

Data Structure1 Report



과목명		자료구조1
담당교수		홍 민
학과		컴퓨터소프트웨어공학과
학년		2학년
학번		20194059
이름		김태완
제출일		2020.04.02

목차

1. 파일입출력을 통한 최대, 최소값 출력

1.1 문제 분석

1.2 소스 코드

1.3 소스 코드 분석

1.4 실행창

1.5 느낀점

2. 느낀점

1. 파일입출력을 통한 최대, 최소값 출력

기출


01 과제

- 제출일: 4월 2일 (목)
 - 레포트 형식으로 LMS에 제출하고
 - 소스 코드는 가상강의 사이트에 올릴것
 - 20060011_홍길동_HW1.zip 으로 압축해서 제출
- 파일 입출력을 통하여 **data.txt**로부터 정수를 읽어오고 읽어온 정수를 출력하고 몇 개의 데이터씩 출력할지를 입력받아 형식에 맞춰 최소값과 최대값을 화면에 표시하시오.
 - 배열을 이용하여 메모리를 할당 할 것
 - data.txt** 파일에 정수의 데이터가 저장되어 있음
 - 저장되어 있는 정수의 수는 50개 이상 10,000개 이하
 - 다음페이지의 실행창과 같이 출력하시오

SCH 순천향대학교

동소거 학생

20194034 강윤경



Min Hong

20184022 이승현

김태완

1.1 문제 분석

이 문제는 data.txt 파일 안의 50 ~ 10000 개 사이의 정수를 읽어 한 줄에 사용자가 입력한 수 만큼 출력하고 그 아래 줄에는 바로 위에 있는 정수가 최대값인 경우에는 "최대값"을 출력 해 주고, 최소값일 경우에는 "최소값을" 최대값과 최소값 모두 아닌 경우 공백을 출력 하는 프로그램이다.

주의해야 할 점은 data.txt 에 저장된 정수 중 최대값과 최소값이 여러 개 있을 경우도 있다는 것이다. 이러한 경우 최대값과 최소값에 해당되는 정수에는 모두 "최대값" 혹은 "최소값" 을 출력 하여야 한다는 점이다.

- 3 -

2020. 3. 27.

1.2 소스코드

```
1  /*****
2  작성자 : 20194059 김태완
3  작성일 : 2020.03.26 (TUE)
4  프로그램 명 : 파일입출력을 이용한 최대,최소값 출력
5  *****/
6
7  #include <stdio.h>
8  #include <stdlib.h>
9  #define _CRT_SECURE_NO_WARNINGS
10
11 int main()
12 {
13     FILE *fp = NULL;
14     int input,i=0,j=0,line=0,count=0,max=0,min=0,end=0;
15     //input : 입력값(데이터 출력 개수)을 저장 하기 위한 변수
16     //i,j : 반복문을 위한 변수
17     //line : 출력해야 할 줄의 개수를 계산한 결과를 저장할 변수
18     //count : data.txt에서 불러온 데이터의 개수를 세는 변수
19     //max,min : 최대, 최소값을 저장 하기 위한 변수
20     //end : 출력을 중단하기 위해 사용한 변수
21     int arr[10000];
22     fp = fopen("data.txt","r"); //data.txt 파일 오픈
23     if(fp == NULL)
24     {
25         printf("파일을 열 수 없습니다."); //파일을 열 수 없을 시 오류 출력 후 프로그램 종료
26         return 0;
27     }
28     while(!feof(fp))
29     {
30         if(count>10000) //데이터의 개수가 10000개 이상일시 오류 출력 후 프로그램 종료
31         {
32             printf("데이터가 너무 많습니다.\n");
33             return 0;
34         }
35         fscanf(fp,"%d",&arr[count]); //data.txt에서 불러온 데이터를 배열 arr에 저장 후
36         count++; //데이터의 개수 카운트
37     }
38     if(count<50) //데이터의 개수가 50개 미만일시 오류 출력 후 프로그램 종료
39     {
40         printf("데이터가 너무 적습니다.\n");
41         return 0;
42     }
43     max = arr[0]; //max변수에 초기값 대입
44     min = arr[0]; //min변수에 초기값 대입
45     for(i=0; i<count; i++)
46     {
47         if(arr[i]>max) max = arr[i]; //최대값 찾기
48         if(arr[i]<min) min = arr[i]; //최소값 찾기
49     }
50     printf("몇개의 데이터씩 출력할지를 선택해 주세요: ");
51     scanf("%d",&input); //출력할 데이터의 개수를 입력받는다
52     printf("\n");
53     for(i=0; i<input; i++)
54     {
55         printf("%5d",i+1);
56     }
57     printf("\n");
58     for(i=0; i<input*5+1; i++)
59     {
60         printf("=");
61     }
62     printf("\n"); //초기 문장 출력
```

```

64 line = count/input+1; //출력해야 할 줄의 총 개수 == 총 데이터개수 / 한줄에 출력할 데이터의 개수 + 1
65 for(i=0; i<line; i++) //위에서 계산한 줄 수 만큼 반복
66 {
67     for(j=0; j<input; j++) //한 줄에 입력한 수 만큼 반복하여 출력
68     {
69         printf("%5d", arr[(i*input)+j]); //i번째 줄은 (i*input)+1 번 인덱스부터 출력되어야 한다
70         end++; //end변수가 count의 값과 같아지면 종료시키기 위해 end의 값을 1씩 증가시킨다
71         if(end==count) break; //end == count일 시 출력 종료
72     }
73     printf("\n");
74     for(j=0; j<input; j++)
75     {
76         if(arr[i*input+j]==max) printf("최대값"); //왼줄에서 출력한 값이 최대값과 같으면 "최대값" 출력
77         else if(arr[i*input+j]==min) printf("최소값"); //왼줄에서 출력한 값이 최소값과 같으면 "최소값"
78         else printf(" "); //최대, 최소 모두 아닐시 공백 출력
79     }
80     printf("\n");
81     for(j=0; j<input+5+1; j++)
82     {
83         printf("-"); //밑의 -----를 input의 수와 필드폭에 맞춰 출력
84     }
85     printf("\n");
86 }
87 printf("\n");
88 fclose(fp); //파일포인터 닫기
89 return 0;
90 }

