

박기홍 4일차 과제

1. HW_001

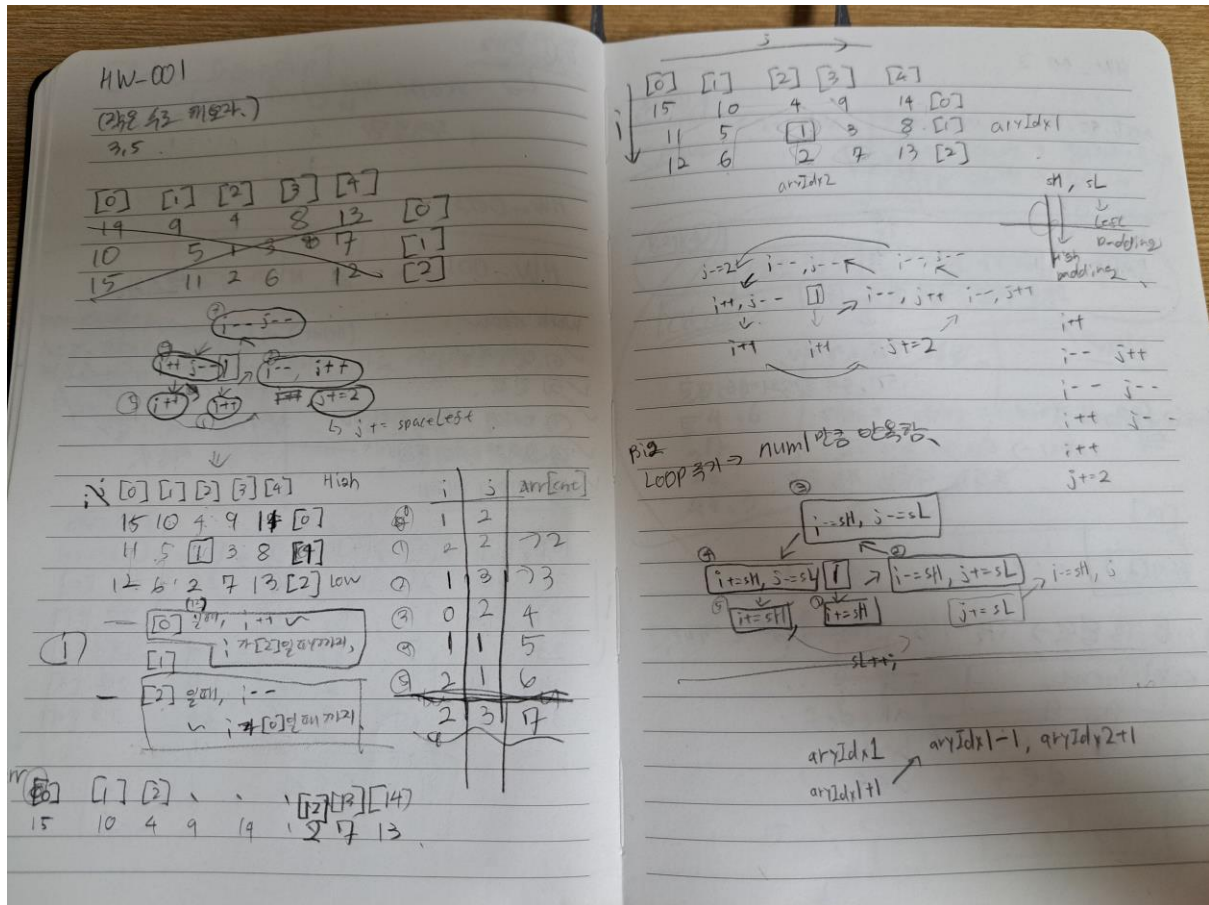
본 과제를 수행하기 위해 다양한 방법으로 접근하도록 시도하고, 규칙을 추론해 보았습니다. 본 과제도 다른 과제와 마찬가지로 2024년 07월 04일 오전 10시 이후부터 ~ 2024년 07월 05일 오전 5시 30분까지 계속 문제 풀이를 시도해 보았으나, 과제 1은 끝내 완수하지 못했습니다. 저의 능력 부족임을 인정하며, 비슷한 유형의 과제를 수행해 낼 수 있도록 로빗 수습 단원 교육 외에 추가적으로 알고리즘을 공부하고 많이 찾아보겠습니다. 현재까지 진행한 소스코드와 테스트 케이스를 보내드립니다.

