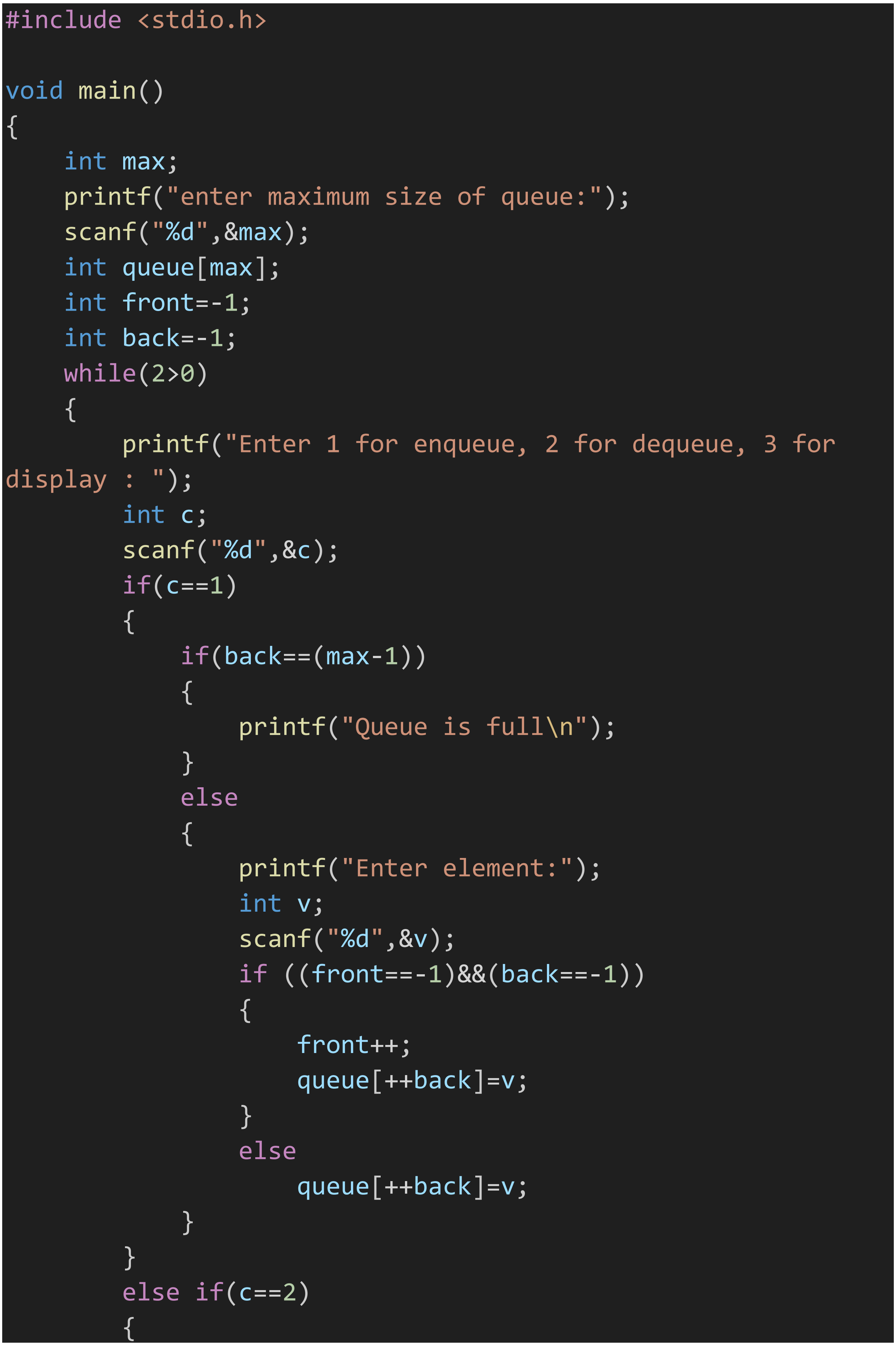
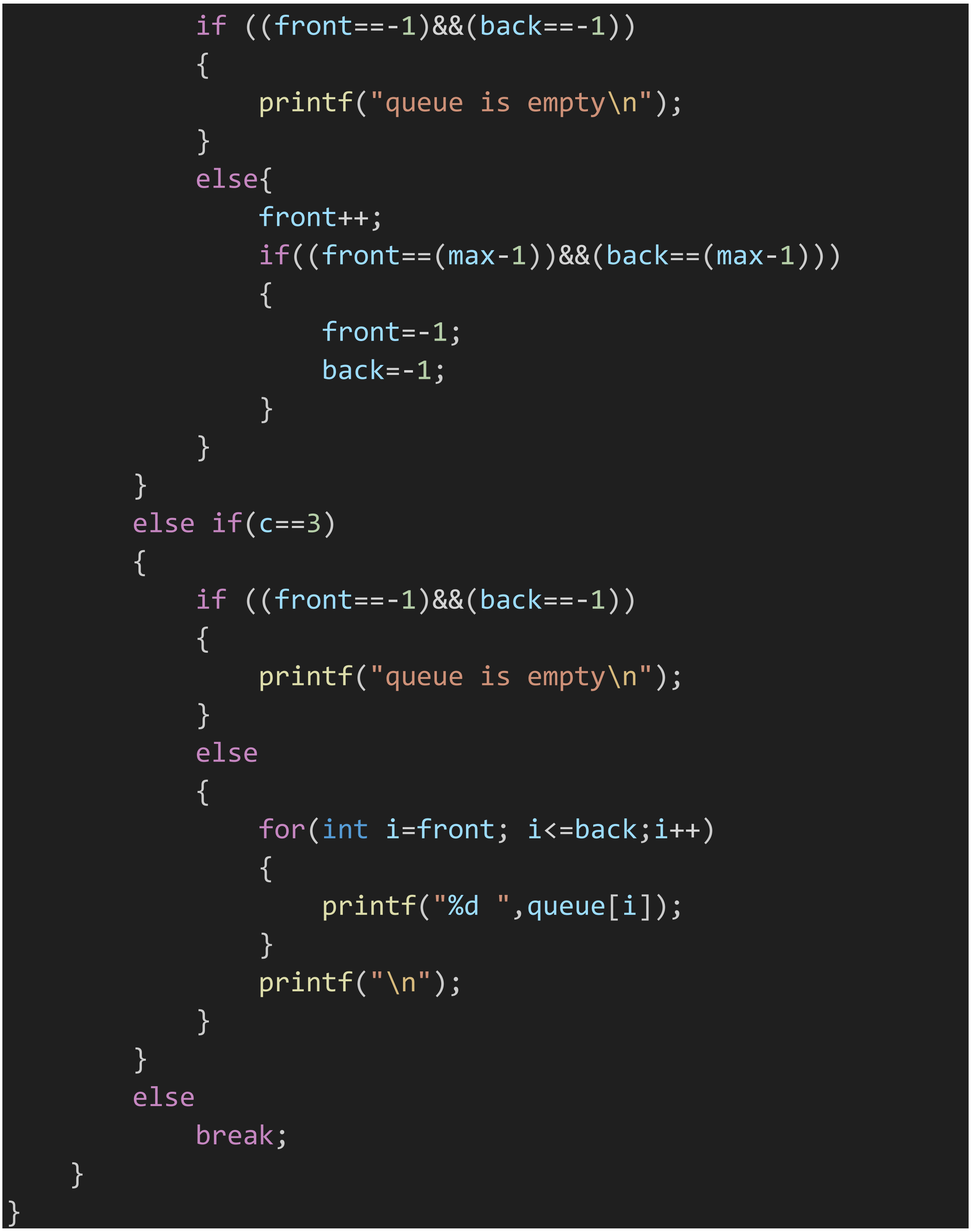
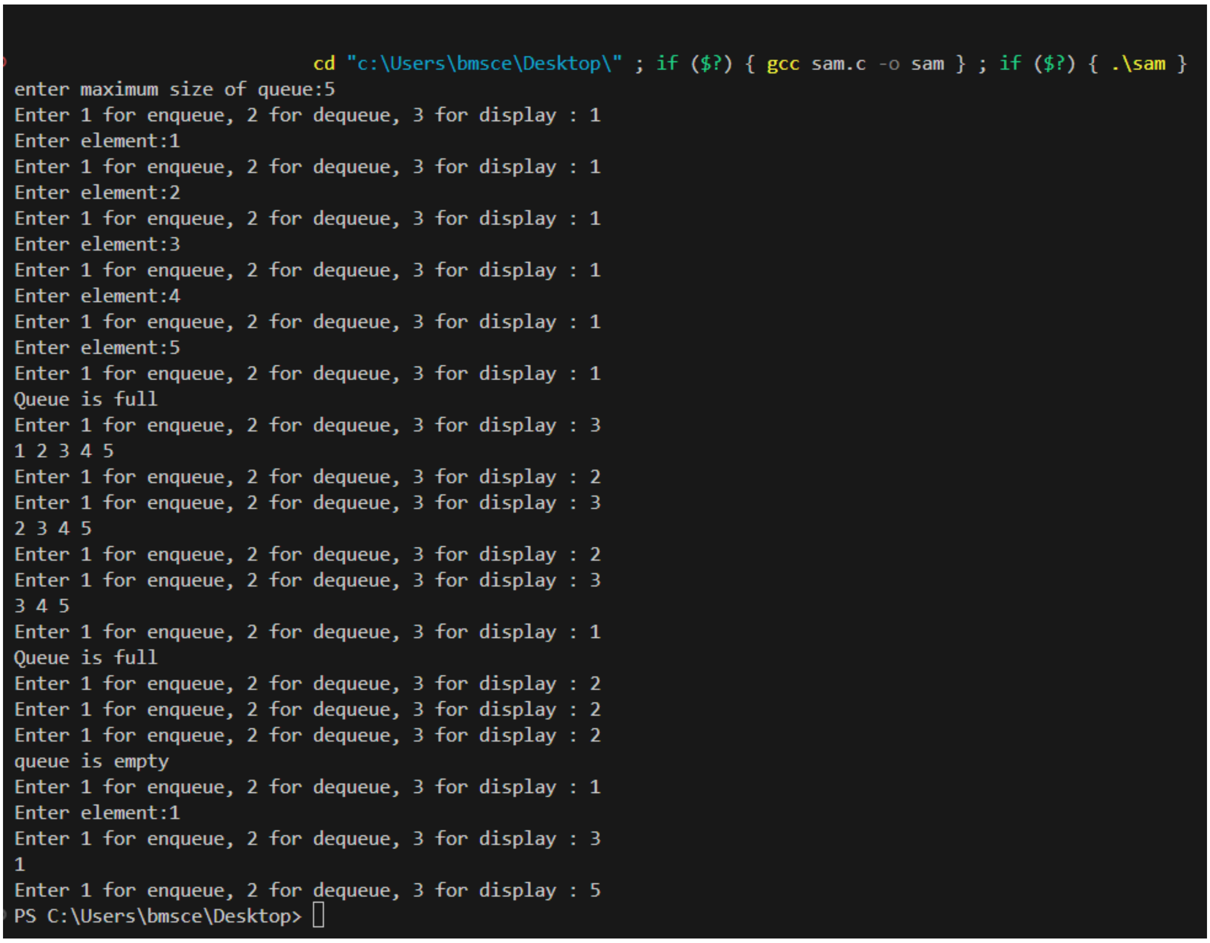


```
#include <stdio.h>

void main()
{
    int max;
    printf("enter maximum size of queue:");
    scanf("%d",&max);
    int queue[max];
    int front=-1;
    int back=-1;
    while(2>0)
    {
        printf("Enter 1 for enqueue, 2 for dequeue, 3 for
display : ");
        int c;
        scanf("%d",&c);
        if(c==1)
        {
            if(back==(max-1))
            {
                printf("Queue is full\n");
            }
            else
            {
                printf("Enter element:");
                int v;
                scanf("%d",&v);
                if ((front==-1)&&(back==-1))
                {
                    front++;
                    queue[++back]=v;
                }
                else
                    queue[++back]=v;
            }
        }
        else if(c==2)
        {
```

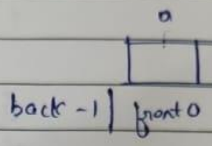


```c
            if ((front==-1)&&(back==-1))
