```
//

    #include<iostream>
    #include<cstdio>
    #include<iomanip>
    #include<cassert>
    #include<string>
    #include<cmath>
    #include<iostream>
    #include<cmath>
    #include<sstream>
    #include<stdio.h>
    #include<stdlib.h>
     using namespace std;

  struct queuenode;
  typedef queuenode*queue;
  struct queuenode
   {
    char item;
    queue qlink;
   };

  void initqueue(queue&q)
   {
    q = new queuenode;
    q->qlink = q;
   }

  bool empty(queue q)
   {
    return q->qlink == q;
   }

  void enqueue(queue&q,char newitem)
   {
    cout << "enqueue(" << newitem << ")" << endl;
    q->item = newitem;
    queue temp = new queuenode;
    temp->qlink = q->qlink;
    q->qlink = temp;
    q = temp;
   }
//
```

```
//

  char dequeue(queue&q)
   {
    assert(not empty(q));
    char rslt = q->qlink->item;
    q->qlink = q->qlink->qlink; // memory leak
    return rslt;
   }

  void traverselinks(queue q,queue lnk)
   {
    //cout << "traverselinks" << endl;
    if(q != lnk)
     {
      cout << " " << lnk->item;
      traverselinks(q,lnk->qlink);
     }
    else cout << endl;
   }

  void traverse(queue q)
    // writes items of q in order
   {
    traverselinks(q,q->qlink);
   }

  int main()
   {
    queue q;
    initqueue(q);
    enqueue(q,’A’);
    traverse(q);
    enqueue(q,’B’);
    traverse(q);
    enqueue(q,’M’);
    traverse(q);
    enqueue(q,’Q’);
    traverse(q);
    cout << dequeue(q) << endl;
    cout << dequeue(q) << endl;
    cout << dequeue(q) << endl;
    cout << dequeue(q) << endl;
    cout << dequeue(q) << endl;
    return 1;
    }
//
```