창의적 소프트웨어 설계 실습 문제 3주차

제출 기한
- 채점서버 제출 : (1차) 10월 8일 토 11:59pm 까지
(2차) 10월 15일 토 11:59pm 까지

3-1. 이진 탐색 (binary search)

앞의 SimpleIntSet 내의 원소를 빠르게 찾을 수 있는 이진 탐색 함수 작성.

- 이진 탐색 알고리즘은 아래 링크를 참조
http://en.wikipedia.org/wiki/Binary_search_algorithm
- 주어진 SimpleIntSet (simple_int_set.h , simple_int_set.cc) 클래스를 구현하고 이를 이용한다. 링크
binary_search.cc 에서 /* Implement here */ 로 된 주석 부분에 코드를 작성한다.
(binary_search_main.cc 는 입력을 받는 부분으로 특별히 수정하지 않고 써도 된다.)
- 해당 원소의 위치를 리턴한다. 주어진 값이 없으면 -1 을 리턴한다.
- 찾을 값이 -999가 입력되면 종료한다.

파일명 : binary_search.h / binary_search.cc / binary_search_main.cc
입력 : SimpleIntSet 입력 이후 탐색하고하 하는 원소 입력
출력 : 해당 탐색값의 위치

```
$ ./binary_search
{ 2 3 5 7 9 }
2
0
5
2
4
-1
-3
-1
100
-1
9
4
-999
$
```

3-2. 간단한 댓글 관리 프로그램 (Simple Reply Administrator)

댓글을 입력하고, 인덱스와 명령어를 통해 삭제하는 프로그램 작성

- 프로그램을 실행한 뒤, 문자열을 입력하여 댓글을 추가할 수 있다.
  (개발의 편의를 위해, code skeleton에 임의로 추가된 댓글이 존재한다.)
- 추가된 댓글에는 추가한 순서대로 0부터 번호를 매긴다.
- 모든 명령어는 '#' 문자로 시작한다.
- remove명령어를 통해 댓글을 삭제할 수 있다.
  - #remove [number]: 특정 번호의 댓글을 삭제한다. (e.g. #remove 3)
    번호가 존재하지 않을 경우 아무런 내용도 출력하지 않는다.
  - #remove [number][number]: 영역 내의 댓글을 모두 삭제한다. (e.g. #remove 25)
    영역이 존재하는 댓글을 초과할 경우, 영역 내 존재하는 댓글만 삭제한다.
  - #remove [number],[number], ... ,[number]: 선택된 댓글을 삭제한다.
    (e.g. #remove 0,1,3) / 존재하는 번호만 삭제한다.
  - #remove 명령어가 성공적으로 실행된 이후에는, 남아있는 댓글 목록을 출력하며 남아있는 댓글의 번호는 순서대로 다시 매겨진다.
  - 복합 명령어는 구현하지 않으며, 오류로 처리된다. (e.g. #remove 1,2-4)
  - #remove asdf 등 명령어에 부적절한 인수가 입력됐을 때는 아무 내용도 출력하지 않는다. (이 경우 일반 문자열로 취급하지 않는다)
- #quit 명령어를 입력하면 프로그램이 종료된다.
- 잘못된 명령어 (#abc 등)는 무시한다.

파일명 : reply_admin.cc
입력 : 단순 문자열 or 명령어가 포함된 문자열 (#remove, #quit)
출력 : 명령어 입력 후 변화된 댓글 목록을 출력

```
$ ./reply_admin
Hello, World!
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 I think male TA looks cool.
6 Hello, World!
#remove 5
0 Hello, Reply Administrator!
1 I will be a good programmer.
2 This class is awesome.
3 Professor Lim is wise.
4 Two TAs are kind and helpful.
5 Hello, World!
#remove 0,5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
```

```
3 Two TAs are kind and helpful.
C++ is so hard
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
Yesterday was too bad.
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
4 C++ is so hard
5 Yesterday was too bad.
#remove 4-5
0 I will be a good programmer.
1 This class is awesome.
2 Professor Lim is wise.
3 Two TAs are kind and helpful.
#quit
```

Code skeleton

```
#include <iostream>
#include <string>
#include <stdio.h>

using namespace std;

const int NUM_OF_CHAT = 200;

int getChatCount(string *_chatList)
{
  int i;
  for(i=0; i<NUM_OF_CHAT; ++i)
  {
    string s = _chatList[i];
    if(s.empty() == true) break;
  }
  return i;
}

void printChat(string *_chatList)
{
  int count = getChatCount(_chatList);
  for(int i=0; i<count; ++i)
  {
    cout << i << " " << _chatList[i] << endl;
  }
}
```

```
// Implement these functions
bool addChat(string *_chatList, string _chat); // returns true when adding chat is
succeeded
bool removeChat(string *_chatList, int _index); // returns true when removing chat
is succeeded

// Implement commented (/* */) areas in main function
int main(void)
{
  string* chats = new string[NUM_OF_CHAT];

  addChat(chats, "Hello, Reply Administrator!");
  addChat(chats, "I will be a good programmer.");
  addChat(chats, "This class is awesome.");
  addChat(chats, "Professor Lim is wise.");
  addChat(chats, "Two TAs are kind and helpful.");
  addChat(chats, "I think male TA looks cool.");

  while(true)
  {
    string command;
    getline(cin, command);

    if(/* #quit */) break;
    else if(/* #remove */)
    {
      /* remove chat */
      if(/* remove is succeeded */) printChat(chats);
    }
    else if(addChat(chats, command)) printChat(chats);
    }

    // delete chatting list
    delete[] chats;

    return 0;
}
```

3-3. 정수 집합 연산 (integer set operations)

입력받은 정수 집합에 대한 합집합, 교집합, 차집합을 계산하는 SompleIntSet class 작성

- 주어진 SimpleIntSet (simple_int_set.h , simple_int_set.cc) 클래스를 구현하고 이를 이용한다. 링크
- 메인함수와 멤버함수의 인터페이스는 변경할 수 없다.
- 집합의 문자열 크기는 1024 를 넘지 않는다고 가정한다.
- 집합 내의 숫자는 오름차순으로 정렬하여 관리한다. (hw01에서 구현한 정렬 재사용 가능)
- 한 집합 내에 동일한 원소는 존재할 수 없다. (e.g. { 1 1 2 3 } 입력시 { 1 2 3 } 으로 처리)
- 주어진 모든 public 멤버함수를 구현한다.
    - Initialize : 문자열을 받아 elements 배열에 정수 원소를 집합 형태로 저장한다.
    - GetUnion : 두 클래스 left, right 의 데이터로부터 합집합을 구한다.
    - GetDifference : 두 클래스 left, right 의 데이터로부터 차집합을 구한다.
    - GetIntersetcion : 두 클래스 left, right 의 데이터로부터 교집합을 구한다.
- 입력에 오류가 있으면 종료한다.

파일명 : simple_int_set.h / simple_int_set.cc / simple_int_set_main.cc (.cc = .cpp)
입력 : { num1 num2 … numk1 } OP { num1 num2 … numk2 }
OP = +, *, -
출력 : 연산 결과 집합을 같은 형식으로 출력

```
$ ./simple_int_set
{ 1 2 3 } + { 3 4 5 }
{ 1 2 3 4 5 }
{ -1 5 3 2 } - { 1 2 3 }
{ -1 5 }
{ -1 5 3 2 } * { 1 2 3 }
{ 2 3 }
0
$
```