**//Array Based Implementation of QUEUE**

#include<iostream>

using namespace std;

#define SIZE 10

class Queue

{

int a[SIZE];

int rear; //same as tail

int front; //same as head

public:

Queue()

{

rear = front = -1;

}

//declaring enqueue, dequeue and display functions

void enqueue(int x);

int dequeue();

void display();

};

// function enqueue - to add data to queue

void Queue :: enqueue(int x)

{

if(front == -1) {

front++;

}

if( rear == SIZE-1)

{

cout << "Queue is full";

}

else

{

a[++rear] = x;

}

}

// function dequeue - to remove data from queue

int Queue :: dequeue()

{

//cout<<a[front];

//front++;

return a[++front]; // following approach [B], explained above

//cout<<a[++front];

//++front;

}

// function to display the queue elements

void Queue :: display()

{

int i;

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

{

cout << a[i] << endl;

}

}

// the main function

int main()

{

Queue q;

q.enqueue(10);

q.enqueue(100);

q.enqueue(1000);

q.enqueue(1001);

q.enqueue(1002);

q.dequeue();

q.enqueue(1003);

q.dequeue();

q.dequeue();

q.enqueue(1004);

q.display();

return 0;

}