

11 주차 Report

문제 1)

1.한주의 레포트를 풀고 작성하는 데까지 걸리는 총 시간(평균)은?

6 ~ 7 시간 소요됩니다.

2. 하나의 프로그래밍 문제에 대해 순수 알고리즘 만을 고민하는 시간은 대략 얼마인가?

5 ~ 10 분 소요됩니다.

3. 학습한 여러 파트 중에서 가장 어려웠던 파트는 무엇인가요? 또한 어려운 파트의 학습은 어떻게 하였나요? 혹시, 지금도 잘 이해가 안되는 파트가 있다면 어느 파트입니까?

현재까지 어려웠던 파트가 없었습니다.

4. 지금까지 학습한 내용에 대해 본인의 이해도는 몇 % 인가요?

95 % 정도라 생각합니다

5. 본 교과목의 학습은 어떻게 하였나요?

강의를 통한 학습 후 강의 자료와 교재로 복습하고 레포트 작성 시 부분부분 교재를 참조하거나 인터넷에 바로 검색을 해봅니다.

6. 향후 비대면 온라인 교육에서 보강, 개선되었으면 하는 점이 있다면? (아이디어?)

교수님의 수업 부분에 대해서는 저는 충분히 흥미를 느끼고 잘 따르려 합니다. 그러나 레포트의 양이 아주 많습니다..... 교수님께서 각자에게 문법 내용 숙지와 이해를 목적으로 코드당 설명과 각 주석을 다 달도록 레포트를 내주시는 것은 충분히 좋은 과정일 것입니다. 그러한 부분은 저도 만족을 합니다만, 문제 수가 많습니다..... 저는 워낙 프로그래밍을 좋아하여 꽤 이른나이부터 이런 분야에 관심을 가지게 되었지만, 매주 나오는 레포트의 양을 보고 어떤 때에는 엄청 하기 싫을 때도 많았습니다. 저는 이러한 부분을 레포트 작성과 BlackBoard 의 Quiz 풀이를 병행하여 레포트 문제 수를 줄이는 것도 괜찮은 방법이라 생각합니다. 한 주차에 배운 개념 중 중요한 부분을 1~2 문제로 레포트에 내고 나머지 부분은 Quiz 를 통해 학습할 수 있도록 하는 것입니다.

7. 본 교과목 관련 본인이 하고 싶은 이야기를 자유롭게 기술 하시오.

각 주마다 제출한 레포트에 주어지는 가산점에 대한 기준이나 그렇게 주어진 이유가 궁금합니다!

문제 2) 다음 프로그램의 오류를 전부 수정하라.

case. 1

Before

```
#define MAX_SIZE 3

int main(void){

    int a[MAX_SIZE]={0, 1, 2, 3};
    int b[3.0];

    for(i=0; i<=MAX_SIZE; i++)
        b[i]=a[i];

    return 0;
}
```

After

```
#define MAX_SIZE 3

int main(void){

    int a[MAX_SIZE]={0, 1, 2}; // 맨 위에서 MAX_SIZE 를 3 으로 정의내렸다.
    //따라서 배열을 초기화 할 때 MAX_SIZE 만큼의 크기를 가지므로 하나를 지웠다.
    int b[3]; // index 에는 정수만 들어와야 할 것이다

    for(i=0; i<=MAX_SIZE; i++)
        b[i]=a[i];

    return 0;
}
```

case. 2

Before

```
#define MAX_SIZE 3

int main(void){

    int a[MAX_SIZE]={0, 1, 2, 3};
    int b[3.0];

    for(i=0; i<=MAX_SIZE; i++)
        b[i]=a[i];

    return 0;
}
```

After

```
#define MAX_SIZE 4
// 배열을 초기화 할 때 초기값이 4 개이므로 MAX_SIZE 를 4 로 바꾸었다.

int main(void){

    int a[MAX_SIZE]={0, 1, 2, 3};
    int b[4]; // index 에는 정수만 들어와야 할 것이다

    for(i=0; i<=MAX_SIZE; i++)
        b[i]=a[i];

    return 0;
}
```

문제 3) 크기가 10 인 1 차원 배열에 난수를 저장한 후에, 최대값과 최소값을 출력하는 프로그램을 작성하라. 난수는 rand() 함수를 호출하여 생성하라.

Hint : `x[i] = rand();` // 난수를 생성하여서 i 번째 배열 요소에 대입한다.

