

# Tech University of Korea(TUK)

## 자료구조 과제 노트

202#-0#학기

담당교수	박정민
학번	•
이름	• 윤##

## 과제수행 요청서

### 과제기간

#### ■ (과제기간) 과제는 교재의 각 장이 끝난 일주일 후, 24시까지 제출

- (제출방법) e-class의 ▲과제게시판에 업로드
- (제출파일명) [학번이름].hwp 예시) 01[2017131023박정민].hwp
- (지각제출불가) 과제를 제출하는 기간에 반드시 제출, ▲지각제출불가
- (개인제출) 과제제출은 ▲개인적으로 수행

### 과제내용

#### ■ (과제내용) 과제의 내용은 총 4가지 1)요점정리, 2)역공부, 3)순공부, 4)자기성찰

- (요점정리) 수업시간의 내용을 ▲재정리 ▲ 개별적으로 공부한 것이 있다면 추가 요점정리
- (역공부) 코드를 분석한 내용정리 ▲손으로 분석한 그림, ▲디버깅 SW를 이용한 분석그림
- (순공부) 역공학을 통해 분석한 소스코드를 근거로 ▲코드를 수정/개선시켜보기, ▲주석 상세히 달기
- (자기성찰) ▲수업을 통해서 배운 것, ▲디버깅을 통해 집중적으로 공부한 것

### 결과물활용

#### ■ (결과물) 결과물은 1)수시고사, 2)과제평가, 3) 개인 정리를 위한 지침

- (수시고사를 위한 참조) 과제들은 수시고사와 연계 ▲오픈북 시에 참조 문서로 활용
- (과제평가) 각 장별로 자료구조 수업이 종강될 때까지 작성하여 매주 평가
- (개인정리를 위한 지침) 과제내용 스스로 잘 정리하기 위한 좋은 가이드라인으로 활용

## 자료구조 과제 목차

### 1. 제1장 자료구조와 알고리즘

1-1 1장 자료구조와 알고리즘 요점정리

1-2 소스코드 디버깅 분석(역공부)

1-3 소스코드 수정/개선 & 주석(순공부)

### 2. 자기성찰

2-1 평가내용 및 느낀 점 (총 50점)

## 1. 제1장 자료구조와 알고리즘

### 1-1 1장 자료구조와 알고리즘 요점정리

#### ■ 자료구조

- 스택
- 큐
- 리스트
- 사전
- 그래프
- 트리

#### ■ 알고리즘의 기술방법

- 자연어
- 순서도(흐름도)
- 유사코드
- 기계어

#### ■ 추상 데이터 타입

- 정보은닉기법

#### ■ 알고리즘의 성능 분석 기법 (통틀어 시간 복잡도)

- 직접실행 : time, 수행 시간 측정
- 간접실행 : 연산의 횟수, 알고리즘의 복잡도 분석, 빅오(약)

#### ■ 자료의 개수 n개 똑같이

- 동일한 하드웨어를 사용

#### ■ 빅오 표기법

- 자료의 개수가 많은 경우 차수가 큰 것이 영향이 크다.

#### ■ 알고리즘 성능 평가 결과

- 최선 : 수행시간이 가장 빠름
- 평균
- 최악 : 최악의 경우를 가지고 실험

### 1-2 소스코드 디버깅 분석(역공부)

#### ■ '순' 분석내용

- calc\_scores.c

#define MAX\_ELEMENTS 3

int scores[3] N 전역배열  
MAX\_ELEMENTS

DATE NO

&scores[0]

