Programming Assignment 6: Queue with Dynamic Array

A queue is a container that elements are inserted and removed in the *last in last out* (*or first in first out*) order. Write a C project to define a data type representing queues using dynamic array of integers and define the following queue operations:

```
#define SEGMENT 50 // Segment size.
// Type definition of queues using a dynamic array.
// The queue elements of integers.
typedef struct {
  int *elem; // Pointer to queue container.
  int head; // Index of the queue head.
  int tail; // Index of the queue tail.
  int capacity; // Capacity of a queue.
} Queue;
```

(1) **void** initial_queue(Queue *): Set a queue to empty, i.e., reset head and tail of a queue.

(2) **void** enqueue(Queue *, **int**): Insert an element to the tail of a queue.

(3) **int** dequeue(Queue *): Remove an element from the head of a queue.

(4) **int** head(Queue): Get the element at the head of a queue.

(5) **int** is_empty(Queue): Check if a queue is empty or not.下

(6) **int** get_size(Queue: Get size of queue (not a basic operation).

(7) **void** clear(Queue *); Clear queue and set capacity to one segment (not a basic operation).

(8) **void** print_queue(Queue): Print elements of a queue from the head to the tail (not a basic operation).

In the main program declares a queue Q. Use random number generator to get the number trials, maximum 10 trials, and perform the following operations in each trial:

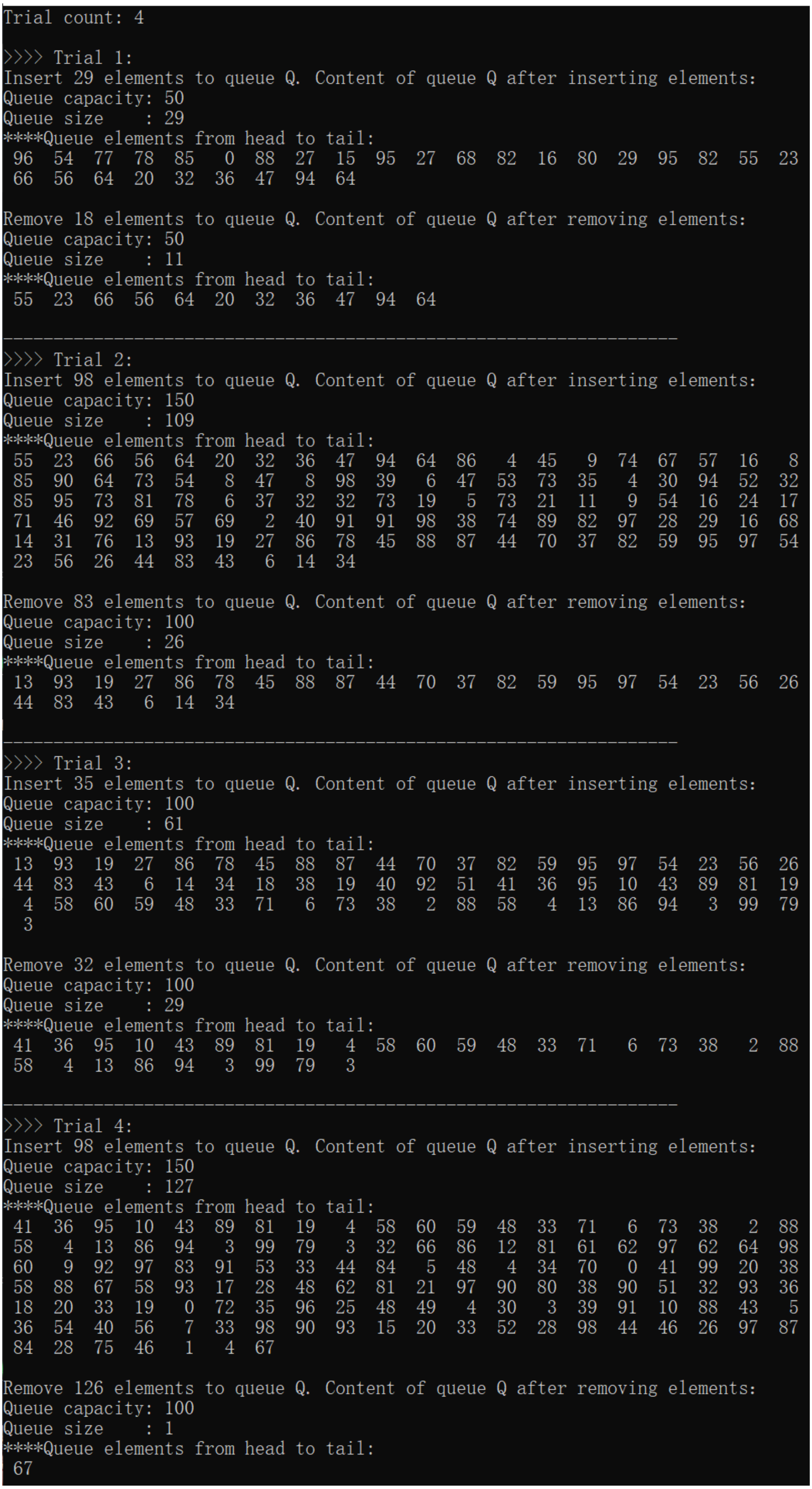
(1) Generate a random number enqueue_count, number between 1 and 100, insert enqueue_count elements to queue Q, and print queue Q from the head to the tail.