[규칙 풀이]

The image shows handwritten notes on two pages of lined paper. The left page is titled "[Robit_Internship]" and "2024.08.09 - 4일차". It lists memory management functions: malloc, calloc, realloc, and free, with brief descriptions in Korean. Below this, it says "HW-001" and "input: 5, 7". A small diagram shows a box with "5" and "7" inside, with arrows pointing to "5" and "7" respectively. The right page is titled "HW-002" and "stack 개념 (push, pop) + 동적할당". It then says "HW-003" and "HW-001". A "Work flow" section lists steps: 1. 연속 선분, 2. 입력, 3. 배열 - 동적 할당하기, 4. 2차원 배열 출력, 5. 메모리 해제. To the right of these steps, there are notes about "이진 탐색" (binary search) and "이진 탐색을 위한 배열" (array for binary search). Below the work flow, there is a 7x7 grid of numbers. The grid is as follows:

35	30	21	11	20	29	34
31	22	12	4	10	19	28
23	13	5	1	3	9	18
24	14	6	2	8	17	27
32	25	15	7	16	26	33

Below the grid, there are two boxes containing the formulas $\frac{(j-1)/2}{6}$ and $\frac{(5-1)/2}{7}$.



[소스코드]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
```

```
int main() {
```

```
    /* 변수 선언하기.
```

```
    * tempNum : 숫자를 pArr에 넣을 때, 해당 숫자의 정보를 나타냄.
    */
```

```
    int num1, num2, i, j, cnt = 0, tempCnt = 0, tempNum = 1;
```

```
    // pArr 전용 index Checker
```

```
    int aryIdxX = 0, aryIdxY = 0;
```

```
    /* aryIdx1과 aryIdx2의 변화를 주기 위한 논리 연산자.
```

```
    * (but, boolean은 안 배운 개념이므로 int type을 사용함)
```

```
    * 0 : false, 1 : true
```

```
    * aryIdx1(isHigh)의 값이 0일 때 : High(1), num1일 때 : Low(0).
```

```
    * aryIdx2(isLeft)의 값이 0일 때 : Left(1), num2일 때 : Left(0).
```

```
    * spaceHigh, spaceLeft : 한 바퀴씩 회전해 돌아 왔을때, 몇 칸 뛰어서 채워야 하는지를
```

나타냄.

```
    */
```

```
    int isHigh = 0, isLeft = 0, spaceHigh = 1, spaceLeft = 1;
```

```
    // 형식 입출력 하기.
```

```
    printf("입력 : ");
```

```
scanf("%d %d", &num1, &num2);
```

```
/* 동적 할당하기.
```

```
* 이중 포인터 X -> 이중 포인터 개념은 배우지 않았기 때문임.
```

```
* int type의 2차원 배열 생성 -> 해당 값들을 동적 배열 (포인터 배열)에 할당하기
```

```
*/
```

```
int* pArr = (int*)calloc((num1 * num2), sizeof(int));
```

```
/* 배열의 기준 점(= 가운데 점)
```

```
* refX = j, refY = i.
```

```
*/
```

```
int refX = 0, refY = 0;
```

```
// 가운데 숫자 지정에 위한 for문.
```

```
for (i = 0; i < num1; i++) {
```

```
    for (j = 0; j < num2; j++) {
```

```
        // 가운데 숫자 지정하기
```

```
        if ((num2 - 1) / 2 == j && (num1 - 1) / 2 == i) {
```

```
            pArr[tempCnt] = tempNum; // tempNum = 1
```

```
            aryIdxY = (num1 - 1) / 2;
```

```
            aryIdxX = (num2 - 1) / 2;
```

```
            tempNum++;
```

```
            refX = aryIdxX;
```

```
            refY = aryIdxY;
```

```
        }else if((num2 - 1) / 2 == j && i == (num1 - 1) / 2 + 1){ // 가운데
```

숫자 바로 아래일 때,

```
        /* 반복문 탈출하기.
```

```
* -> 가운데 숫자를 제외한 모든 숫자를 채우기 위해 반복문을
```

탈출함.

