以下例子主要在說明如何在純種 C 語言中實作出一個佇列(Queue, 排隊),
至於 Queue 的定義以及用途請參考計概課本或是資料結構課本,
或者用 google.com 找一找應該也可看到許多實例與說明!

```
tsaiwn@magpie % cat -n que.c
     1  // que.c -- use C to inplement Queue as a circular queue
     2  enum{ NELEMENT=99 };  // good habit
     3  static long data[NELEMENT];
     4  static int phead=0, ptail=0; // phead 指向queue的前頭, ptail指向尾巴
     5  initQueue(void){ phead = ptail = 0; }
     6  void enque(long);
     7  long deque(void);
     8  long front(void) { return data[phead]; }
     9  int isempty(void) { return (phead == ptail) ;}
    10  int isfull(void) { return ((ptail+1) % NELEMENT) == phead; } //留一個不用!
    11  ////// //////
    12   void push(long x) { enque(x); }
    13   long pop(void) { return deque( ); }
    14   long top(void) { return front( ); }
    15   int empty(void) { return isempty( );}
    16   int full(void) { return isfull( );}
    17  /////// /////// /////// /////// /////// ///////
    18  void enque(long x) {
    19      if( isfull() ) { return; }  // may want to give some error message?
    20      data[ptail] = x;
    21      ptail++;
    22      ptail %=  NELEMENT;    /* x %= y;  means x = x % y; */
    23  }
    24  long deque(void) {
    25     if( !isempty() ) {
    26        long tmp=data[phead];
    27        phead++; phead %= NELEMENT;
    28        return tmp;
    29     } // if !isempty
    30  }
tsaiwn@magpie % gcc -c que.c        <=== 要求只要生出機器碼檔案 que.o
tsaiwn@magpie % nm que.o          <=== 要求列出 que.o 內的 symbols
0000018c C data
000000f4 T deque
0000009e T empty
000000b2 T enque
00000019 T front
000000a8 T full
00000000 T initQueue
0000002a T isempty
00000040 T isfull
00000000 D phead
00000087 T pop
00000004 D ptail
00000071 T push
00000094 T top
tsaiwn@magpie % cat -n tstque.c
     1  /// tstque.c -- example to use the circular Queue written in C language
     2  // by Wen-Nung Tsai, tsaiwn@csie.nctu.edu.tw
     3  // gcc -c que.c
     4  // gcc tstque.c que.o ; ./a.out
     5  /*********************************/
     6  #include <stdio.h>
     7  int main( ) {
     8     printf("Test queue written in C Language.\n");
     9     enque(53);
    10     enque(770);
    11     enque(880);
    12     while( !isempty( ) ) {
    13        printf("%ld ", deque( ) );
    14     }
    15     printf("\n===bye\n");
    16     return 0;
    17  }
tsaiwn@magpie % gcc tstque.c que.o
tsaiwn@magpie % ./a.out
Test queue written in C Language.
53 770 880
===bye
tsaiwn@magpie % echo p621
```

用 class 把 queue 有關 的 data 以及 functions 封藏 (encapsulate)起來!
以下連測試的主程式也寫在一起是偷懶的寫法!

```
tsaiwn@magpie % cat -n myque.cpp
     1  // myque.cpp  -- queue class by tsaiwn@csie.nctu.edu.tw
     2  // g++ myque.cpp ; ./a.out
     3  // 以下的 #ifndef 是用來 check 是否爲 GNU 的 g++ ?
     4  // 若不是則使用舊式的寫法! 例如 Turbo C++ 3.0 只認識舊式的寫法!
     5  #ifndef __GNUG__
     6    #include <iostream.h>
     7  #else
     8    #include<iostream>
     9    using namespace std;
    10  #endif
    11  class MyQueue {
    12      enum{ NELEMENT=99 };  // good habit
    13      long data[NELEMENT];
    14      int phead, ptail;   // phead 指向 queue 的前頭, ptail指向尾巴
    15      void initQueue(void){ phead = ptail = 0; }
    16    public:
    17      MyQueue(void) { initQueue( ); }
    18      void enque(long);
    19      long deque(void);
    20      long front(void) { return data[phead]; }
    21      int isempty(void) { return (phead == ptail) ;}
    22      int isfull(void) { return ((ptail+1) % NELEMENT) == phead; }
    23      ////// //////  注意 circular queue 的表達方式
    24      void push(long x) { enque(x); }
    25      long pop(void) { return deque( ); }
    26      long top(void) { return front( ); }
    27      int empty(void) { return isempty( );}
    28      int full(void) { return isfull( );}
    29  }; // class MyQueue
    30  ////// ////// //////
    31  void MyQueue::enque(long x) {
    32      if( isfull() ) { return; }  // may want to give some error message?