- 문제 분석 & 동작 설명

크기가 10 인 1 차원 배열을 형성한 후, 힌트처럼 `x[i] = rand();` 를 이용해 랜덤 한 수를 배열의 방 하나씩 끼워 넣어주도록 하자. 이 문제에서 핵심이 되는 부분은 배열 내에서 최대값과 최소값을 판별해 내야하는 것이다. 따라서 이는 배열의 첫번째 요소를 최소값과 최대값으로 다 가정하고 배열의 두번째 요소부터 마지막 요소까지 이 최소값과 최대값을 비교하도록 하자. 만약 배열의 요소가 현재의 최소값보다 작다면 새로운 최소값으로 갱신하고, 현재의 최소값보다 크다면 새로운 최대값으로 갱신하도록 하자. 그러고나서 마지막에 최종 최소값과 최대값을 출력해주면 되겠다.

- 프로그램 소스 (주석 포함)

```
Source.c* x
Week11 Report // 배열에 난수를 불러 최대 최소 구별하는 프로그램
1 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
2
3 int main(void) { // main function
4
5     int x[10], i, minimum, maximum;
6     // 정수형 10칸짜리 배열 x, 반복문 돌릴 정수형 변수, 최소값과 최대값을 저장할 정수형 변수 초기화
7
8     for (i = 0; i <= 9; i++) { // i가 0부터 1씩 더하여 9가 될때까지 아래의 문장을 반복
9         x[i] = rand(); // x 배열의 각 칸에 랜덤한 수를 넣음
10    }
11
12    // 배열 내 요소들의 비교를 통해 최소, 최대값 판별
13    minimum = x[0]; // 첫번째 요소를 최소값으로 지정
14    maximum = x[0]; // 첫번째 요소를 최대값으로도 지정
15
16    for (i = 0; i <= 9; i++) { // i가 0부터 1씩 더하여 9가 될때까지 아래의 문장을 반복
17        if (maximum < x[i]) // 만약에 x[i]가 한칸 앞인 x[i-1]보다 크다면
18            maximum = x[i]; // 그 x[i]는 최대값이므로 maximum에 저장
19
20        if (minimum > x[i]) // 만약에 x[i]가 한칸 앞인 x[i-1]보다 작다면
21            minimum = x[i]; // 그 x[i]는 최소값이므로 minimum에 저장
22    }
23    // 배열 내 요소 중 전 후 관계의 크기를 비교하여
24    // 반복문의 매 턴마다 끝날때까지 최소값, 최대값이 갱신됨
25
26    printf("최대값은 %d\n최소값은 %d", maximum, minimum); // 배열 내에서 판별한 최대값, 최소값을 출력
27
28    return 0; // main function 이 0으로 반환
29 }
```

Microsoft Visual Studio
최대값은 29358
최소값은 41
C:\Users\Gyuha_wa...
port 3\Debug\Week1...
To automatically d...
le when debugging...
Press any key to c...

문제 4) 2 개의 정수 배열 a, b 를 받아서 대응되는 배열 원소가 같은지를 검사하는 함수 `array_equal(int a[], int b[], int size)`를 작성하고 테스트하라. 이 함수는 `a[0]`와 `b[0]`, `a[1]`과 `b[1]`, ... , `a[size-1]`와 `b[size-1]`가 같은지를 검사한다. 만약 전체 원소가 같다면 1 을 반환하고 그렇지 않으면 0 을 반환한다.

Hint : 반복 루프를 이용하여서 배열의 각 요소가 같은지를 검사한다. 만약 하나라도 다르면 0 을 바로 반환하면 된다.

- 문제 분석 & 동작 설명

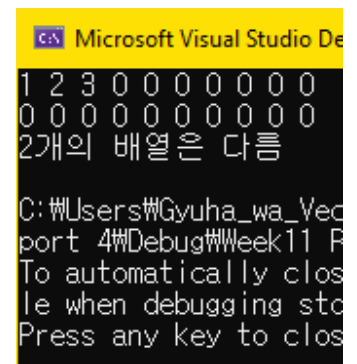
10 의 같은 사이즈 배열을 2 개를 선언하여 각각 입력받도록 하자. 그 때, 반복문을 이용하여 한 칸씩 차례대로 넣어주도록 하자. 문제에서 사용자 정의함수를 생성해라 하였다. 그 함수는 두 배열의 각 요소가 같은지 다른지 비교하는 기능을 포함하여야 한다. 이것도 반복문을 이용하여 각 칸의 요소를 비교하여야 한다. 같으면 1 로 리턴하고, 다르면 그 달라지는 순간에 반복문을 탈출하여 0 으로 리턴해주자. 리턴 해준 값이 1 이면 입력받은 두 배열은 같다고 출력하고, 0 이면 입력받은 두 배열은 다르다고 출력해주자.

