Lab2: Recursion 연습

1) Lab2-1: Binary Search (이진 탐색)

a. 알고리즘은 두가지 버전으로 (Iterative version, Recursive version)

b. 입력 데이터: 프로그램에서 구성

ex) int list[] = {10,20,30,40,50,60,70,80,90,100 };

c. Search Number: 실행시 입력받음

d. 출력: return it's position if found, otherwise return –1.

e. 메뉴구성 **(1. Binary Search 2. Recursive Search 3. Exit)** 하여 **3.Exit** 선택시 까지 계속할 것.

● **알고리즘 (강의노트)**

- **iterative version(반복 알고리즘)**

```
procedure binary-search( ){
  while (left <= right)  {
    middle = (left + right)/2;
    switch (COMPARE(list[middle],  searchnum)) {
      case  -1: left = middle + 1;       break;
      case   0: return middle;
      case   1: right = middle - 1;     break;
    }
    return -1;   }
```

- **Recursive Version (순환알고리즘)**

```
procedure binsearch ( int list[],   int searchnum,   int left,   int right) {
  if (left <= right)   {
    middle = (left + right) / 2;
    switch (COMPARE(list[middle], searchnum)) {
      case   -1: return   binsearch(list, searchnum,   middle+1, right);
      case    0: return middle;
      case    1: return    binsearch(list, searchnum,   left,   middle-1);
    }
  }
  return -1;   }
```

(출력 예)

Numbers in the List : 10 20 30 40 50 60 70 80 90 100

Enter methods of Search : 1. Bin-search 2. Recursive search 3. Exit **:** **1**
Enter an integer to search :10
⇨ **10 is at position 1**

Enter methods of Search : 1. Bin-search 2. Recursive search 3. Exit : **2**
Enter an integer to search :20
⇨ **20 is at position 2**

Enter methods of Search : 1. Bin-search 2. Recursive search 3. Exit : **1**
Enter an integer to search :22
⇨ **22 is not in the list**

Enter methods of Search : 1. Bin-search 2. Recursive search 3. Exit : **3**

2) Lab2-2: Compute Area of the Figures (다면체 면적 구하기)

- **Triangle, Rectangle, Circle 의 각 제원을 데이터화일에서 받아서 면적을 계산하여, 출력한다.**
- **각 다면체는 class 또는 struct로 구성**

```
class Angle {
private:
        int type;
public:
        int width;
        int height;
        int round;
        Angle(int inputType) : type(inputType) { width = height = round = 0; }
        double calculate() {
                switch (type) {
                case 1: …..   //triangle
                case 2: ….   //rectangle
                case 3: ….   //circle
                }
        }
};
```

```
struct Angle {
        string fig;
        int width;
        int height;
        int round;
        double area;
};
Angle angle[7];
```

Input Data file: (7개 데이터)

triangle 20 40
rectangle 40 80
circle 50
rectangle 30 60
triangle 10 20
triangle 50 30
circle 20

OutPut:

triangle 20.00 40.00 400.00
rectangle 40.00 80.00 3200.00
circle 50.00 7850.00
rectangle 30.00 60.00 1800.00
triangle 10.00 20.00 100.00
triangle 50.00 30.00 750.00
circle 20.00 1256.00

Press any key to continue

**** 주의사항**

1) Ecampus에 제출시 (1. 소스화일, 2. 출력물 스크린 샷) 제출할것

2) 시간 연장하여, 11시까지 제출완료할 것. (미제출시, 1주후까지 부분점수)