```

1.3 소스코드 분석

```

1 1/*****
2 작성자 : 20194059 김태완
3 작성일 : 2020.03.26 (TUE)
4 프로그램 명 : 파일입출력을 이용한 최대, 최소값 출력
5 *****/
6
7 #include <stdio.h>
8 #include <stdlib.h>
9 #define _CRT_SECURE_NO_WARNINGS
10

```

1. 소스코드 작성 날짜, 작성자, 프로그램 이름을 포함한 주석을 단다.
2. 프로그램에 필요한 헤더 파일을 포함한다.

```

11 int main()
12 {
13     FILE *fp = NULL;
14     int input, i=0, j=0, line=0, count=0, max=0, min=0, end=0;
15     //input : 입력값(데이터 출력 개수)을 저장 하기 위한 변수
16     //i, j : 반복문을 위한 변수
17     //line : 출력해야 할 줄의 개수를 계산한 결과를 저장할 변수
18     //count : data.txt에서 불러온 데이터의 개수를 세는 변수
19     //max, min : 최대, 최소값을 저장 하기 위한 변수
20     //end : 출력을 중단하기 위해 사용한 변수
21     int arr[10000];

```

3. 메인 함수를 열고 소스코드 작성에 필요한 변수들을 선언 후 초기화가 필요한 변수들을 초기화 해준다.

```

22 | fp = fopen("data.txt", "r"); //data.txt 파일 오픈
23 | if(fp == NULL)
24 | {
25 |     printf("파일을 열 수 없습니다."); //파일을 열 수 없을 시 오류 출력 후 프로그램 종료
26 |     return 0;
27 | }

```

4. 파일 포인터 fp 를 이용해 data.txt 를 읽기 모드로 열고 파일이 제대로 열리지 않을 시 파일을 열 수 없다는 경고문을 띄운 뒤 프로그램을 종료한다.

```

28 | while(!feof(fp))
29 | {
30 |     if(count>10000) //데이터의 개수가 10000개 이상일시 오류 출력 후 프로그램 종료
31 |     {
32 |         printf("데이터가 너무 많습니다.\n");
33 |         return 0;
34 |     }
35 |     fscanf(fp, "%d", &arr[count]); //data.txt에서 불러온 데이터를 배열 arr에 저장 후
36 |     count++; //데이터의 개수 카운트
37 | }
38 | if(count<50) //데이터의 개수가 50개 미만일시 오류 출력 후 프로그램 종료
39 | {
40 |     printf("데이터가 너무 적습니다.\n");
41 |     return 0;
42 | }

