1. **Write a program** including the following steps

(1) using **array implementation of trees**

(a) Use **rand()%100+1** to get 40 random numbers, **output** the numbers (one by one, one space in between, and 10 numbers in one line), **insert** the numbers into a created **array** one by one (from tree[0]~tree[39]) immediately to form a tree implemented by an array.

(b) output the data in the tree nodes one by one in *preorder*.

(c) output the data in the tree nodes one by one in *inorder*.

(d) output the data in the tree nodes one by one in *postorder*.

(e) output the data in the tree nodes one by one in *levelorder*.

(f) output the data in the tree nodes one by one in *inorder* (using iterative inorder traversal).

(2) using **linked implementation of trees**

(a) Use **rand()%100+1** to get 10 random numbers, **output** the numbers (one by one, one space in between, and 5 numbers in one line), **insert** the numbers into a created **highly skewed** tree one by one immediately.

(b) output the data in the tree nodes one by one in *preorder*.

(c) output the data in the tree nodes one by one in *inorder*.

(d) output the data in the tree nodes one by one in *postorder*.

(e) output the data in the tree nodes one by one in *levelorder*.

(f) output the data in the tree nodes one by one in *inorder* (using iterative inorder traversal).

**2.** **Write a program** including the following steps.

(1) Use **rand()%100+1** to get 40 random numbers, **output** the numbers (one by one, one space in between, and 10 numbers in one line), **insert** the odd numbers into a created **linear** list (**listPointer list1**) one by one (sorted in ascending order – **from small to big**), and **insert** the even numbers into another created **linear** list (**listPointer list2**) one by one (sorted in ascending order – **from big to small**). (Note: Be sure to use the two functions **void insertnum\_s2b(listPointer \*p, int x) and void insertnum\_b2s(listPointer \*p, int x)**.)

(2) output the linear list data from **list1** and **list2** (one by one from the first to the last, one space in between, and 10 numbers in one line). (Note: Be sure to use the function **void printL(listPointer p)**. )

(3) produce a new linear list (**listPointer p**) that contains the linear list **list1** followed by the linear list **list2**, and output data from the new linear list **list** (one by one from the first to the last, one space in between, and 10 numbers in one line). (Note: Be sure to use the function **listPointer concatenate(listPointer p1, listPointer p2)**. )

(4) invert the linear list **list** to form a new linear list **invlist** and output the linear list data from the new linear list **invlist** (one by one from the first to the last, one space in between, and 10 numbers in one line). (Note: Be sure to use the function **listPointer invert(listPointer p)**. )

(5) change the linear list **invlist** to a **circular** list. (Note: Be sure to use the function **listPointer Lin2Cir(listPointer p)**. )

(6) output the circular list data(one by one from the first to the last, one space in between, and 10 numbers in one line). (Note: Be sure to use the function **void printC(listPointer p)**. )

(7) start from the first node of the circular list in (6), count around the list and delete the 6th node each time (using **int delete(listPointer p)**), and output the deleted number until the list becomes empty.

3. (1) **Write a program** including the following steps (**implementing Linked Queues by Linked Stacks**).

(a) Use **rand()%100+1** to get 40 random numbers, output the numbers (one by one, one space in between, and 10 numbers in one line) and push the numbers into a created queue **Q** one by one.

(b) Assign and output integer i the 2nd element from the tail of **Q**, leaving **Q** unchanged. (Note: Be sure to do in the way like the shadow following the body.)

(c) Assign and output integer j the 8th element from the head of **Q**, leaving Q unchanged.

(d) Assign and output integer k the 3rd element from the tail of **Q**. (Note: Be sure to do in the way like the shadow following the body.)

(2) **Write a program** including the following steps (**implementing Linked Stacks by Linked Queues**). **(Note: Be sure to assume the stack top is the queue front.)**

(a) Use **rand()%100+1** to get 40 random numbers, output the numbers (one by one, one space in between, and 10 numbers in one line) and push the numbers into a created stack **S** one by one.

(b) Assign and output integer m the 4th element from the bottom of **S**, leaving **S** unchanged. (Note: Be sure to do in the way like the shadow following the body.)

(c) Assign and output integer n the 12th element from the top of **S**, leaving **S** unchanged.

**Note:** Please

1. put necessary **English Dev-C++ DEBUG window screenshots** to show required **Dev-C programs** and **highlighted required execution results**,
2. comment student ID+your name **in every screenshots**, and
3. put reports into one word file named by student\_ID+your\_name.