```
        */
```

```
        pArr[tempCnt] = tempNum; // tempCnt = 12, tempNum = 2
```

```
        cnt = tempCnt;
```

```
        aryIdxY++;
```

```
        break;
```

```
    }
```

```
    tempCnt++;
```

```
}
```

```
}
```

```
/*printf("%d %d %d\n", cnt, tempCnt, tempNum);
```

```
printf("%d %d %d %d\n", aryIdxX, aryIdxY, refX, refY);*/
```

```
// 가운데 제외 숫자를 지정하는 for문.
```

```
for (i = 0; i < num1; i++) {
```

```
    for (j = 0; j < num2; j++) {
```

```
        // 숫자 작성하는 알고리즘.
```

```
        if (aryIdxX > refX) {
```

```
            if (aryIdxY > refY) {
```

```
                aryIdxX++;
```

```
                aryIdxY--;
```

```
                /*printf("%d %d\n", aryIdxX, aryIdxY);*/
```

```
            }
```

```
        }else if(aryIdxX == refX){
```

```
            if (aryIdxY > refY) {
```

```
                aryIdxX++;
```

```
                aryIdxY--;
```

```

        pArr[tempCnt] = ++tempNum;
        /*printf("%d %d\\n\\n", aryIdxX, aryIdxY);*/
    }
}
}

```

```

cnt = 0;
// 숫자 출력 알고리즘.
printf("\\n출력 : ");
for (i = 0; i < num1; i++) {
    if (i != 0) { // 출력 형식을 준수하기 위한 조건문.
        printf(" ");
    }
    for (j = 0; j < num2; j++) {
        printf("%d ", pArr[cnt]);
        cnt++;
    }
    printf("\\n");
}

```

```

free(pArr);

```

```

}

```

[실행결과]

Test Case #1

```

Microsoft Visual Studio 디버그 콘솔
입력 : 5 7

출력 : 0 0 0 0 0 0 0
      0 0 0 0 0 0 0
      0 0 0 1 0 0 0
      0 0 0 2 0 0 0
      0 0 0 3 0 0 0

C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\WC_Language\HomeWork\HW_20240704\64\Debug\HW_20240704.exe
(프로세스 13052개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

Test Case #2

```
Microsoft Visual Studio 디버그 콘솔
입력 : 11 15

출력 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0

C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\HW_20240704.exe
(프로세스 25068개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

Test Case #3

```
Microsoft Visual Studio 디버그 콘솔
입력 : 9 11

출력 : 0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 1 0 0 0 0 0 0
      0 0 0 0 0 2 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0 0 0 0 0
      0 0 0 0 0 3 0 0 0 0 0 0

C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\HW_20240704.exe
(프로세스 14584개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

2. HW_002

[소스코드]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

int main() {
    /*변수 생성 및 동적 할당하기
    * htmlArySize는 max 값을 몰라서 임의로 지정한 것임.
    * 나중에 수정할 필요 있음.
    */
```

```

int htmlArySize = 1;
int i;
int templdx = 0, targetIdx = 0, tempWordSize = 0;
char* htmlAry = (char*)calloc(htmlArySize, sizeof(char) * 100);

// 형식 입력출력하기
printf("입력 : ");
scanf("%s", &htmlAry[0]);

int j = 0, k = 0;
int isBreak = 0, tagIdx = 0;
int tagCnt = 0; // 태그의 개수 확인용 (태그 개수가 5개다? -> 잘못된 html 코드)
int wordCnt = 0; // 단어 개수 확인용.