```

5. fscanf() 함수를 통해 파일이 끝날 때 까지 정수를 읽어 arr[] 배열에 저장한다. 이때 데이터의 개수가 10000 개를 초과하거나 50 개 미만일 경우 데이터가 너무 많습니다. 혹은 데이터가 너무 적습니다. 라는 안내문을 띄우고 프로그램을 종료한다.

```

43 | max = arr[0]; //max변수에 초기값 대입
44 | min = arr[0]; //min변수에 초기값 대입
45 | for(i=0; i<count; i++)
46 | {
47 |     if(arr[i]>max) max = arr[i]; //최대값 찾기
48 |     if(arr[i]<min) min = arr[i]; //최소값 찾기
49 | }

```

6. 최대값과 최소값을 구하기 위해 max 와 min 변수를 배열 arr[]의 0 번 인덱스로 초기화 시킨 후 최대값과 최소값을 구하는 알고리즘을 통해 최대/최소값을 구한다.

```

50 | printf("몇개의 데이터씩 출력할지를 선택해 주세요: ");
51 | scanf("%d", &input); //출력할 데이터의 개수를 입력받는다
52 | printf("\n");
53 | for(i=0; i<input; i++)
54 | {
55 |     printf("%5d", i+1);
56 | }
57 | printf("\n");
58 | for(i=0; i<input*5+1; i++)
59 | {
60 |     printf("=");
61 | }
62 | printf("\n"); //초기 문장 출력

```

7. 몇개의 데이터를 출력할지 묻는 안내문을 출력한 후, 사용자로부터 정수를 입력 받아 input 변수에 저장한다.

8. 필드 폭을 5 로 지정하여 1 부터 입력 받은 정수 까지 출력하고 줄을 바꿔준다.

9. 위의 정수 출력에서 필드 폭을 5로 정해주었기 때문에 '='도 (입력 받은 변수) * 5 만큼 반복하여 출력해야 하지만 깔끔한 출력을 위해 +1 을 해준 input * 5 + 1 만큼 반복하여 출력해 준다.

```

64 line = count/input+1; //출력해야할 줄의 총 개수 == 총 데이터개수 / 한줄에 출력할 데이터의 개수 + 1
65 for(i=0; i<line; i++) //위에서 계산한 줄 수 만큼 반복
66 {
67     for(j=0; j<input; j++) //한 줄에 입력한 수 만큼 반복하여 출력
68     {
69         printf("%5d", arr[(i*input)+j]); //i번째 줄은 (i*input)+1 번 인덱스부터 출력되어야 한다
70         end++; //end변수가 count의 값과 같아지면 종료시키기 위해 end의 값을 1씩 증가시킨다
71         if(end==count) break; //end == count일 시 출력 종료
72     }
73     printf("\n");
74     for(j=0; j<input+1; j++)
75     {
76         if(arr[i*input+j]==max) printf("최대값"); //왼줄에서 출력한 값이 최대값과 같으면 "최대값" 출력
77         else if(arr[i*input+j]==min) printf("최소값"); //왼줄에서 출력한 값이 최소값과 같으면 "최소값" 출력
78         else printf(" "); //최대, 최소 모두 아닐시 공백 출력
79     }
80     printf("\n");
81     for(j=0; j<input*5+1; j++)
82     {
83         printf("-"); //밑의 -----를 input의 수와 필드폭에 맞춰 출력
84     }
85     printf("\n");
86 }

```

10. 출력해야 하는 총 줄의 개수를

(총 데이터의 개수 / 한 줄에 출력할 데이터의 개수) + 1 로 정해준다.

이때 이렇게 정해주는 이유는 예를 들어 총 63 개의 데이터를 한 줄에 15 개씩 출력해주려면 (63 / 15) + 1 인 총 5 줄을 출력 해 주어야 하기 때문이다.

또한 총 줄의 개수를 미리 정해주어야 하는 이유는 정수 데이터가 입력되는 공간, 최대/최소값 혹은 공백이 출력되는 공간, 마지막 -----가 출력되는 공간을 하나의 반복 주기로 생각하였기 때문에 이 하나의 반복 주기를 출력할 줄의 개수만큼 한번 더 반복 해 주어야 하기 때문에 미리 출력할 줄의 개수를 정해 주었다.