30	scores[0]	① 입력: n 입력, 학생들 수
20	scores[1]	② 처리: 가장 큰 점수 처리
40	scores[2]	③ 출력: 점수 출력 (큰 점수)

int get\_max\_score(int n)

largest	30	largest	30		40	40
i		i	1		2	3

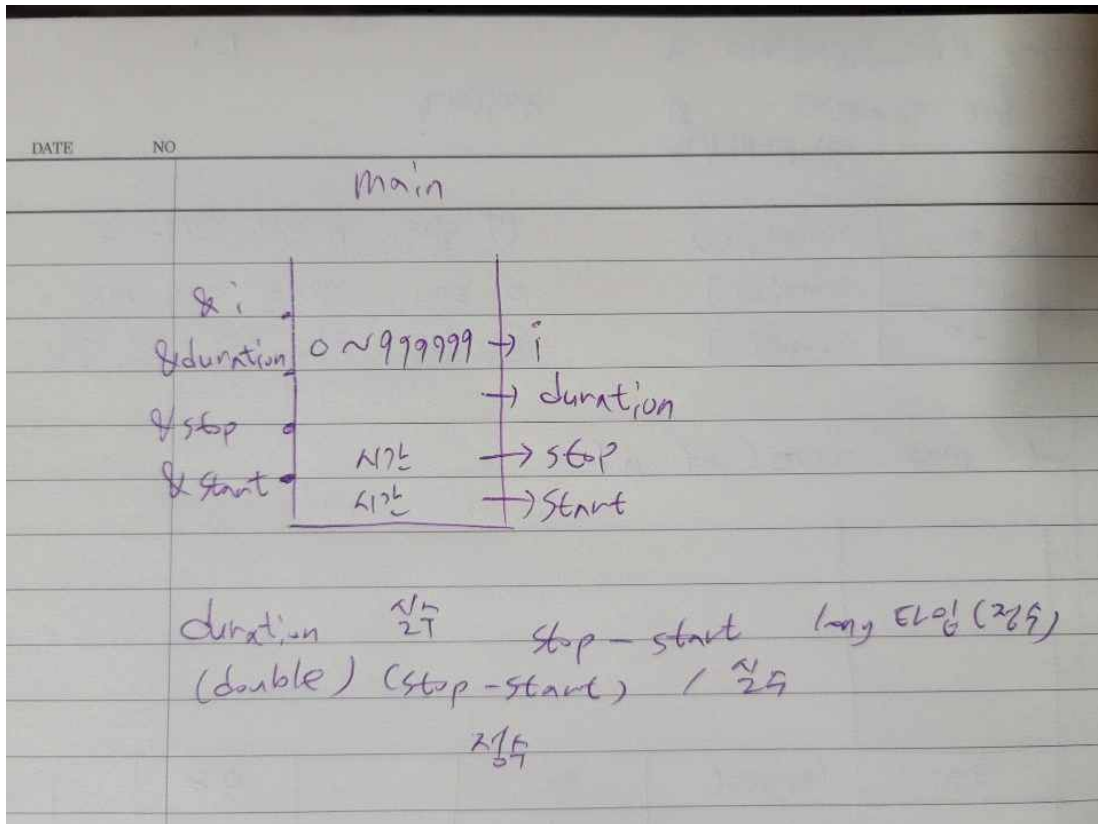
N=3 largest = scores[0]

