# Fundamentals of Data Structures

## Assignment 1

Ankit Das

19070122023

CS A1

Implement a Queue Data Structure

## queue_using_array.h

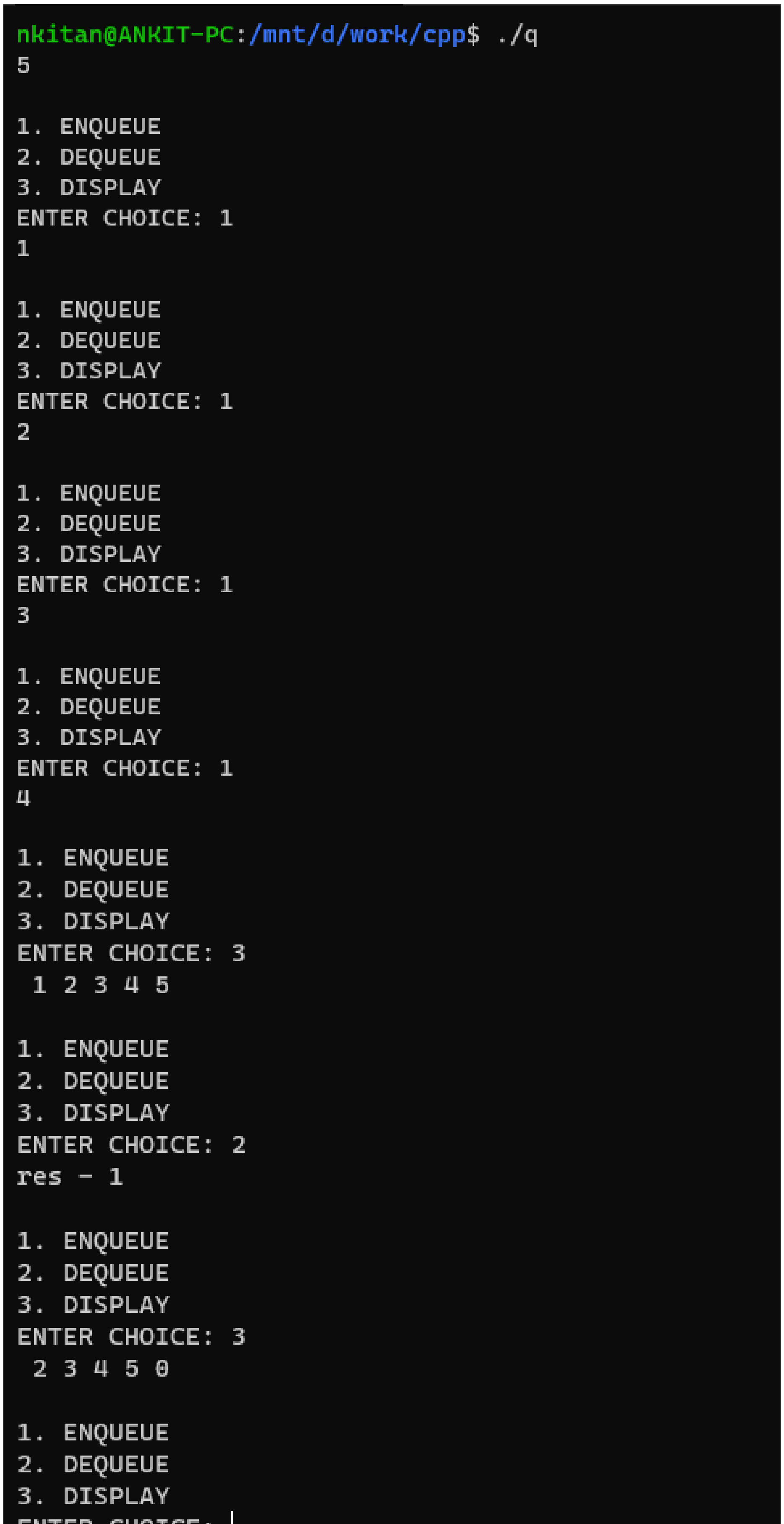
```
/*
Ankit Das
queue class in c++
30 July 2020
*/

#include<iostream>

class queue{
     int *array;     // pointer to the array
     int size;       // size of the queue
     int end = 0;    // element count

public:
     queue(int n){  // constructs a queue of size n
      size = n;
      array = (int*)malloc(sizeof(int)*n);
     }
     bool isEmpty(){
       if(end > 0){
           return false;
       }
       else{
           return true;
       }
     }

     void display(){  // display the queue
         unsigned int i = 0;
         for(i = 0; i < size; i++){
             printf(" %d",*(array + i));
         }
         printf("\n");
     }
     void enqueue(int)
// takes value n and adds it to the end of the queue
        if(end == size){
// check if end of array is reached
std::cout<<"Overflow Error: Cannot add more elements!
"<<std::endl;
         }
         else{
```

```
         *(array + end) = n;     // add element to end
         end++; // increment element count when added
        }
     }
 int dequeue(){
// return the value at the beginning and shifts the q
ueue forward
   if(!isEmpty()){  // check if elements are left
            int value = *(array);
            for(int i = 0; i < (end - 1); i++)
                *(array + i) = *(array + (i + 1));
            *(array + end - 1) = 0;
             // set vacant position to 0
            end--
;             // decrement element count
            return value;
            }
        else{
std::cout<<"UnderflowError: No more elements left!"<<
std::endl;
            return 0;
        }
     }
};
```

Queue.cpp

```
#include "queue_using_array.h"

int main(){

    int n = 0;
    std::cin>>n;

    class queue Q(n);
    int result = 0;
    int choice = 0;
    bool run = true;

while(run){
    std::cout<<"\n1. ENQUEUE\n";
    std::cout<<"2. DEQUEUE\n";
    std::cout<<"3. DISPLAY\n";
    std::cout<<"ENTER CHOICE:  ";
    std::cin>>choice;



    switch(choice){
    case 1:
        std::cin>>result;
        Q.enqueue(result);
        break;

    case 2:
        result = Q.dequeue();
        printf("data - %d\n",result);
        break;
    case 3:
        Q.display();
        break;
    case 4:
        run = false;
        break;
    }



}
    return 0;
}
```

OUTPUT-



```
nkitan@ANKIT-PC:/mnt/d/work/cpp$ ./q
5

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 1
1

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 1
2

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 1
3

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 1
4

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 3
 1 2 3 4 5

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 2
res - 1

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
ENTER CHOICE: 3
 2 3 4 5 0

1. ENQUEUE
2. DEQUEUE
3. DISPLAY
```