11. 위에서 정수를 출력할 때 필드 폭을 5로 지정 해 주었으므로 이번에도

마찬가지로 필드 폭을 5로 지정하여 출력한다. 이때 end 변수에 +1 씩을

해 준 이유는, 출력을 할 때마다 +1 씩을 해 주다가 end 변수와 count 변수의 값이 같아지면 모든 데이터의 출력을 완료 한 것이기 때문에 출력을 멈출 조건을 만들기 위해 end++; 이라는 문장을 써 주었다.

12. end 와 count 의 값이 같아지면 출력을 모두 마친것으로 간주하여

반복문을 빠져나가고 줄을 바꿔준다.

13. 위 줄에 있는 정수 값이 최대값이라면 "최대값"을 출력 해 주고 정수의 값이 최소값이라면 "최소값"을 출력 해 주고 최대값 최소값이 모두 아니라면 공백을 출력 해 준다.

14. 9 번과 마찬가지로 정수를 출력 할 때의 필드 폭을 5로 지정 해 주었으므로 -을 (입력 받은 수 * 5)번에 깔끔한 출력을 위해 1을 더한 만큼 반복하여 출력 해 준다.

```
87 | printf("\n");  
88 | fclose(fp); //파일포인터 닫기  
89 | return 0;  
90 | }
```

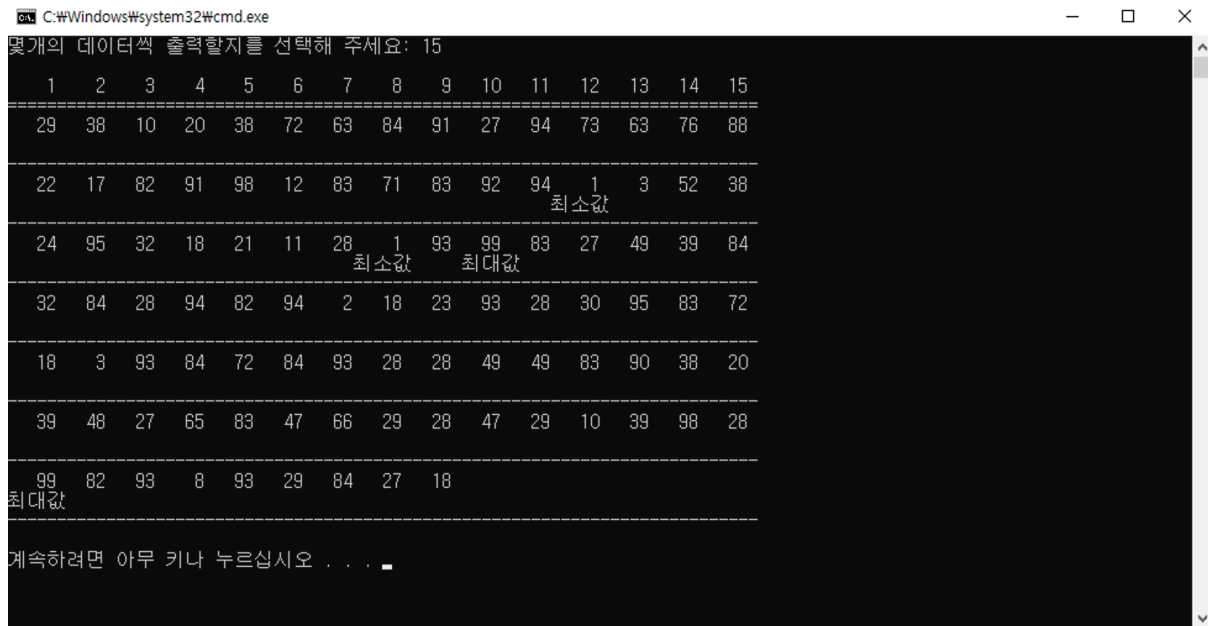
15. 반복을 모두 마쳤다면 열었던 파일을 닫고 0을 반환하며 메인 함수를 닫는다.