return largest

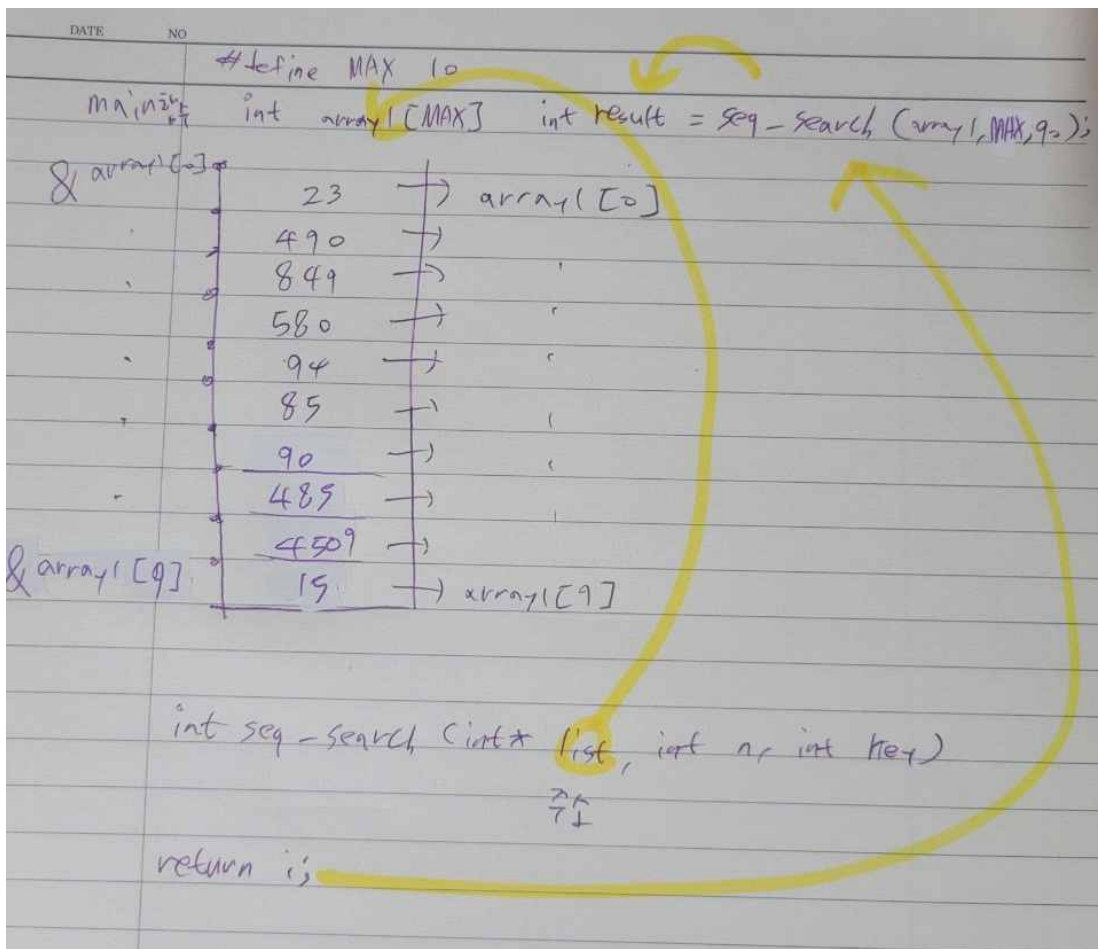
40

result = get\_max\_score(MAX\_ELEMENTS);