            {
                printf("queue is empty\n");
            }
            else{
                front++;
                if((front==(max-1))&&(back==(max-1)))
                {
                    front=-1;
                    back=-1;
                }
            }
        }
        else if(c==3)
        {
            if ((front==-1)&&(back==-1))
            {
                printf("queue is empty\n");
            }
            else
            {
                for(int i=front; i<=back;i++)
                {
                    printf("%d ",queue[i]);
                }
                printf("\n");
            }
        }
        else
            break;
    }
}
```



```
                              cd "c:\Users\bmsce\Desktop\" ; if ($?) { gcc sam.c -o sam } ; if ($?) { .\sam }
enter maximum size of queue:5
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:1
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:2
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:3
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:4
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:5
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Queue is full
Enter 1 for enqueue, 2 for dequeue, 3 for display : 3
1 2 3 4 5
Enter 1 for enqueue, 2 for dequeue, 3 for display : 2
Enter 1 for enqueue, 2 for dequeue, 3 for display : 3
2 3 4 5
Enter 1 for enqueue, 2 for dequeue, 3 for display : 2
Enter 1 for enqueue, 2 for dequeue, 3 for display : 3
3 4 5
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Queue is full
Enter 1 for enqueue, 2 for dequeue, 3 for display : 2
Enter 1 for enqueue, 2 for dequeue, 3 for display : 2
Enter 1 for enqueue, 2 for dequeue, 3 for display : 2
queue is empty
Enter 1 for enqueue, 2 for dequeue, 3 for display : 1
Enter element:1
Enter 1 for enqueue, 2 for dequeue, 3 for display : 3
1
Enter 1 for enqueue, 2 for dequeue, 3 for display : 5
PS C:\Users\bmsce\Desktop>
```

Write a program to simulate the working of a queue of integers using an array. Provided the following operations, Insert delete, display.