1.4 실행창

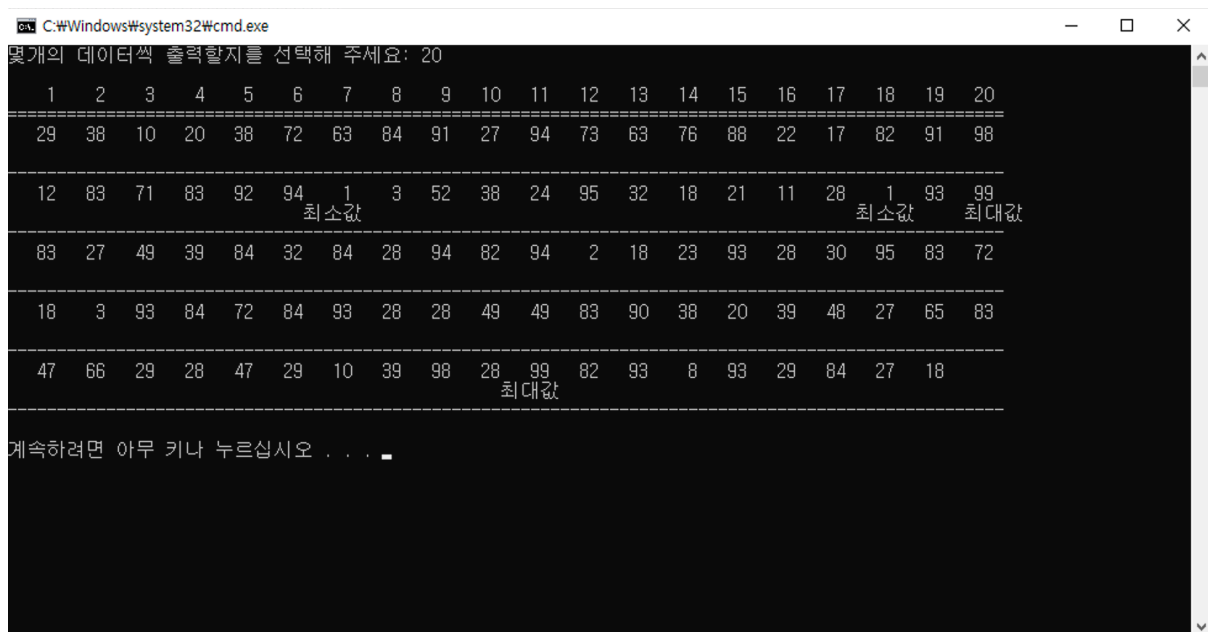
1) 데이터의 개수가 50 개 이상 10000 개 이하일 경우



<사진 1> 첫번째 테스트 파일 data.txt

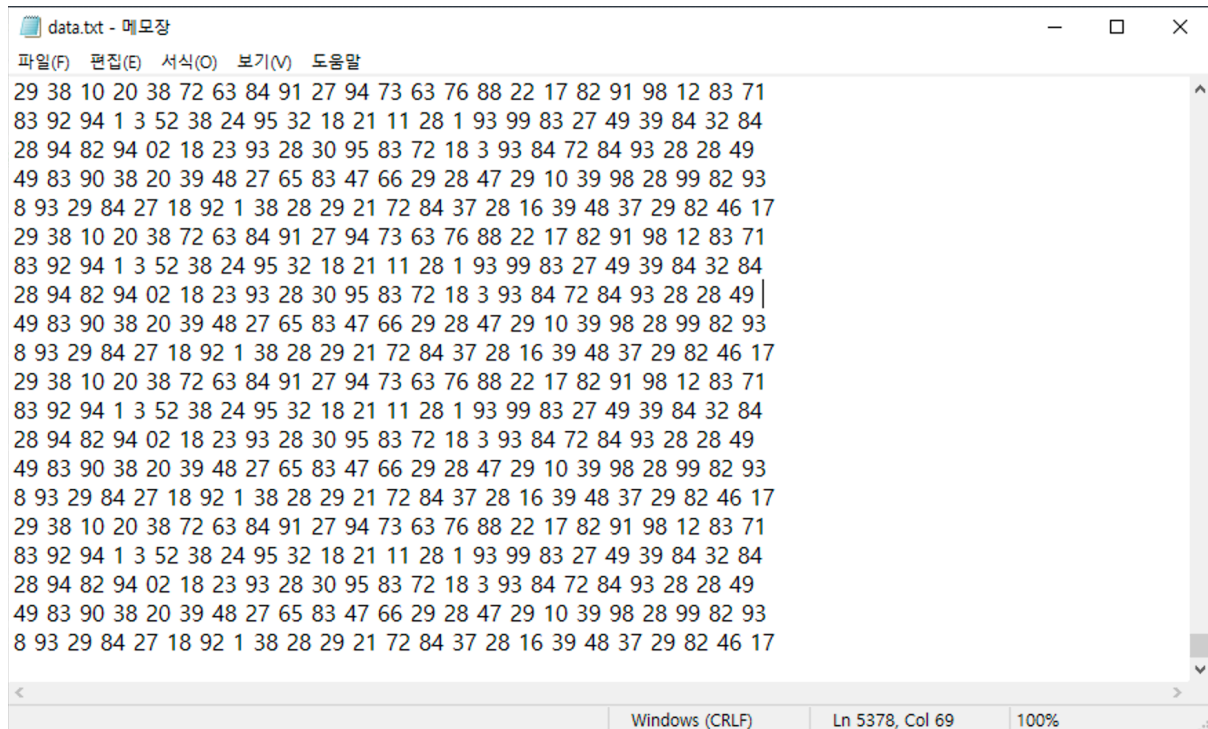