- calc\_time.c



- seq\_search.c



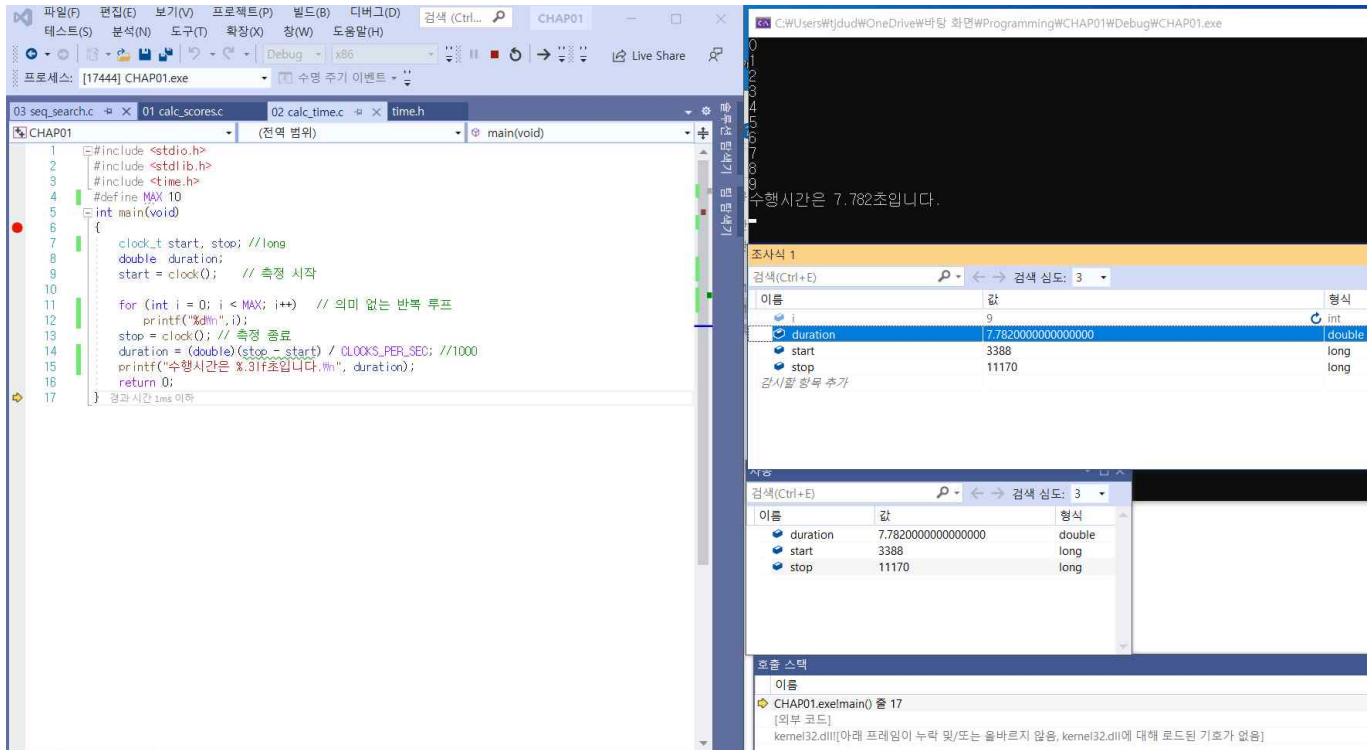
## ■ 디버깅SW를 이용한 분석내용

- calc\_scores.c
- 호출 스택에 get\_max\_score 살아있다가 다시 메인함수로 돌아가면 메모리에서 사라짐
- 따라서 호출 스택에서도 사라짐