- 프로그램 소스 (주석 포함)

```

Source.c
1 // 배열의 요소를 비교하여 같은지 판별해주는 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3 #define SIZE 10 // 배열 사이즈를 10으로 정의
4
5 int array_equal(int x[], int y[], int size) // 사용자 지정 함수 array_equal
6 // 메인함수의 배열 a, b를 받을 정수형 배열 x,y를 매개변수로 지정
7 // 10으로 정의된 SIZE를 받을 정수형 변수 size를 매개변수로 지정
8 {
9     int i, result; // 반복문 돌릴 정수형 지역변수 i 선언
10     // a, b 배열의 요소 각각을 비교하고 1 or 0을 저장할 정수형 지역변수 result 선언
11
12     for (i = 0; i < SIZE; i++) // i가 0부터 1씩 더해지면서 SIZE-1이 될 때까지 아래의 문장을 반복함
13     {
14         if (x[i] == y[i]) // 만약에 x[i]와 y[i]의 수가 같다면
15             result = 1; // result에 1을 저장
16         else {
17             // 만약에 x[i]와 y[i]의 수가 다르다면
18             result = 0; // result에 0을 저장
19             break; // 두 배열이 다른 것을 확인하였으므로 더 이상 확인 없이 for문 탈출
20         }
21     }
22     return result; // result 값을 정수형으로 리턴
23 }
24
25 int main(void) { // main function
26     int a[SIZE], b[SIZE], i;
27     // 두 개의 정수 배열 a, b와 반복문 돌릴 정수형 지역변수 i 선언
28
29     // 정수 배열 a, b 입력받기 위한 반복문
30     for (i = 0; i < SIZE; i++) { // i가 0부터 1씩 더해지면서 SIZE-1이 될 때까지 아래의 문장을 반복함
31         scanf_s("%d", &a[i]); // 정수 배열 a의 0번째에서 9번째까지 방에 넣을 10개의 요소 입력받음
32     }
33
34     for (i = 0; i < SIZE; i++) { // i가 0부터 1씩 더해지면서 SIZE-1이 될 때까지 아래의 문장을 반복함
35         scanf_s("%d", &b[i]); // 정수 배열 b의 0번째에서 9번째까지 방에 넣을 10개의 요소 입력받음
36     }
37
38     if (array_equal(a, b, SIZE) == 1) // 만약 사용자 지정 함수 array_equal의 리턴 값이 1과 같다면
39         printf("2개의 배열은 같음\n"); // 두 개의 정수 배열 a, b가 같다고 출력
40     else if (array_equal(a, b, SIZE) == 0) // 만약 사용자 지정 함수 array_equal의 리턴 값이 0과 이면
41         printf("2개의 배열은 다름\n"); // 두 개의 정수 배열 a, b가 다르다고 출력
42
43     return 0; // main function 이 0으로 반환
44 }

```



문제 5) 다음과 같은 2 차원 표를 배열로 생성하고, 각 열의 합계를 구하여 출력하는 프로그램을 작성하라.

12	56	32	16	98
99	56	34	41	3
65	3	87	78	21

Hint : 2 차원 배열을 주어진 표로 초기화 한다. 각 행의 합계, 각 열의 합계를 중첩 반복문을 통하여 계산한다.

- 문제 분석 & 동작 설명

문제에 3 개의 행과 5 개의 열로 구성된 2 차원 배열의 표가 주어져 있다. 행 단위로 끊어 2 차원 배열을 선언해주자. 표의 (0,0) 부터 (4,0)까지의 요소를 반복문을 이용하여 다 더하도록 하고, 반복문의 중첩을 이용하여 (0,1) 부터 (4,1), (0,2) 부터 (4,2)의 요소도 다 더해주도록 하자. 이렇게 각 행 단위의 더해진 값들을 출력해주자. 그리고 표의 (0,0) 부터 (0,2)까지의 요소를 반복문을 이용하여 다 더하도록 하고, 반복문의 중첩을 이용하여 (1,0) 부터 (1,2), (2,0) 부터 (2,2), (3,0) 부터 (3,2), (4,0) 부터 (4,2)의 요소도 다 더해주도록 하자. 이렇게 각 열 단위의 더해진 값들을 출력해주자.