```
# include <stdio.h>

void main()
 {

   int max;
   printf("Enter maximum size of the queue");
   scanf("%d", &max~~[illegible]~~);
   ~~int stack [~~
    int ~~queue~~ queue [max];
    int ~~front~~, back = -1;
    ~~int c =~~ int front = 0;
    printf("Enter 1 for ~~insert~~ enqueue, 2 for ~~delete~~ dequeue, 3 for display")
    ~~Scanf~~        insertion
    int c;
    scanf("%d", &c);
    if (c==1)
     {
        if(back == max-1)
          {
             printf("Queue Overflow");
          }
        else
          {
             printf("Enter the elements to be inserted");
             ~~queue for front++~~ int v;
             back ++
             queue[back] = scanf("%d," &v)
          }
     }
```

```
else if (c==2)
{
    if((front==0)~~and~~&&(back==-1))
    {
        printf("Queue is empty");
    }
    else
    {
        front++;
        if((front==(max-1))~~and~~&&(back==(max-1))){
        {
            front=0;
            back=-1;
        }
    }
}
else if (c==3)
{
    ~~for (int i=front; i<=back~~
    if ((front==0) && (back==-1))
    {
        printf("Queue is empty");
    }
    else
    {
        for (int i=front; i<=back; i++)
        {
            printf("%d", & queue[i]);
        }
    }
}
else
    printf("Invalid Input");
}
```

```
#include <stdio.h>

void main()
{
    int max;
    printf("Enter max-size of the queue");
    scanf("%d", &max);
    int queue [max].
    int front, back = -1;
   {while(True){
   {printf("Enter 1 for enqueue, 2 for dequeue, 3 for
            display , Anything else to exit ");
    int c;
    scanf("%d",&c),
    if (c==1)
     {
        ~~if ((front == -1) && (back == -1))~~   if(back==(max-1))
        ~~{~~                                    {
           ~~front ++;~~                             printf("Queue is full
           ~~back ++;~~                          }
      else
      {printf("Enter value to be inserted");
       {int v;
        scanf("%d", &v);
        if ((front == -1) && (back == -1))
        {
           front ++
           back ++
           queue[back] = v;
        }
        else
        {
               back ++;
           ~~queue b++~~
           queue [back] = v;
         }
    } }
~~else~~   else if (c==2)
      {
         if (front==-1)&&(back==-1))
         {
            printf("Queue is empty");
         }
         else
         {
            front++
            if ((front==(max-1))&&(back==(max-1))
            {
               front=-1;
               back=-1;
            }
         }
      }
    else if (c==3)
    {
       if((front==-1)&&(back==-1))
       {
          printf("Queue is empty");
       }
       else
       {
          for(int i=front, i<=back ; i++)
          {
             printf("%d ", &queue[i]);
          }
       }
   else ,
   {
     ~~printf("Invalid Input");~~ break;
   }
}}
```

Output:

```
Enter maximum size of queue: 5
Enter 1 for enqueue, 2 for dequeue, 3 for display, 4 for exit: 1
Enter ~~enter~~ element: 1
enter 1 for enqueue, 2 for dequeve, 3 for display, 4 for exit: 1
Enter element: 2
Enter 1 . . . .     " . .  . " " " " " " : 1
Enter element: 3
Enter 1 for  "   "   "   "   "   " : 1
Enter element: 4
Enter 1 for  "   "   "   "   " : 1
Enter element: 5
enter 1 for  "   "   "   "   " : 1
~~Queue~~ Queue is full
Enter 1 for  "   "   "   "   " : 3
1 2 3 4 5
Enter 1   "   "   "   " : 2
Enter 1   "   "   "   " : 3
2 3 4 5
Enter 1   "   "   "   " : 2
Enter 1   "   "   "   " : 3
3 4 5
Enter 1   "   "   "   " : 1
Queue is full
  "        "        "     : 2
  "        "        "     : 2
  "        "        "     : 2
Queue is now empty
  "        "        "     : 1
1
  "        "        "     : 3
1
  "        "        "     : 4
```