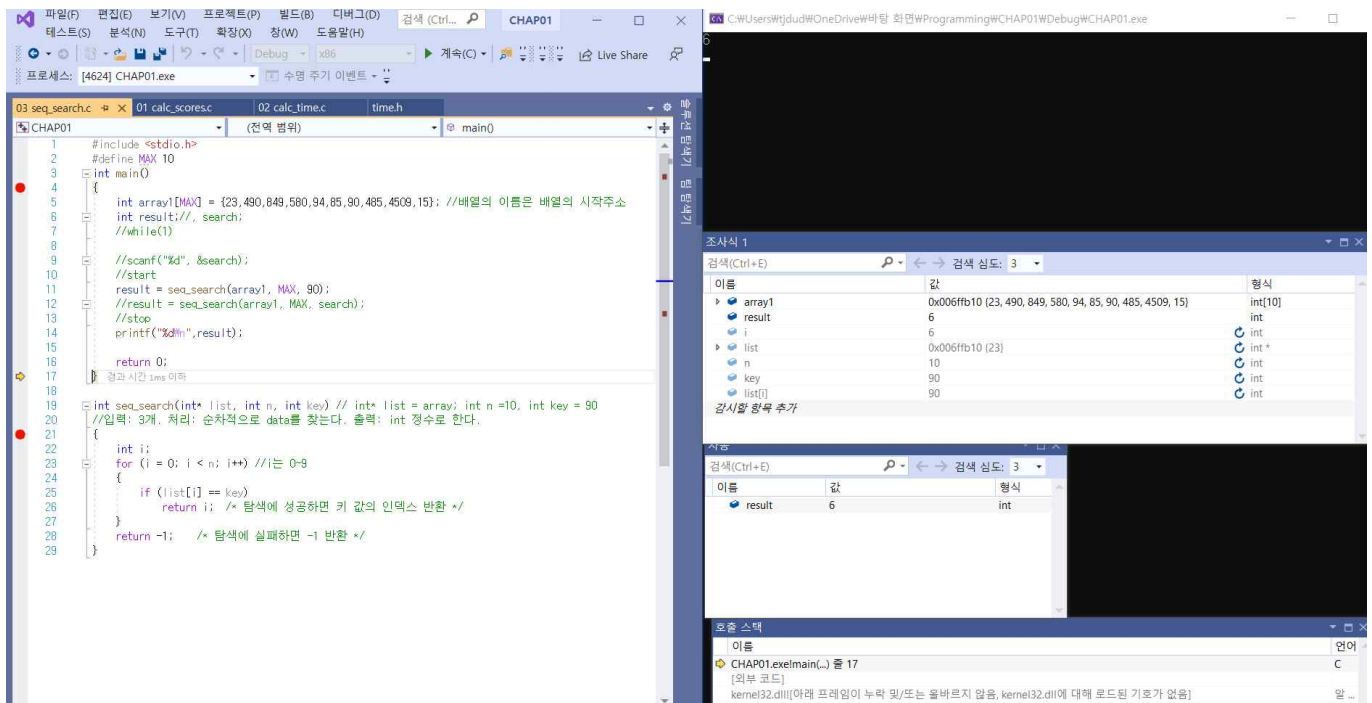
The screenshot displays a debugger interface with the following components:

- Source Code Window:** Shows the implementation of `get_max_score` and `main` in `calc_scores.c`. The `main` function calls `get_max_score` with `MAX_ELEMENTS` as an argument.
- Call Stack Window:** Shows the current call stack with `CHAP01.exe!get_max_score(int n) 줄 33` as the active function.
- Memory Window:** Shows the state of the `scores` array, with values 30, 20, and 40 at indices 0, 1, and 2 respectively.
- Registers Window:** Shows the values of the variables `i`, `largest`, and `scores[i]`.
- Variables Window:** Shows the values of the variables `scores[0]`, `scores[1]`, and `scores[2]`.
- Registers Window:** Shows the values of the variables `i`, `largest`, and `scores[i]`.
- Variables Window:** Shows the values of the variables `scores[0]`, `scores[1]`, and `scores[2]`.

- calc\_time.c



### • seq\_search.c



## 1-3 소스코드 수정/개선 & 주석(순공부)

### ■ 수정/개선 & 주석1 (수정/개선된 코드에 주석을 달고 & 설명)

- calc\_scores.c 코드
- 키보드를 통해 점수를 받았습니다.



- 키보드를 통해 받을 때 for문으로 scanf("%d", &scores[i]);를 한꺼번에 처리했습니다.
- start, stop 코드로 키보드로 입력받을 때 시간을 측정했습니다.
- system("pause");를 통해 콘솔 창이 바로 꺼지지 않고 멈춰있게 했습니다.
- 속성\c/c++\전처리기\전처리기 정의에 \_CRT\_SECURE\_NO\_WARNINGS를 입력하여 scanf\_s를 쓰지 않을 때 오류를 해결했습니다.

```
#include <stdio.h>
#include<stdlib.h>
#include <time.h>
#define MAX_ELEMENTS 3
int scores[MAX_ELEMENTS]; // 자료구조, 전역배열

int get_max_score(int n); // 함수의 선언, 함수의 리스트

int main(void)
{
    clock_t start, stop; //long
    double duration;
    int result;
    start = clock(); // 측정 시작
    for (int i = 0; i < MAX_ELEMENTS; i++) //for문으로 점수 받기
    {
        scanf("%d", &scores[i]);
    }
    //에니악 대포 탄도 계산(시간측정중요하다는 예)
    //시간 측정

    result = get_max_score(MAX_ELEMENTS);
    stop = clock();

    printf("결과:%d\n ",result);
    duration = (double)(stop - start) / CLOCKS_PER_SEC; //1000
    printf("수행시간은 %.3lf초입니다.\n", duration);
    system("pause");
    return 0;
}

int get_max_score(int n) // 학생의 숫자는 n
{
    int i, largest;
    largest = scores[0]; // 알고리즘
    for (i = 1; i<n; i++)
    {
        if (scores[i] > largest)
        {

