Page:
Date:

Experiment No. 02

Aim :- Implementation of Queue using Array for real-world application.

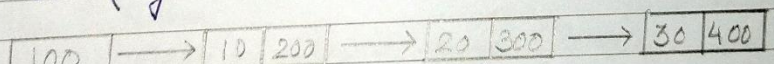
Objective :-

1. To introduce the concepts of data structures and analysis procedure.
2. To conceptualize linear data structures and its implementation for various real-world applications.

Theory :-

1.] Introduce to linear and non-linear data structure.

- Linear Data structure :-
  i] Organize data elements in a linear fashion.
  ii] Each data element is attached one after the other.
  iii] Only one other element can be directly reached while traversing

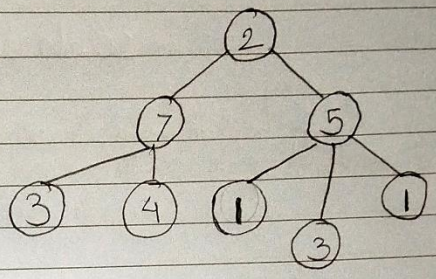


  iv] Eg :- Array, stack, Queue, Lists

- Non-linear Data structure :-
  i] Organization of the data element is not in a sequential fashion.
  ii] It is possible to attach a data item to search other data elements to reflect a special relationship among them.

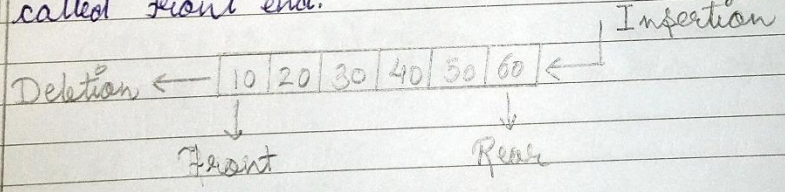
Page:
Date:

iii] Eg:- Graphs, Tree



2.] Introduce To Queue.

Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First out (FIFO). In a queue, new elements are added to queue from one end called Rear end and are always removed from other end called Front end.



Operation in Queue:-

- Enque → Adds an item in queue (Rear).
- Deque → Removes an item in queue (Front).
- Front → Get the front item from queue.
- Rear → Get the rear item from queue.

Page:
Date:

3] Algorithm :-

- QINSERT (Q, F, R, N, Y) :- Given F & R, pointers to front & rear elements of queue Q having N elements elements Y insertion in queue Q.

1. If R ≥ N
   then write ('Overflow')
   Return
2. R ← R+1 // [Increment rear Pointer]
3. Q[R] ← Y // [Insert element]
4. If F = 0 // [Is front pointer set properly ?]
   then F ← 1
   Return.

- QDELETE (Q, F, R) :- Given F & R, pointers to front & rear elements of queue Q, element Y is to be deleted.

1. If F = 0
   Then write ('Underflow')
   Return (0)
2. Y ← Q[F] // [Delet element]
3. If F = R // [Queue is empty]
   then F ← R ← 0
   else F ← F+1 // (increment front pointer)
4. Return [Y] // [Return element].

Page:
Date:

Example :-

A Best example of queue can be a single-lane one-way road, where the vechicle enters first. More examples can be seen as queues at the ticket windows and bus stops.