- 프로그램 소스 (주석 포함)

```

1 // 2 차원 배열의 행끼리, 열끼리의 합을 구하는 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3
4 // 2차원 표의 행과 열을 정의
5 #define ROWS 3 // 3개의 행
6 #define COLS 5 // 5개의 열
7
8 int main() { // main function
9
10     // 문제에 제시된 표의 값들을 배열에 초기화
11     int arr[ROWS][COLS] = {
12         {12,56,32,16,98}, // 0행
13         {99,56,34,41,3}, // 1행
14         {65,3,87,78,21}, // 2행
15     };
16
17     int rows, rowSum, cols, colSum;
18     // 행의 수, 그 행 요소들의 합, 열의 수, 그 열 요소들의 합을 저장할 변수 각각 선언
19
20     for (rows = 0; rows < ROWS; rows++) { // 행이 0부터 1씩 더하여 2가 될때까지 반복
21         rowSum = 0; // 각 행 요소들의 합을 0으로 초기화
22         for (cols = 0; cols < COLS; cols++) // 열이 0부터 1씩 더하여 4가 될때까지 반복
23             rowSum += arr[rows][cols]; // 각 행의 요소들의 합계를 rowSum에 저장
24         printf("%d행의 합계 : %d\n", rows, rowSum); // 합계를 출력
25     }
26
27     for (cols = 0; cols < COLS; cols++) { // 열이 0부터 1씩 더하여 4가 될때까지 반복
28         colSum = 0; // 각 열 요소들의 합을 0으로 초기화
29         for (rows = 0; rows < ROWS; rows++) // 행이 0부터 1씩 더하여 2가 될때까지 반복
30             colSum += arr[rows][cols]; // 각 열의 요소들의 합계를 colSum에 저장
31         printf("%d열의 합계 : %d\n", cols, colSum); // 합계를 출력
32     }
33     return 0; // main function 이 0으로 반환
34 }

```

```

Microsoft Visual Studio
0행의 합계 : 214
1행의 합계 : 233
2행의 합계 : 254
0열의 합계 : 176
1열의 합계 : 115
2열의 합계 : 153
3열의 합계 : 135
4열의 합계 : 122

C:\Users\Gyuha_wa_...
port 5\Debug\Week1
To automatically c
le when debugging
Press any key to c

```

문제 6) 사용자가 입력하는 10 개의 실수 자료의 평균과 표준 편차를 계산하는 프로그램을 작성하라. 자료들은 난수를 생성하여서 작성된다. (정수로 생성하여서 실수로 변환하라.) 평균은 n 개의 실수가 주어질 때, 다음과 같이 계산된다.

$$\text{평균} : m = \frac{1}{n} \sum_{i=1}^n x^i$$

표준 편차는 분산의 양의 제곱근으로 분산은 다음과 같이 계산된다. 표준 편차는 자료가 평균값 주위에 어느 정도의 넓이를 분포하고 있는가를 나타내는 하나의 척도이다.

$$\text{분산} : v = \frac{1}{n} \sum_{i=1}^n (x^i - m)^2$$

Hint : 사용자가 입력하는 자료 값들은 모두 배열에 저장한다. 평균과 표준편차를 구하는 함수를 작성하고 함수의 인수로 배열을 넘기도록 하자.

- 문제 분석 & 동작 설명

입력된 10 개의 수치를 이용하여 평균과 표준편차를 계산해야 한다. 10 개의 수치를 저장할 배열을 만들고, 그 배열을 가지고 놀면 될 것이다. 문제에서는 입력 받은 값을 실수로 변환해라 했으므로 배열 선언할 때 실수형으로 선언해주면 되겠다. 배열에 10 개 값을 입력 받아야 하므로 이는 반복문을 통해 하나씩 받아 배열에 저장해주자. 힌트에서 평균과 표준편차를 구하는 함수를 작성하자 했으니 힌트를 따라 함수를 각각 만들어주고 함수의 매개변수로 배열과 배열의 크기를 넘기도록 하자. 평균을 구하는 함수 내에서는 반복문을 이용하여 배열을 이루는 각 요소들의 총합을 계산한 뒤 배열의 크기로 나누어 주면 될 것이다. 이를 출력하도록 하자. 그리고 표준편차를 구하는 함수 내에서는 반복문을 이용하여 배열을 이루는 각 요소들에서 앞서 구했던 평균을 빼 값의 제곱값을 계산한 뒤 배열의 크기로 나누고 루트를 씌워주면 될 것이다. 이를 출력하도록 하자