```

```

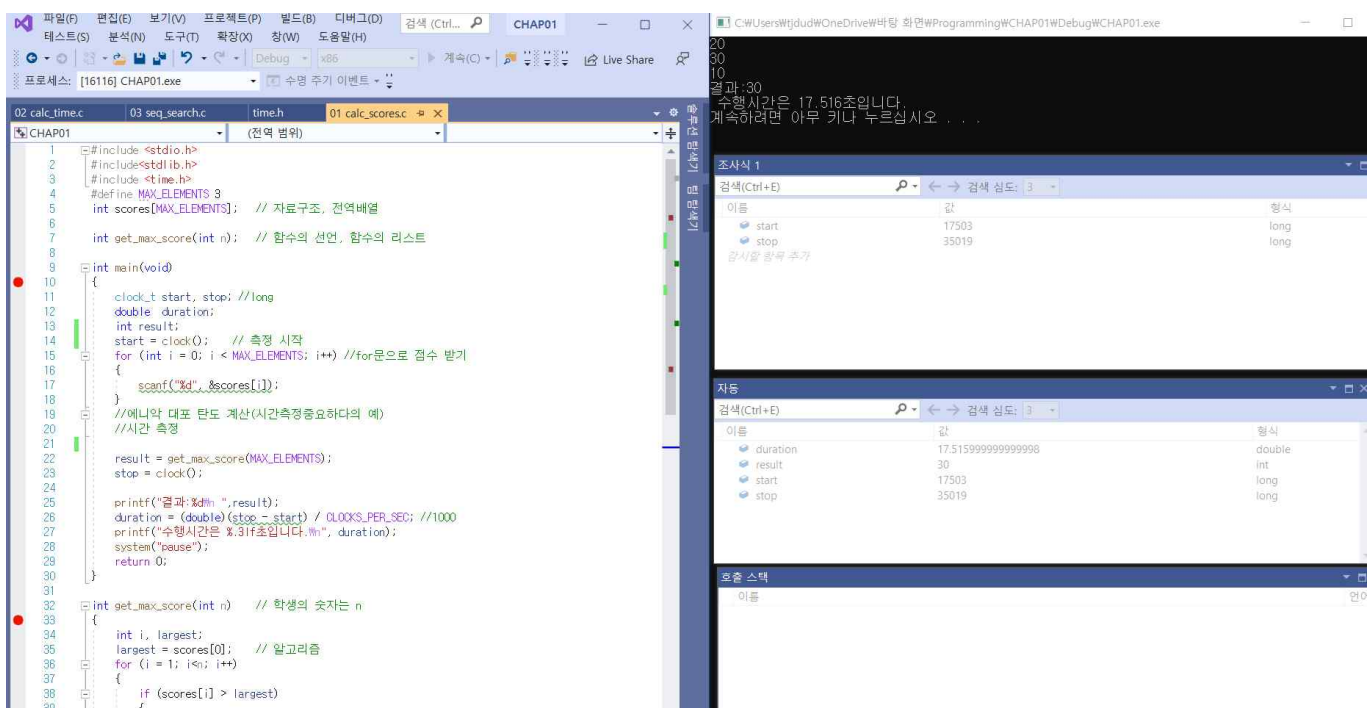
40         largest = scores[i];
41     }
42 }
43 return largest;
44 }// 입력: n, 처리: 큰점수 획득, 출력: int(정수)