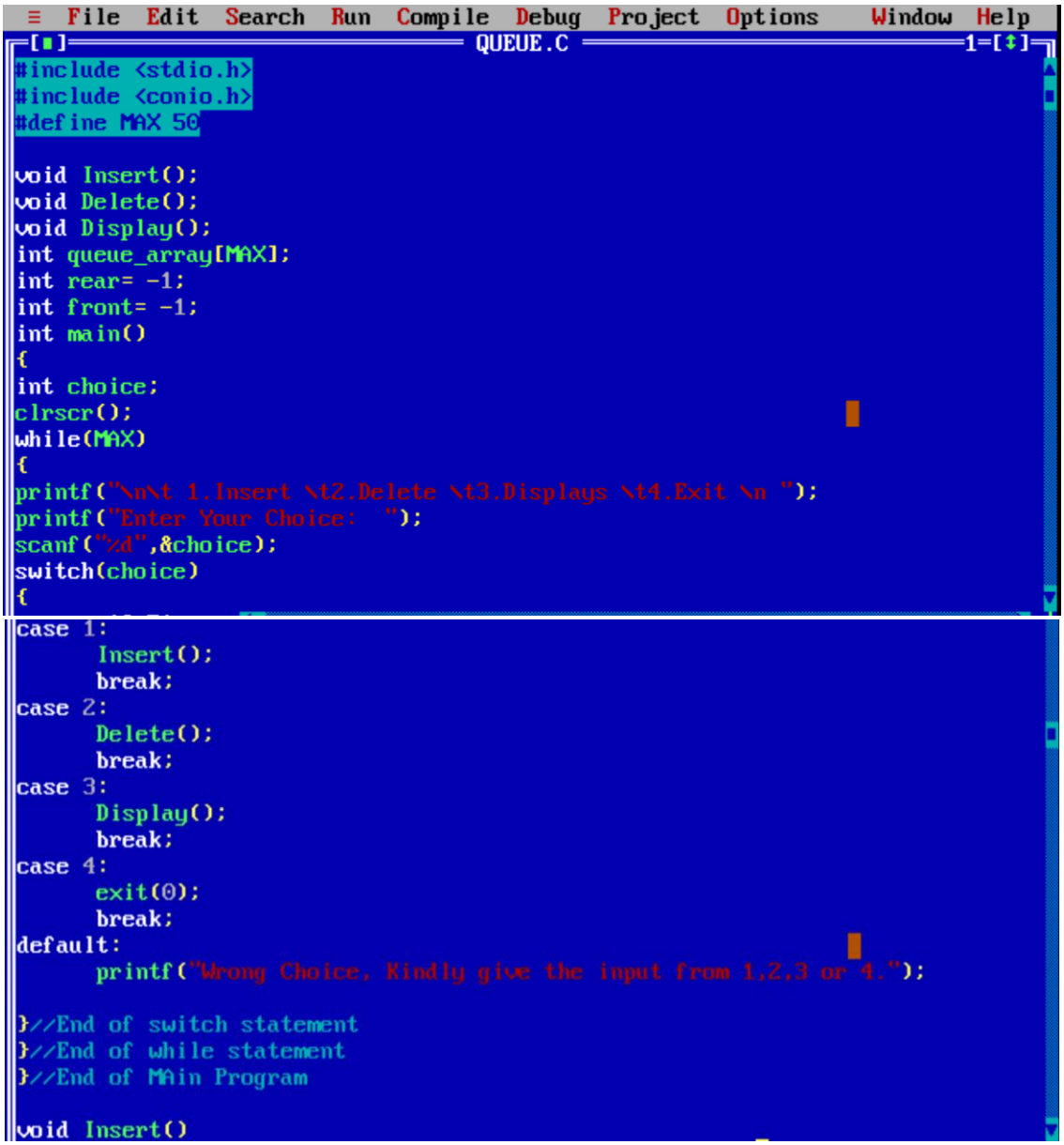
Conclusion :-

Thus, we have learn how to implement of queue using array. It is used when the values goes from one end and get remove from another in the form of First In First Out.

