مقرر الخوارزميات و بنى المعطيات ١

الجلسة العملي الرابعة

(الفصل الثاني ٢٠٢٤-٢٠٢٥)

تحقيق الرتل باستخدام المصفوفات

```
#include "stdafx.h"
#include<iostream>
using namespace std;
const int QUEUE_CAPACITY=128;
class Queue
{ /****  function members ****/
public:
Queue();
bool empty( ) const ;
void addQ(const int & value);
QueueElement front( ) const;
void removeQ();
void display() const ;
```

```
private:  /****  data members ****/
// implement queue as array
int myArray[QUEUE_CAPACITY];
//two index one refer to the front of array and other t
the rear
 int myFront, myBack;
};
 //end of class declaration
```

```
// constructor implementation
inline Queue::Queue()
{ myFront=myBack=0; }

// function :testing if queueis empty
 inline bool Queue::empty( ) const
 { return (myFront==myBack)  ;}
```

```
// function :adding element to the rear(back) of queue

void Queue::addQ(const int & value)
{    int newBack=(myBack+1)%QUEUE_CAPACITY;
 if (newBack==myFront)
            cout<<"**** QUEUE IS FULL
****"<<endl;
  else
{          myArray[myBack]=value;
myBack=newBack;}
}
```

```
// function :return the element at the front of the queue
int Queue::front( ) const
{  if (!empty() )
     return myArray[myFront];
cout<<"*** QUEUE IS EMPTY *** \n";}

// function :removing the element at the front of the queue
void Queue::removeQ( )
{          if (empty( ))
 cout<<"**** QUEUE IS EMPTY ****\n";
           else
 myFront=(myFront+1)%QUEUE_CAPACITY;
}
```

```
// function :displaying the element of the queue
void Queue::display() const
{
for (int i=myFront ; i<myBack ; i++)
cout<<myArray[i]<<"  ";
}
// end of implementation function of the class Queue
```

```
//Main function
void main() {
Queue q1; // creating object q1
//calling function empty
cout<<"Queue is empty "<<q1.empty( )<<endl;
// adding element 7,77,5,55 to quque
q1.addQ(7);
q1.addQ(77);
q1.addQ(5);
 q1.addQ(55);
cout<<"Queue is empty "<<q1.empty( )<<endl;
// displaying the element of the queue by calling display function
   q1.display( );
// removing two element at the front of the queue by Calling remove function
q1.removeQ( );
q1.removeQ( );
cout<<"\n\n";
 q1.display();
 cout<<"\n\n";
system("pause");}
```

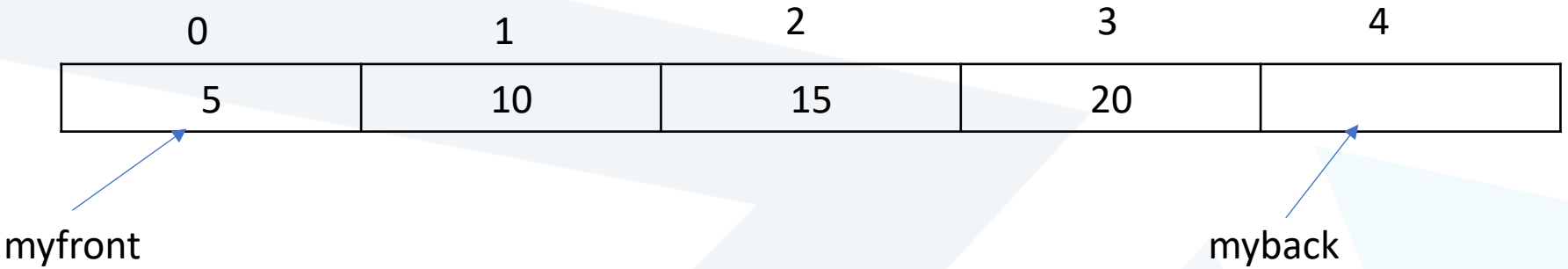
فيكون الخرج الناتج على الشكل التالي:

```
Queue is empty 1
Queue is empty 0
7  77  5  55

5  55

Press any key to continue . . .
```