〈사진 2〉 첫번째 테스트 파일에 대한 출력 결과 (입력 : 15)

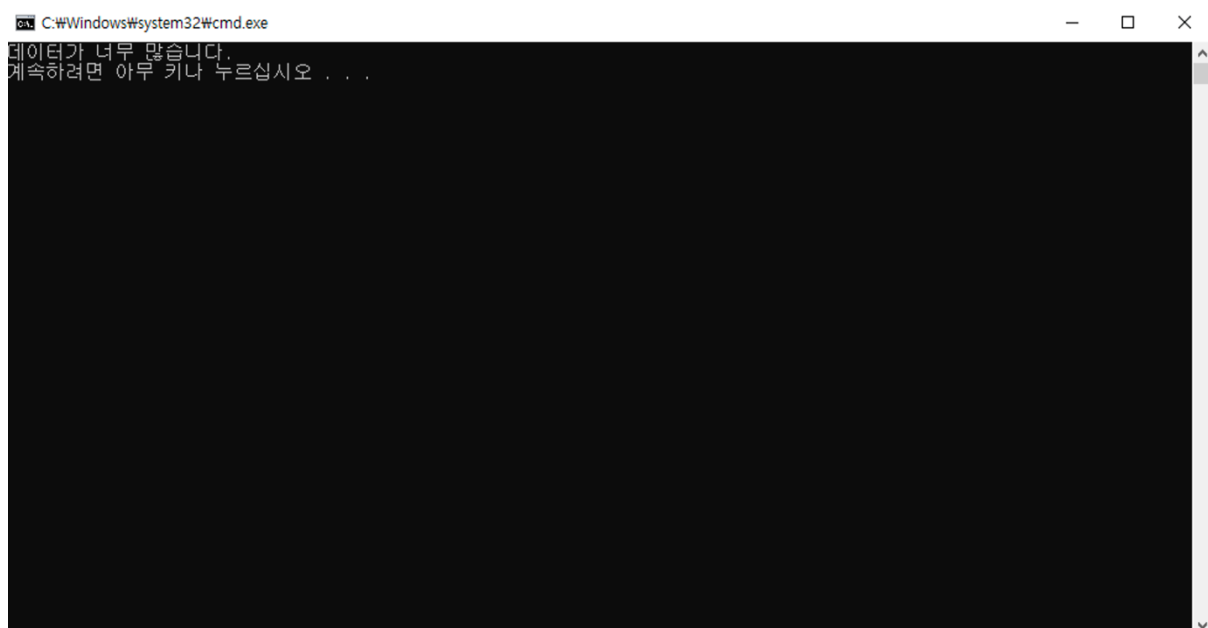


〈사진 3〉 첫번째 테스트 파일에 대한 출력 결과 (입력 : 20)

