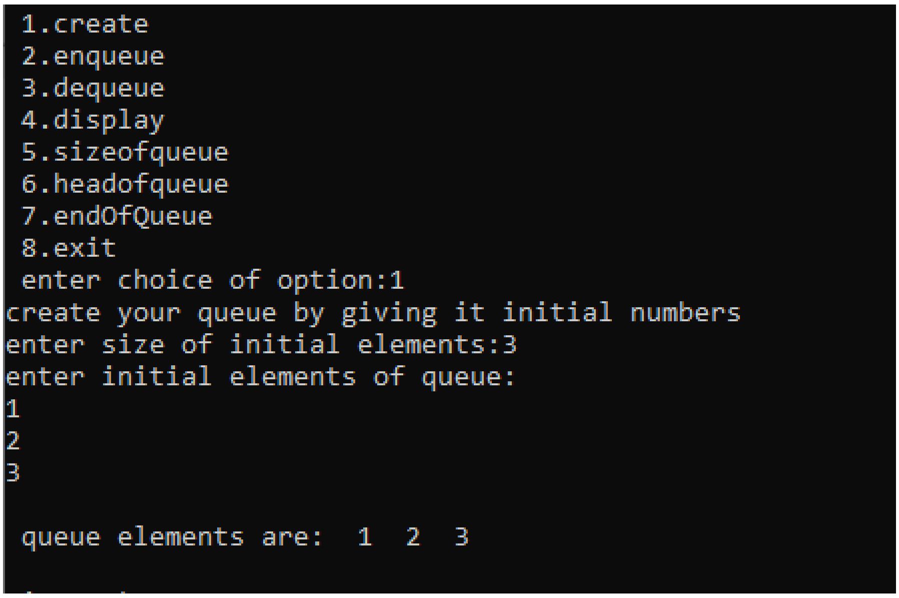
```
void headOfQueue(){
    if(tail==-1){
        printf("queue is empty!!\n");
    }
    else{
    printf("head of queue is :%d",queue[head]);
}}
```

7.**endOfQueue:**This function prints the element at the tail of queue.

```
void endOfQueue(){
     if(head==n-1){
        printf("queue is full\n");}
    else{

            printf("end of queue is:%d ",queue[tail]);}
}
```

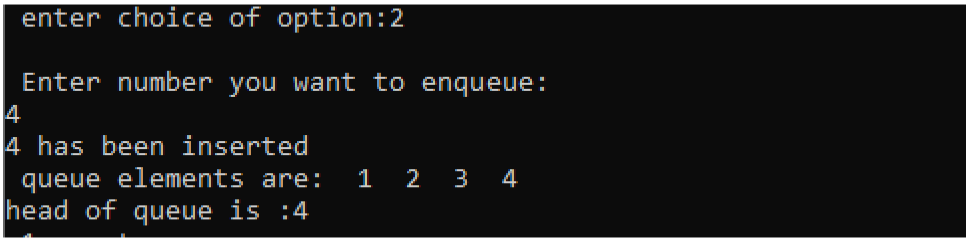
Main function

The main function is program using switch function .operations are performed based on the choice entered by user. If the user enters a choice of 7 the program ends and if user enters a choice that's not in the list of choice a message "**invalid choice is displayed**".



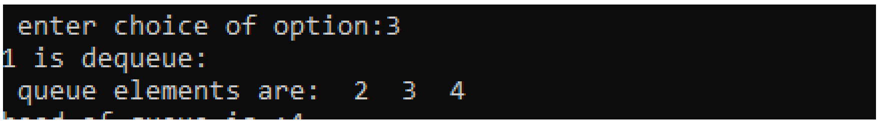
```
 1.create
 2.enqueue
 3.dequeue
 4.display
 5.sizeofqueue
 6.headofqueue
 7.endOfQueue
 8.exit
 enter choice of option:1
create your queue by giving it initial numbers
enter size of initial elements:3
enter initial elements of queue:
1
2
3

 queue elements are:  1  2  3
```

The above screen shot above create a queue of 3 elements and display it.



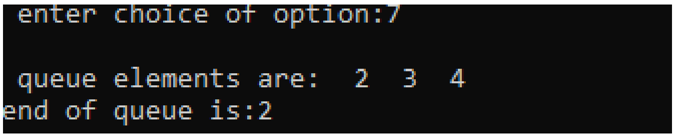
```
 enter choice of option:2

 Enter number you want to enqueue:
4
4 has been inserted
 queue elements are:  1  2  3  4
head of queue is :4
```

The above program insert 4 to the created queue, display and print the top element of the queue.



```
 enter choice of option:3
1 is dequeue:
 queue elements are:  2  3  4
head of queue is :4
```

The above screen shot dequeue 1 from the queue and print the queue.



```
 enter choice of option:7

 queue elements are:  2  3  4
end of queue is:2
```

The above screen shot displays queue and print head.