

# 프로그래밍 과제 01

1.  $n \leq 100$ 개의 정수를 입력받아 최소값, 최대값, 평균, 그리고 표준편차를 계산하여 출력하는 프로그램을 작성하라. 단, 포인터가 아닌 변수를 하나라도 사용해서는 안된다. 입력 형식은 다음과 같다. 먼저 정수  $n$ 이 주어지고, 이어서  $n$ 개의 정수들이 주어진다.

입력의 예	출력
5	
1 2 3 4 5	1 5 3 1.41421
10	
9 -6 2 7 11 10 7 -11 8 7	-11 11 4.4 6.93109
20	
7 18 3 10 -16 8 -19 3 8 1 -9 -12 34 8 11 7 9 13 0 -1	-19 34 4.15 11.7443

2. 평면상의  $n$ 개의 점의 좌표가 주어진다. 이 점들 중에서 서로 가장 멀리 떨어져있는 한 쌍의 점을 찾아서 두 점의 좌표와 두 점 간의 거리를 출력하는 프로그램을 작성하라. 각 점의  $x, y$ 좌표는 모두 정수이며, 하나의 점을 저장하는 구조체 `struct Point2D`를 정의하여 사용하라. 동적 생성된 배열만을 사용해야 하며, 배열의 원소를 액세스할 때 배열 인덱스 연산자 `[]`를 사용해서는 안된다. 또한 처음에 배열을 동적 생성할 때는 길이를 4로 하고, 필요할 때만 배열 재할당을 하여 길이를 늘려라. 불필요한 배열을 삭제하여 쓰레기(garbage)가 생성되지 않게 하라. 입력은 `input2.txt` 파일로부터 받는다. 파일의 첫 줄에는 점의 개수  $n \leq 1000$ 이 주어지고, 이어진  $n$ 줄에는 한 줄에 하나의 점의  $x$ 좌표와  $y$ 좌표가 주어진다.

입력 파일의 예	출력
20	
1 3	
17 4	
-8 9	
-12 11	
9 8	
7 13	
0 8	
-4 -11	
12 -8	-12 11
3 -8	3 -19
-11 -9	33.541
5 7	
13 1	
-9 7	
8 -12	
9 9	
5 6	
3 -19	
12 10	
8 -6	

3. 아래의 프로그램은 C언어에서 사용자로부터  $n$ 개의 문자열을 입력받아 `char *` 타입의 배열 `words`에 저장한 후 출력하는 것이다. 방법 1~3이 잘못된 이유를 각각 정확히 설명하고 올바른 방법을 제시하라.

```
#include <stdio.h>
#include <string.h>
#define MAXWORDS 100
#define MAXLEN 30
```

```

int main() {
    char *words[MAXWORDS];
    int n;
    scanf("%d", &n);

    char buf[MAXLEN];
    for (int i=0; i<n; i++) {
        // 방법1:
        scanf("%s", words[i]);

        // 방법2:
        scanf("%s", buf);
        words[i] = buf;

        // 방법3:
        scanf("%s", buf);
        strcpy(words[i], buf);
    }

    for (int i=0; i<n; i++)
        printf("%s\n", words[i]);

    return 0;
}

```

텍스트

4. 텍스트 파일 input4.txt에 등장하는 길이가 2이상인 모든 단어들의 목록을 **중복없이** 만들어서 한 줄에 하나씩 출력한다. 출력 순서는 파일에 단어가 등장하는 순서이다. 총 단어수는 1000개 이하이고, 각 단어의 길이는 20이하이다. 쉼표, 마침표 등의 기호와 숫자도 단어의 일부로 간주한다.

입력 파일의 예	출력
링크된 input4.txt	AI
	juggernaut
	Nvidia
	(NVDA)
	reported
	second
	quarter
	earnings
	...
	as
	much
	6%
	immediate
	reaction
	to
	numbers.

5. 텍스트 파일 input4.txt에 등장하는 길이가 2이상인 모든 단어들의 중복 없는 목록과 각 단어의 등장 빈도를 계산하여 사전식 순서로 정렬하여 출력하는 프로그램을 작성하라. 대소문자 구분을 하며 모든 대문자는 소문자보다 사전식 순서로 앞선다. 단, 각 단어의 앞뒤에 붙은 특수문자(쉼표, 마침표, 괄호 등)와 숫자는 제거해야 한다. 총 단어수는 1000개 이하이고, 각 단어의 길이는 20이하이다. 쉼표, 마침표 등의 기호도 단어의 일부로 간주한다.

입력 파일의 예	출력
링크된 input4.txt	AI 1 Analysts 2 EPS 1 NVDA 1 Nvidia 2 Shares 1 That 1 The 2 Wednesday 1 about 1 ... stock 1 that 1 the 8 third 1 to 1 top 2 trading 1 were 3 while 1 year 2

6. 프로그램을 실행하면 화면에 프롬프트(\$)와 한 칸의 공백문자를 출력하고 사용자의 입력을 기다린다. 사용자가 공백이 포함된 문장을 입력하고 엔터(↵) 키를 치면 입력된 문장 전체(공백 문자를 포함)를 그대로 출력하고 콜론(:) 문자를 출력한 후 입력된 문장의 길이(공백 문자를 포함한 문자 개수)를 출력한다. 다시 콜론을 출력하고 문장을 구성하는 단어의 개수를 출력한다. 이 일을 사용자가 “exit”이라고 입력할 때까지 반복한다.

입출력 예
<pre>\$ hello↵ hello:5:1 \$ welcome to the class↵ welcome to the class:24:4 \$ programming is fun, right? ↵ programming is fun, right? :37:4 \$ exit</pre>

7. 다음과 같은 방식으로 2에서  $n \leq 10000$  사이의 모든 소수(prime number)를 구하는 프로그램을 작성하라. 먼저 양의 정수  $n$ 을 입력 받은 후 하나의 정수 배열에 2에서  $n$ 까지의 양의 정수들을 순서대로 저장한다. 그런 다음 배열에 저장된 각각의 정수들을 순서대로 보면서 그 정수의 배수들을 배열에서 삭제한다. 구체적으로 말하면 먼저 첫 번째 정수인 2에 대해서 2 이후에 저장된 모든 2의 배수들을 제거한다. 물론 2는 제거하지 않아야 한다. 다음은 3 이후의 모든 3의 배수들을 제거한다. 4는 이미 제거되었을 것이므로 다음은 5 이후에 등장하는 모든 5의 배수들을 삭제한다. 6은 이미 제거되었을 것이므로 다음은 7 차례이다. 정수를 삭제할 때 그 자리를 빈 자리로 남겨둬서는 안되며 뒤에 저장된 모든 정수들을 앞으로 한 칸 옮겨서 남아있는 정수들이 항상 배열에 연속된 위치에 저장되도록 해야한다. 이런 식으로 모든 배수들을 제거하면 소수들만 남아있게 될 것이다. 남아있는 모든 소수들을 출력한다. C++ 벡터(vector) 등의 컨테이너를 사용해서는 안된다.