2) 데이터의 개수가 10000 개를 초과할 경우

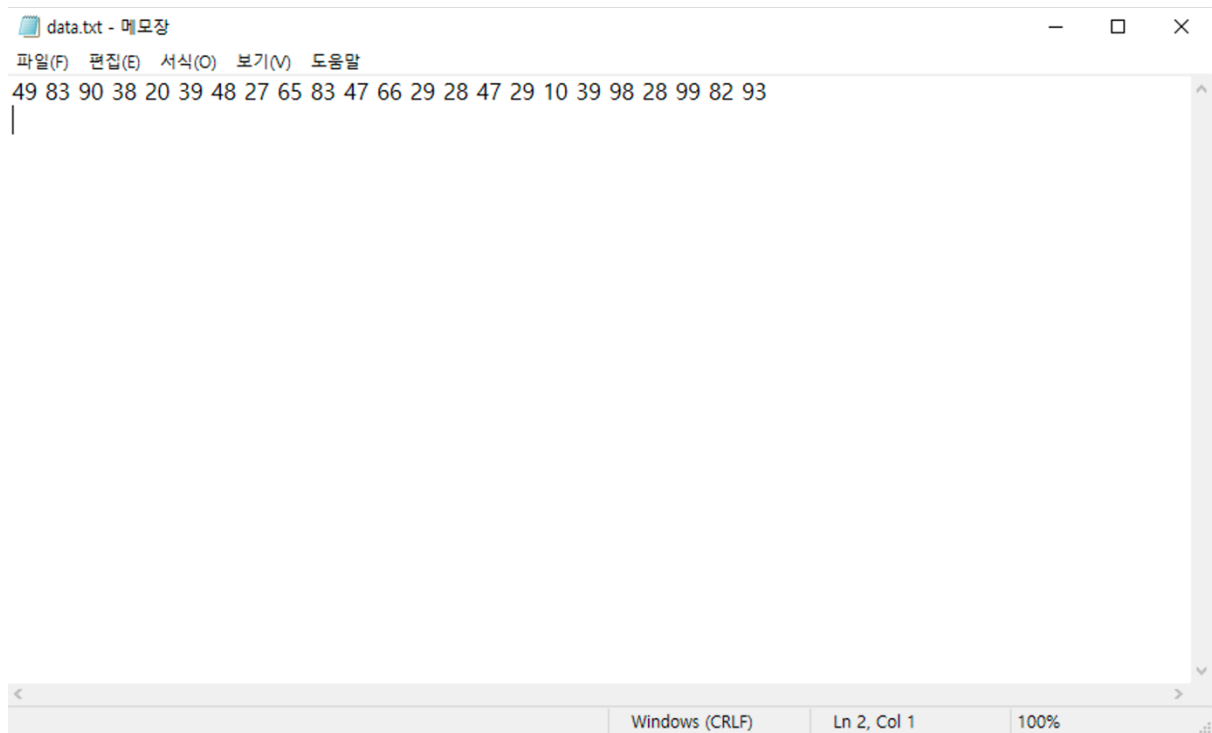


<사진 4> 두번째 테스트 파일 data.txt

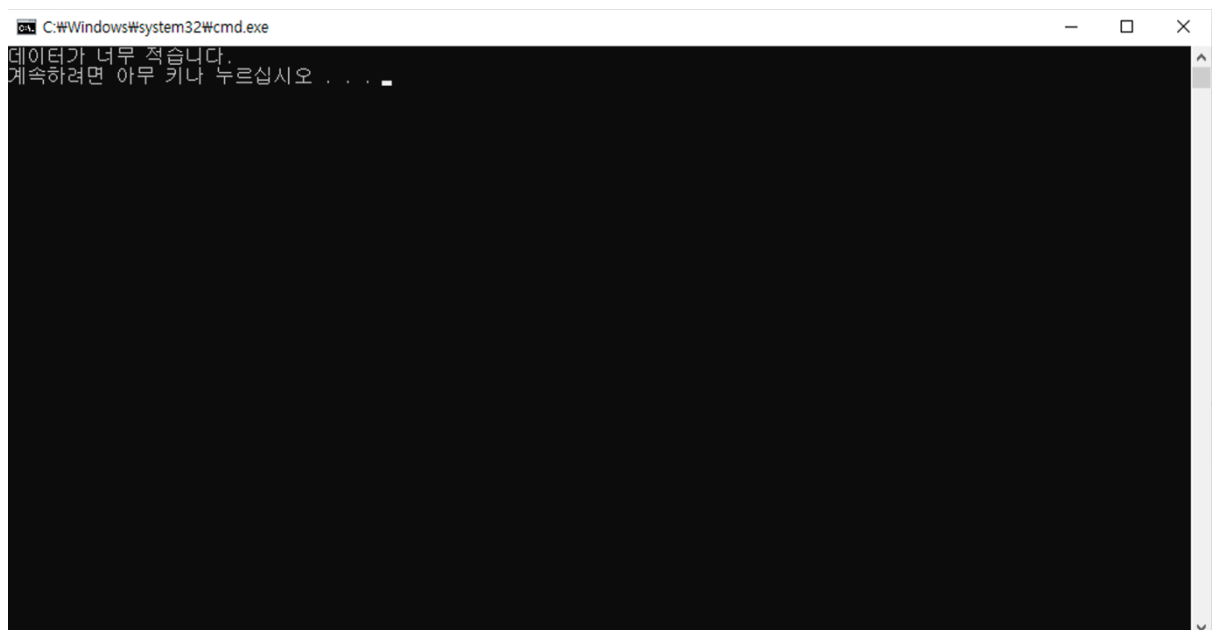


<사진 5> 두번째 테스트 파일에 대한 출력 결과

2) 데이터의 개수가 50 개 미만일 경우



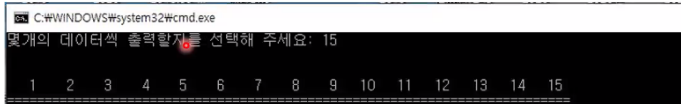
<사진 6> 세번째 테스트 파일 data.txt



<사진 7> 세번째 테스트 파일에 대한 출력 결과

1.5 느낀점

처음 이 문제를 받고 가장 먼저 든 생각은 반복 루틴을 찾아야 한다는 것 이였다. 작년 c 프로그래밍 수업을 통해 최대/최소값을 찾는 알고리즘, 파일을 읽고 쓰는 파일입출력은 모두 배웠기 때문에 그러한 부분의 코드를 작성 하는데는 어려움이 없었다. 하지만 이 프로그램이 어떤 규칙을 갖고 있는지는 쉽게 눈에 들어오지 않았다. 그렇게 열심히 문제와 예시 출력창을 보다 보니 한가지 규칙을 발견 할 수 있었다.



이 세줄, 초기 출력문을 제외하고는 정수 데이터, 최대/최소/공백, -를 이용한 줄이 세가지가 반복적인 규칙을 갖고 있다는 점이다. 따라서 이 세가지 규칙을 하나의 반복문에 넣어서 이중반복문을 통해 구현에 성공했다. 