    33      data[ptail] = x;
    34      ptail++;
    35      ptail %=  NELEMENT;    /* x %= y;  means x = x % y; */
    36  }
    37  long MyQueue::deque(void) {
    38     if( !isempty() ) {
    39        long tmp=data[phead];
    40        phead++; phead %= NELEMENT;
    41        return tmp;
    42     } // if !isempty
    43  }
    44  ////// ////// ////// ////// ////// //////
    45  int main( ) {
    46      MyQueue xo;
    47      xo.enque(53);
    48      xo.push(770);
    49      xo.push(880);
    50      while(!xo.empty( ) ) {
    51         cout << xo.top( ) << " "; xo.pop( );
    52      }
    53      cout << endl;   // new line
    54      return 0;
    55  }
    56  // 此例是把 MyQueue 與 主程式寫在一個檔案, 這是偷懶且不好的方法
    57  // 正確方式應該把 MyQueue 獨立, 且應把宣告寫在 .h 檔案, 實作寫 .cpp檔案
    58  // 但是若寫爲 template class 則通常全部寫在一個 .h 檔案.
tsaiwn@magpie % gcc myque.cpp
tsaiwn@magpie % ./a.out
53 770 880

tsaiwn@magpie % echo 622
```

```
tsaiwn@magpie % cat -n myque2.h
     1  // myque2.h  -- queue class by tsaiwn@csie.nctu.edu.tw
     2  // g++ -c myque2.cpp
     3  // g++ testmyq2.cpp myque2.o
     4  #ifndef __MYQUE2__
     5  #define __MYQUE2__
     6  class MyQueue {
     7      enum{ NELEMENT=99 };  // good habit
     8      long data[NELEMENT];
     9      int phead, ptail;   // phead 指向 queue 的前頭, ptail指向尾巴
    10      void initQueue(void){ phead = ptail = 0; }
    11    public:
    12      MyQueue(void) { initQueue( ); }
    13      void enque(long);
    14      long deque(void);
    15      long front(void) { return data[phead]; }
    16      int isempty(void) { return (phead == ptail) ;}
    17      int isfull(void) { return ((ptail+1) % NELEMENT) == phead; }
    18      ////// //////  注意 circular queue 的表達方式
    19      void push(long x) { enque(x); }
    20      long pop(void) { return deque( ); }
    21      long top(void) { return front( ); }
    22      int empty(void) { return isempty( );}
    23      int full(void) { return isfull( );}
    24  }; // class MyQueue
    25  ////// ////// //////
    26  #endif
tsaiwn@magpie % cat -n myque2.cpp
     1  // myque2.cpp  -- 把宣告寫在 myque2.h
     2  // CopyLeft by tsaiwn@csie.nctu.edu.tw
     3  #include "myque2.h"
     4  // some implementations are in myque2.h
     5  ////// ////// //////
     6  void MyQueue::enque(long x) {
     7      if( isfull() ) { return; }  // may want to give some error message?
     8      data[ptail] = x;
     9      ptail++;
    10      ptail %=  NELEMENT;    /* x %= y;  means x = x % y; */
    11  }
    12  long MyQueue::deque(void) {
    13     if( !isempty() ) {
    14        long tmp=data[phead];
    15        phead++; phead %= NELEMENT;
    16        return tmp;
    17     } // if !isempty
    18  }
tsaiwn@magpie % g++ -c myque2.cpp  ;  nm myque2.o
00000056 T _ZN7MyQueue5dequeEv
00000000 T _ZN7MyQueue5enqueEl
00000000 W _ZN7MyQueue6isfullEv
00000000 W _ZN7MyQueue7isemptyEv
tsaiwn@magpie % cat -n testmyq2.cpp
     1  //testmyq2.cpp -- copyLeft by tsaiwn@csie.nctu.edu.tw
     2  // 以下的 #ifndef 是用來 check 是否爲 GNU 的 g++ ?
     3  // 若不是則使用舊式的寫法! 例如 Turbo C++ 3.0 只認識舊式的寫法!