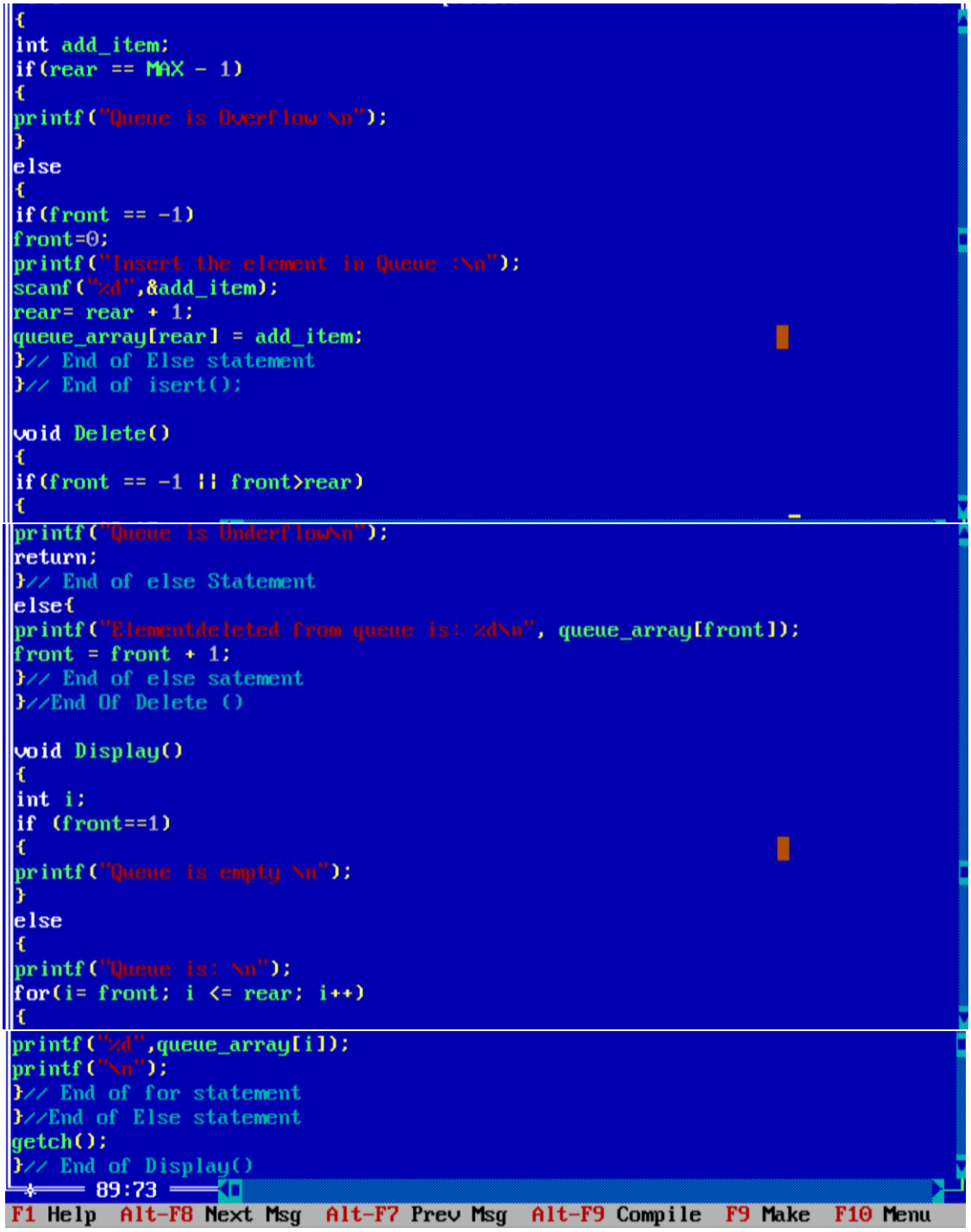
Outcome :-

Apply the concepts of queue for real-world application.

Program for Queue(QUEUE.C)



```
≡ File  Edit  Search  Run  Compile  Debug  Project  Options  Window  Help
[■]                          QUEUE.C                                1=[↕]
#include <stdio.h>
#include <conio.h>
#define MAX 50

void Insert();
void Delete();
void Display();
int queue_array[MAX];
int rear= -1;
int front= -1;
int main()
{
int choice;
clrscr();
while(MAX)
{
printf("\n\t 1.Insert \t2.Delete \t3.Displays \t4.Exit \n ");
printf("Enter Your Choice:  ");
scanf("%d",&choice);
switch(choice)
{
case 1:
      Insert();
      break;
case 2:
      Delete();
      break;
case 3:
      Display();
      break;
case 4:
      exit(0);
      break;
default:
      printf("Wrong Choice, Kindly give the input from 1,2,3 or 4.");

}//End of switch statement
}//End of while statement
}//End of MAin Program

void Insert()
```

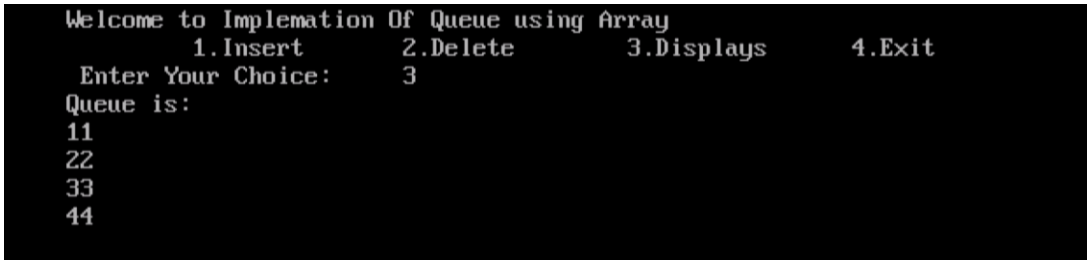


```
{
int add_item;
if(rear == MAX - 1)
{
printf("Queue is Overflow \n");
}
else
{
if(front == -1)
front=0;
printf("Insert the element in Queue :\n");
scanf("%d",&add_item);
rear= rear + 1;
queue_array[rear] = add_item;
}// End of Else statement
}// End of isert();

void Delete()
{
if(front == -1 || front>rear)
{
printf("Queue is Underflow\n");
return;
}// End of else Statement
else{
printf("Elementdeleted from queue is: %d\n", queue_array[front]);
front = front + 1;
}// End of else satement
}//End Of Delete ()

void Display()
{
int i;
if (front==1)
{
printf("Queue is empty \n");
}
else
{
printf("Queue is: \n");
for(i= front; i <= rear; i++)
{
printf("%d",queue_array[i]);
printf("\n");
}// End of for statement
}//End of Else statement
getch();
}// End of Display()
```

OUTPUT

1. Enqueue (insert the element)

2. Display the element



3. Dequeue (Deleting the element)