(2) Generate a random number dequeue_count, number between 1 and current queue size, remove dequeue_count elements to queue Q, and print queue Q from the head to the tail.

Assume the value of stack elements is a random number between 0 and 99. Program solutions: queue_dynamic_array.dev, queue_dynamic_array.h, queue_dynamic_array.c, and queue_dynamic_array_main.c. Place the above files in directory named **assgn6_DXXXXXXX** and compress the directory.

In this assignment, you must submit two files: the source code of the solution **assgn6_DXXXXXXX.YYY** (**YYY** is the extended compression file name **zip**, **rar**, or **7z**) (**80%**) and the assignment report **assgn6_DXXXXXXX.pdf** (**20%**), where **DXXXXXXX** is your student ID. The assignment report should explain how you deign and implement queues using dynamic array. Sample outputs: Programming assignment 5 is due by **11:59 pm, Sunday, December 18**. Submit your solution and the report to **iLearn2**.

Example of program execution:



```
Trial count: 4

>>>> Trial 1:
Insert 29 elements to queue Q. Content of queue Q after inserting elements:
Queue capacity: 50
Queue size    : 29
****Queue elements from head to tail:
 96  54  77  78  85   0  88  27  15  95  27  68  82  16  80  29  95  82  55  23
 66  56  64  20  32  36  47  94  64

Remove 18 elements to queue Q. Content of queue Q after removing elements:
Queue capacity: 50
Queue size    : 11
****Queue elements from head to tail:
 55  23  66  56  64  20  32  36  47  94  64

-------------------------------------------------------------------
>>>> Trial 2:
Insert 98 elements to queue Q. Content of queue Q after inserting elements:
Queue capacity: 150
Queue size    : 109
****Queue elements from head to tail:
 55  23  66  56  64  20  32  36  47  94  64  86   4  45   9  74  67  57  16   8
 85  90  64  73  54   8  47   8  98  39   6  47  53  73  35   4  30  94  52  32
 85  95  73  81  78   6  37  32  32  73  19   5  73  21  11   9  54  16  24  17
 71  46  92  69  57  69   2  40  91  91  98  38  74  89  82  97  28  29  16  68
 14  31  76  13  93  19  27  86  78  45  88  87  44  70  37  82  59  95  97  54
 23  56  26  44  83  43   6  14  34

Remove 83 elements to queue Q. Content of queue Q after removing elements:
Queue capacity: 100
Queue size    : 26
****Queue elements from head to tail:
 13  93  19  27  86  78  45  88  87  44  70  37  82  59  95  97  54  23  56  26
 44  83  43   6  14  34

-------------------------------------------------------------------
>>>> Trial 3:
Insert 35 elements to queue Q. Content of queue Q after inserting elements:
Queue capacity: 100
Queue size    : 61
****Queue elements from head to tail:
 13  93  19  27  86  78  45  88  87  44  70  37  82  59  95  97  54  23  56  26
 44  83  43   6  14  34  18  38  19  40  92  51  41  36  95  10  43  89  81  19
  4  58  60  59  48  33  71   6  73  38   2  88  58   4  13  86  94   3  99  79
  3

Remove 32 elements to queue Q. Content of queue Q after removing elements:
Queue capacity: 100
Queue size    : 29
****Queue elements from head to tail:
 41  36  95  10  43  89  81  19   4  58  60  59  48  33  71   6  73  38   2  88
 58   4  13  86  94   3  99  79   3

-------------------------------------------------------------------
>>>> Trial 4:
Insert 98 elements to queue Q. Content of queue Q after inserting elements:
Queue capacity: 150
Queue size    : 127
****Queue elements from head to tail:
 41  36  95  10  43  89  81  19   4  58  60  59  48  33  71   6  73  38   2  88
 58   4  13  86  94   3  99  79   3  32  66  86  12  81  61  62  97  62  64  98
 60   9  92  97  83  91  53  33  44  84   5  48   4  34  70   0  41  99  20  38
 58  88  67  58  93  17  28  48  62  81  21  97  90  80  38  90  51  32  93  36
 18  20  33  19   0  72  35  96  25  48  49   4  30   3  39  91  10  88  43   5
 36  54  40  56   7  33  98  90  93  15  20  33  52  28  98  44  46  26  97  87
 84  28  75  46   1   4  67

Remove 126 elements to queue Q. Content of queue Q after removing elements:
Queue capacity: 100
Queue size    : 1
****Queue elements from head to tail:
 67
```