     4  #ifndef __GNUG__
     5    #include <iostream.h>
     6  #else
     7    #include<iostream>
     8    using namespace std;
     9  #endif
    10  #include "myque2.h"
    11  ////// ////// ////// ////// ////// //////
    12  int main( ) {
    13      MyQueue xo;
    14      xo.enque(53);
    15      xo.push(770);
    16      xo.push(880);
    17      while(!xo.empty( ) ) {
    18         cout << xo.top( ) << " "; xo.pop( );
    19      }
    20      cout << endl;   // new line
    21      return 0;
    22  }
tsaiwn@magpie % g++ testmyq2.cpp myque2.o
tsaiwn@magpie % ./a.out
53 770 880
tsaiwn@magpie % echo p623
```

此例只是把前面 myque2.h 與 myque2.cpp 再合寫成 template class
注意以下的 Line 4, Line 25, line 26, Line 32, Line 33

```
tsaiwn@magpie % cat -n myque3.h
     1  // myque3.h  -- template queue class by tsaiwn@csie.nctu.edu.tw
     2  #ifndef __MYQUE3__
     3  #define __MYQUE3__
     4  template <class HAHA>
     5  class MyQueue {
     6      enum{ NELEMENT=99 };  // good habit
     7      HAHA data[NELEMENT];
     8      int phead, ptail;   // phead 指向 queue 的前頭, ptail指向尾巴
     9      void initQueue(void){ phead = ptail = 0; }
    10    public:
    11      MyQueue(void) { initQueue( ); }
    12      void enque(HAHA);
    13      HAHA deque(void);
    14      HAHA front(void) { return data[phead]; }
    15      int isempty(void) { return (phead == ptail) ;}
    16      int isfull(void) { return ((ptail+1) % NELEMENT) == phead; }
    17      ////// //////  注意 circular queue 的表達方式
    18      void push(HAHA x) { enque(x); }
    19      HAHA pop(void) { return deque( ); }
    20      HAHA top(void) { return front( ); }
    21      int empty(void) { return isempty( );}
    22      int full(void) { return isfull( );}
    23  }; // class MyQueue
    24  ////// ////// //////
    25  template <class HAHA>
    26  void MyQueue<HAHA>::enque(HAHA x) {
    27      if( isfull() ) { return; }  // may want to give some error message?
    28      data[ptail] = x;
    29      ptail++;
    30      ptail %=  NELEMENT;    /* x %= y;  means x = x % y; */
    31  }
    32  template <class HAHA>
    33  HAHA MyQueue<HAHA>::deque(void) {
    34     if( !isempty() ) {
    35        HAHA tmp=data[phead];
    36        phead++; phead %= NELEMENT;
    37        return tmp;
    38     } // if !isempty
    39  }
    40  #endif
tsaiwn@magpie % cat -n testmyq3.cpp
     1  //testmyq3.cpp -- using template class MyQueue in myque3.h
     2  // CopyLeft by tsaiwn@csie.nctu.edu.tw
     3  #include "myque3.h"
     4  // 以下的 #ifndef 是用來 check 是否爲 GNU 的 g++ ?
     5  // 若不是則使用舊式的寫法! 例如 Turbo C++ 3.0 只認識舊式的寫法!
     6  #ifndef __GNUG__
     7    #include <iostream.h>
     8  #else
     9    #include<iostream>
    10    using namespace std;
    11  #endif
    12  ////// ////// ////// ////// ////// //////
    13  int main( ) {
    14      MyQueue<long> xo;
    15      xo.enque(53);
    16      xo.push(770);
    17      xo.push(880);
    18      while(!xo.empty( ) ) {
    19         cout << xo.top( ) << " "; xo.pop( );
    20      }
    21      cout << endl;   // new line
    22      return 0;
    23  }
tsaiwn@magpie % g++ testmyq3.cpp
tsaiwn@magpie % ./a.out
53 770 880

tsaiwn@magpie % echo p624
```

這是直接使用 C++ STL 類別程式庫中的 queue, 它是一個 template class.
該 queue 的加入 queue 也用 push, 所以我們用 #define enque push
這個 macro definition 把程式中用到 enque 的改爲 push
另外, queue 的頭部應該也可以用 top( ), 但測試有問題所以#define 成爲 front

```
tsaiwn@magpie % cat -n testmyq4.cpp
     1  //testmyq4.cpp -- using class queue in STL
     2  // CopyLeft by tsaiwn@csie.nctu.edu.tw
     3  #define enque push
     4  #define top front
     5  // 以下的 #ifndef 是用來 check 是否爲 GNU 的 g++ ?
     6  // 若不是則使用舊式的寫法! 例如 Turbo C++ 3.0 只認識舊式的寫法!
     7  #ifndef __GNUG__
     8    #include <iostream.h>
     9    #include <stl.h>
    10  #else
    11    #include<iostream>
    12    #include <queue>
    13    using namespace std;
    14  #endif
    15  ////// ////// ////// ////// ////// //////
    16  int main( ) {
    17      queue<long> xo;       // queue is a template class in STL
    18      xo.enque(53);
    19      xo.push(770);
    20      xo.push(880);
    21      while(!xo.empty( ) ) {
    22         cout << xo.top( ) << " "; xo.pop( );
    23      }
    24      cout << endl;   // new line
    25      return 0;
    26  }
tsaiwn@magpie % g++ testmyq4.cpp
tsaiwn@magpie % ./a.out
53 770 880

tsaiwn@magpie % echo p625
```

http://www.csie.nctu.edu.tw/~tsaiwn/cpp/handouts/queue/