```

```

20
30
010
1 결과:30
2 수행시간은 4.447초입니다.
3 계속하려면 아무 키나 누르십시오 . . .
4
5

```



## ■ 수정/개선 & 주석2

- seq\_search.c 코드
- #define \_CRT\_SECURE\_NO\_WARNINGS 로 scanf\_s()를 사용하지 않아 생기는 문제 해결
- int seq\_search(int\* list, int n, int key); 함수의 선언부
- 키보드를 통해 key를 받기 위해 scanf함수를 사용
- 배열에 있는 값을 보여주기 위해 while 무한 반복문으로 0~9까지 보여주고 10이 될 때 break를 써서 빠져나왔습니다.
- calc\_time.c에서 배운 코드를 토대로 start, stop 코드를 집어넣어 시간을 측정하였습니다.

```

#define _CRT_SECURE_NO_WARNINGS // scanf_s()를 사용하지 않아 생기는 문제 해결

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 10

int seq_search(int* list, int n, int key); //함수의 선언부

int main()
{
    clock_t start, stop;
    double duration;
    int array1[MAX] = {23,490,849,580,94,85,90,485,4509,15}; //배열의 이름은 배열의 시작주소
    int result, search; //키보드로 key 받기
    int k = 0;

    //측정 시작
    /*
    while (1) //무한반복문으로 배열에 있는 값 보여주기
    {
        if (k == 10)
            break;

        printf("%d\n", array1[k]);
        k++;
    }
    */

    scanf("%d", &search);
    start = clock();
    result = seq_search(array1, MAX, search); //키보드로 받은 search
    stop = clock(); // 측정 종료
    printf("%d\n", result);

    duration = (double)(stop - start) / CLOCKS_PER_SEC; //1000
    printf("수행시간은 %.3lf초입니다.\n", duration);
}

```

```

    return 0;
}

int seq_search(int* list, int n, int key) // int* list = array; int n =10, int key = 90
//입력: 3개, 처리: 순차적으로 data를 찾는다. 출력: int 정수로 한다.
{
    int i;
    for (i = 0; i < n; i++) //i는 0~9
    {
        if (list[i] == key)
            return i; /* 탐색에 성공하면 키 값의 인덱스 반환 */
    }
    return -1; /* 탐색에 실패하면 -1 반환 */
}

```

- start를 scanf()함수 밑에 두고 디버깅 했을 때 사진
- 키를 찾는 시간 측정(디버깅 할때는 f10을 누르며 느리게 찾아서 초가 나오지만 디버깅 하지 않고 실행 했을 때는 컴퓨터가 빨라서 순식간에 찾기 때문에 0초가 나왔습니다.)

90  
6  
수행시간은 9.266초입니다.

조사식 1

이름	값	형식
start	13314	long
stop	22580	long

감시할 항목 추가

자동

이름	값	형식
duration	9.2660000000000000	double

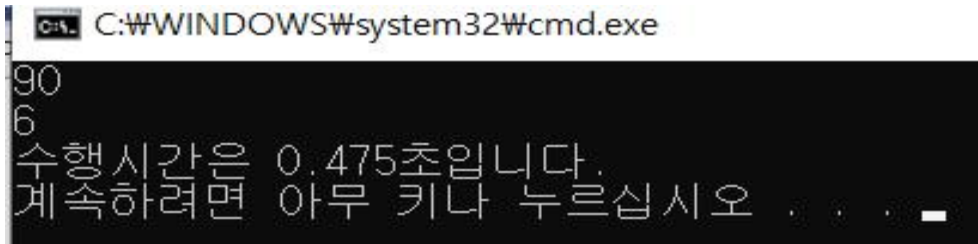
호출 스택

이름	언어
CHAP01.exe!main(...) 줄 40	C
[외부 코드]	
kernel32.dll![] [아래 프레임이 누락 및/또는 올바르지 않음, kernel32.dll에 대해 로드된 기호가 없음]	알 ...