이 문제를 푸는데 큰 어려움은 없었지만 한가지 아쉬웠던 점은 아직 main 함수를 제외한 나머지 사용자 정의 함수를 사용하는데 미숙함이 있어 main 함수 안에 모든 알고리즘을 넣었다는 점이다. 다음에 시간이 생긴다면 이 소스코드를 바탕으로 최대/최소를 구하는 함수, 출력을 위한 함수 등 여러가지 함수로 모듈화 하여 다시 작성 해 보고 싶다.

2. 느낀점

많은 선배들, 그리고 제 담당 교수님 이신 김석훈교수님도 지난 겨울방학에 놀지 말고 c 언어 복습을 하고 오라고 말씀 하셨다. 그래서 이번 겨울방학을 한달 남긴 시점부터 나름 c 언어에서 내가 많이 어려웠던 부분, 앞으로 소스코드를 작성할 때 가장 중요하다고 생각되는 문법들을 다시 보고 복습했었다. 다른 친구들은 방학인데 뭐하러 공부를 하냐고 한마디씩 했었지만 그때 해냈던 것 들이 지금에서야 빛을 본 것 같다. 지금 이 문제를 풀면서 내가 방학 동안 파일 입출력, 최대/최소를 구하는 알고리즘 등 c 언어 복습을 하지 않았다면 정말 하나도 기억이 안나 하나하나 처음부터 다시 시작하는 기분 이었을 것 같다. 이 부분에 대해 스스로 정말 뿌듯하게 느껴진다. 하지만 이제 자료구조라는 과목의 첫 과제임에도 불구하고 이렇게 쉽지 않은 문제가 주어 진다는 것은 앞으로 이것보다 두배 세배는 어렵고 복잡하고 긴 문제들이 출제가 될 것 같다는 생각이 든다. 따라서 앞으로도 꾸준히 아니 더 열심히 해야겠다는 각오를 다시 한번 하게 되었다.