- 프로그램 소스 (주석 포함)

(Line 1 ~ Line 33)

```

1 // 배열의 요소들의 평균과 표준편차를 구하는 프로그램
2 #include <stdio.h> // Standard Input Output 함수 헤더파일 전처리기
3
4 #include <math.h> // 제곱근을 계산해주는 함수 sqrt()를 사용하기 위한 math 함수 헤더파일 전처리기
5
6 #define SIZE 10 // SIZE를 10으로 정의
7
8 static double average, deviation;
9 // 평균과 표준편차 연산하는 함수 내에서 반복문 돌릴때마다 나오는 계산값인 평균과 편차값이 계속 누적되도록 실수형 정적 변수로 선언
10
11 void avg(double* a, int size) { // 배열 내 요소들의 평균을 구해주는 사용자 지정 함수 avg
12
13     int i; // 사용자 지정 함수 avg 내의 반복문에서 사용 할 정수형 지역변수 i 선언
14     double sum = 0; // 배열 내 요소들의 합을 저장할 정수형 지역변수 sum을 0으로 초기화
15
16     for (i = 0; i < size; i++) // i가 0부터 1씩 더해져 9가 될때까지
17         sum += a[i]; // 각 요소들의 합을 누적하여 계산 후 sum에 저장
18
19     average = sum / size; // 계산한 값을 배열의 크기인 10으로 나누고 이를 평균 average에 저장
20
21     printf("평균값은 %f \n", average); // average 출력
22 }
23
24 void stdev(double* a, int size) { // 배열 내 요소들의 표준편차를 구해주는 사용자 지정 함수 stdev
25
26     int i; // 사용자 지정 함수 stdev 내의 반복문에서 사용 할 정수형 지역변수 i 선언
27     double sum = 0; // 배열 내 요소들의 합을 저장할 정수형 지역변수 sum을 0으로 초기화
28
29     for (i = 0; i < size; i++) // i가 0부터 1씩 더해져 9가 될때까지
30         sum += (a[i] - average) * (a[i] - average); // 각 요소에서 평균을 뺀 값의 제곱을 누적하여 계산 후 sum에 저장
31
32     deviation = sqrt(sum / size); // 계산한 값을 배열의 크기인 10으로 나누고 루트 씌워서 이를 표준편차 deviation에 저장
33     printf("표준편차값은 %f \n", deviation); // deviation 출력
34 }

```

(Line 34 ~ Line 50)

```

34 int main() { // main function
35
36     double arr[SIZE] = { 0 }; // 크기가 10짜리 배열 arr의 요소를 모두 0으로 초기화
37     int i; // 반복문을 돌리기 위해 정수형 변수 i 선언
38
39     for (i = 0; i < SIZE; i++) { // i가 0부터 1씩 더해져 9가 될때까지 아래의 문장들을 반복
40         printf("데이터를 입력하시오: "); // 사용자에게 데이터를 입력해라고 출력
41         scanf_s("%lf", &arr[i]); // 입력받은 데이터를 엔터단위로 입력받아서 배열 arr에 0번 방부터 9번 방까지 순차적으로 저장
42     }
43
44     avg(arr, SIZE); // 배열 내 요소들의 평균을 구해주는 사용자 지정 함수 avg의 실수형 매개변수 a에는 배열 arr,
45     // 정수형 매개변수 size에는 위에서 정의된 SIZE를 올려줌
46     stdev(arr, SIZE); // 배열 내 요소들의 표준편차를 구해주는 사용자 지정 함수 stdev의 실수형 매개변수 a에는 배열 arr,
47     // 정수형 매개변수 size에는 위에서 정의된 SIZE를 올려줌
48
49     return 0; // main function 이 0으로 반환
50 }

```



```
Microsoft Visual Studio Debug Console
데이터베이스 연결: 10
데이터베이스 연결: 20
데이터베이스 연결: 30
데이터베이스 연결: 40
데이터베이스 연결: 50
데이터베이스 연결: 60
데이터베이스 연결: 70
데이터베이스 연결: 80
데이터베이스 연결: 90
데이터베이스 연결: 100
평균값: 55.000000
표준편차값: 28.722813

C:\Users\Gyuha_wa_Vect0r\Documents\학교\4. 대학교\충북대학교\1학년\기초컴퓨터프로그래밍 10분반\1주차\과제\6번\Week11 Report 6\Debug\Week11 Report 6.exe (process 18588) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```