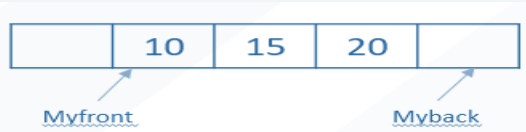
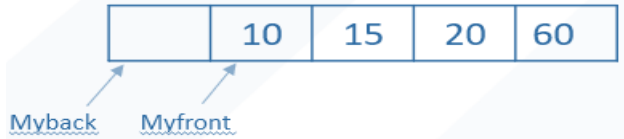
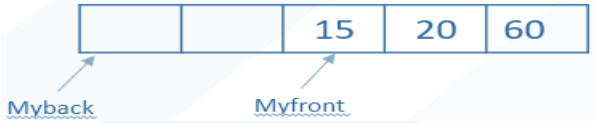
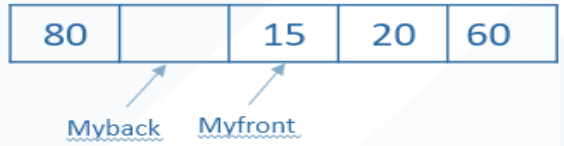
بفرض لدينا الرتل التالي:




```
int s=0;
for (int i=5;i<=8;i++)
if(i%2==0)
   Q.add(i*10);
else
   {s=s+Q.front();
   Q.remove();}
cout<<“s=“<<s;
```

أوجد خرج الكود البرمجي التالي المطبق على الرتل السابق وأعد رسم الرتل بعد التعديل.

حالة الرتل في كل تكرار

i	s	Queue
5	5	
6		
7	15	
8		

الخرج النهائي:

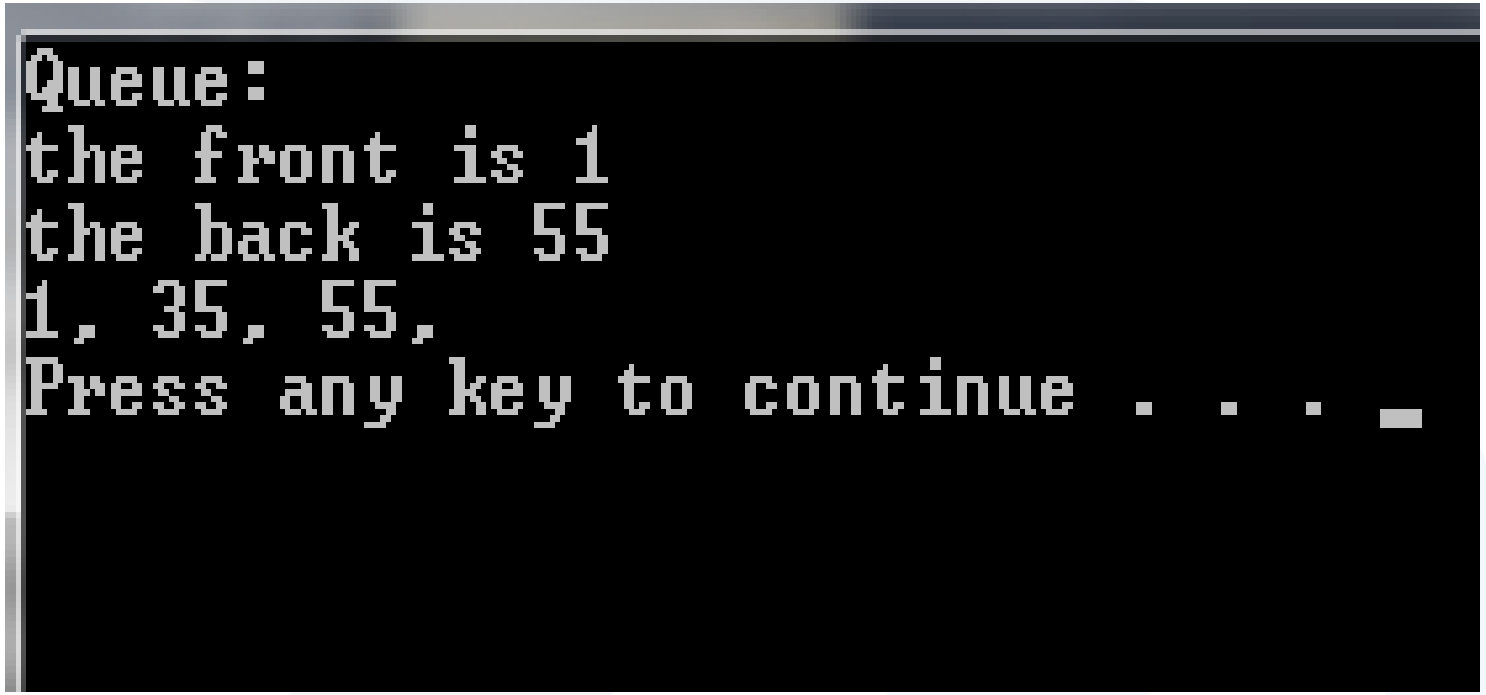
S=15

اكتب برنامج يقوم بإنشاء رتل يحوي الأرقام التالية
بالترتيب ١ ٣٥ ٥٥
ومن ثم قم بطباعة العنصر الأول والعنصر الأخير
ثم قم بإفراغ الرتل وطباعة عناصره

```
#include <iostream>
#include <queue> //   تضمين مكتبة queue القياسية
using namespace std;
int main() {
queue<int> num; //  إنشاء رتل
 num.push(1);
 num.push(35);          // دفع عناصر إلى مؤخرة الرتل
 num.push(55);
cout << "Queue: "<<endl;
  cout << "the front is "<< num.front()<<endl; // طباعة العنصر في مقدمة الرتل
  cout << "the back is "<< num.back()<<endl; ; // طباعة العنصر في نهاية الرتل

while(!num.empty()) {
    // print the element                    //  إفراغ الرتل وطباعة عناصره
    cout << num.front() << ", ";
    num.pop();}
cout << endl;    system("pause");    return 0;    }
```

فيكون الخرج الناتج على الشكل التالي:



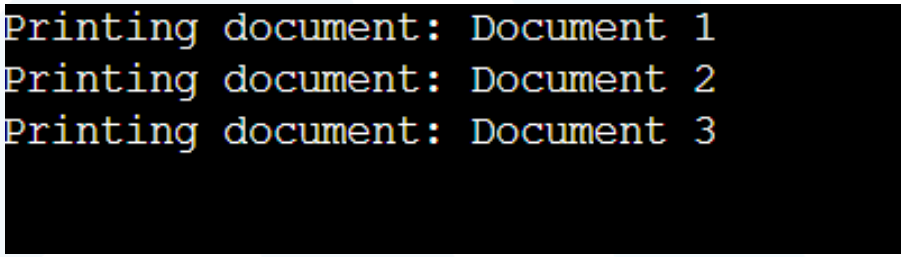
أوجد خرج البرنامج التالي:

```
#include <iostream>
#include <queue>
#include <string>
int main() {
      queue<string> printQueue;

       printQueue.push("Document 1");
       printQueue.push("Document 2");
       printQueue.push("Document 3");

       while (!printQueue.empty()) {
          string document = printQueue.front();
         cout << "Printing document: " << document << endl;
           printQueue.pop();
       }
       system("pause");
       return 0;
   }
```

فيكون الخرج الناتج على الشكل التالي:



جَامعة
المَنارة
MANARA UNIVERSITY

انتهت الجلسة