C:\WINDOWS\system32\cmd.exe

90  
6  
수행시간은 0.000초입니다.  
계속하려면 아무 키나 누르십시오 . . .

- start를 scanf()함수 위에 두고 디버깅 했을 때 사진
- scanf()함수의 입력시간 측정



```
C:\WINDOWS\system32\cmd.exe
90
6
수행시간은 0.475초입니다.
계속하려면 아무 키나 누르십시오...
```

## 2. 자기성찰

### 2-1 평가내용 및 느낀 점 (총 50점)

평가 내용 (30점 만점)	3점	2점	1점
1. 온라인강의는 주어진 기한 내 듣기 완료를 하였나?	●		
2. 이번 강의 수업은 이해도가 높았나?	●		
3. 역공부, 주어진 소스코드 분석을 위해 손으로 그림을 그려보았나?	●		
4. 역공부, 주어진 소스코드 분석을 위해 디버깅 SW로 분석해보았나?	●		
5. 역공부, 소스코드를 구현한 개발자의 노력을 생각해보았나?	●		
6. 순공부, 3번과 4번을 수행하면서 코드를 개선시킬 노력은 하였나?	●		
7. 순공부, 개선코드 주석은 꼼꼼히 작성하였나?	●		
8. 과제 '양식(hwp)'을 준수하여 작성하였나?	●		
9. 오타 없이 표의 양식, 그림의 넓이&높이 등 정돈된 보고서인가?	●		
10. 시간 마감에 급급한 대충 작성한 보고서가 아닌, 열심히 공부한 정직한 보고서인가?	●		
총 점	30 점		

‘객관식 30점, 주관식 20점, 총 50점을 스스로 자기성찰’ 하고, 정직한 보고서를 위해 최선을 다하자!

노력지수 (10점 만점)	<ul style="list-style-type: none"> <li>• 9점 <ul style="list-style-type: none"> <li>- 이미 굳어진 필기체 때문에 손 분석 할 때 글씨체가 예쁘지 않아서 1점을 감점하였습니다. 그래도 최대한 깔끔하게 쓰기 위해 노력하였고 포인터가 주소를 가리키는 부분은 색깔펜으로 처리했습니다. 또한 사진은 넓이를 최대한 맞추려고 노력했습니다. 손 분석과 sw 디버깅을 통해 이번 강의 코드를 완벽 이해했습니다.</li> </ul> </li> </ul>
느낀 점 (10점 만점)	<ul style="list-style-type: none"> <li>• 10점 <ul style="list-style-type: none"> <li>- 수정/개선 작업을 통해 시간 측정을 했을 때 0.000초가 나와서 “원래 그런 것인가?” 라고 넘어가려다가 혹시 몰라, 질의응답 란에 문의했습니다. 답변을 통해 코드를 다시 분석해볼 수 있었습니다. 찾는 숫자도 앞에 있고 MAX값이 작아서 그럴 수 있다는 것을 깨닫게 되었습니다.</li> <li>- start가 scanf 위에 있느냐 아래에 있느냐를 통해 위에 있으면 scanf에 입력한 시간이 측정된다는 것을 알게 되었습니다. start가 아래에 있을 때 디버깅을 직접하면 f10으로 일일이 측정하기 때문에 시간이 측정된다는 것을 발견하였고 디버깅을 하지 않고 실행하면 컴퓨터가 빠르기 때문에 0초만에 결과가 나온다는 것을 알게 되었습니다. 데이터의 규모가 작고, 찾고자 하는 데이터가 앞에 있으면 순식간에 찾을 수 있다는 것을 느끼게 되었습니다. 이번 과제를 하면서 컴퓨터가 엄청 빠르다는 것을 느끼게 되었습니다.</li> </ul> </li> </ul>