printf("Wn출력 : ");

// 예외 처리 위한 알고리즘.
while (htmlAry[templdx] != NULL) {
    // 1. '<'로 시작 ~ '>'로 끝날 때,
    if (htmlAry[templdx] == '<') {
        while (1) {
            if (htmlAry[j] == '>') {
                tempWordSize++;
                isBreak = 0;
                tagCnt++;
                wordCnt++; // 단어 수에 포함 시키기1
                break;
            }else if (htmlAry[j] != '>') {
                tempWordSize++;
            }
            templdx++;
            j++;
            tagIdx++;
        }
        /*printf("%d %d", templdx, j);*/
    }else if (htmlAry[templdx] != '>' && htmlAry[templdx] != '<') {
        // 2. '>'로 시작 ~ '<'로 끝날 때 = '>'와 '<'로 시작하고 끝나지 않을
        while (1) {
            if (htmlAry[j] == '>' || htmlAry[j] == '<') {
                /*tempWordSize++;*/
                isBreak = 1;
                wordCnt++; // 단어 수에 포함시키기 2
                break;
            }else if (htmlAry[j] != '>' && htmlAry[j] != '<') {
                tempWordSize++;
                isBreak = 0;
            }
            templdx++;
            j++;
        }
    }

    if (isBreak == 1) {
        continue;
    }
}

```

때

```

    }
    templdx++;
    j++;
    tagldx++;
}

if (tagCnt % 2 != 0) {
    printf("잘못된 태그 구조이므로, 프로그램을 종료합니다.\n");
    return 0;
}

// 초기화
j = 0, templdx = 0, targetldx = 0, tempWordSize = 0, isBreak = 0, tagldx = 0;
tagCnt = 0; // 태그의 개수 확인용 (태그 개수가 5개다? -> 잘못된 html 코드)
// 기존 wordCnt는 출력을 위해 비교 변수로 사용됨. wordCnt2는 비교 변수와 비교를 할 때
// 사용하기 위해 선언함.
int wordCnt2 = 0;

// 출력을 위한 알고리즘.
while (htmlAry[templdx] != NULL){
    // 첫 번째 태그 제외 나머지 태그의 여백 균일화 작업.
    if (0 < tagCnt) {
        printf(" ");
    }
    // Html의 모든 태그(비 태그 포함)가 홀수 형태일 때,
    if (wordCnt % 2 != 0) {
        if (wordCnt2 <= wordCnt / 2) {
            for (k = 0; k < wordCnt2; k++) {
                printf(" ");
            }
        }else if (wordCnt2 > wordCnt / 2) {
            for (k = 0; k < wordCnt - wordCnt2 - 1; k++) {
                printf(" ");
            }
        }
    }else{
        // Html의 모든 태그(비 태그 포함)가 짝수 형태일 때,
        if (wordCnt2 < wordCnt / 2) {
            for (k = 0; k < wordCnt2; k++) {
                printf(" ");
            }
        }else if (wordCnt2 >= wordCnt / 2) {
            for (k = 0; k < wordCnt - wordCnt2 - 1; k++) {
                printf(" ");
            }
        }
    }
}

// 1. '<'로 시작 ~ '>'로 끝날 때,
if (htmlAry[templdx] == '<') {
    while (1) {
        if (htmlAry[j] == '>') {
            tempWordSize++;
            printf("%c", htmlAry[j]);
        }
    }
}

```

```

        isBreak = 0;
        tagCnt++;
        wordCnt2++; // 단어 수에 포함 시키기1
        break;
    }else if (htmlAry[j] != '>') {
        tempWordSize++;
        printf("%c", htmlAry[j]);
    }
    templdx++;
    j++;
    tagIdx++;
}
/*printf("%d %d", templdx, j);*/
}else if (htmlAry[templdx] != '>' && htmlAry[templdx] != '<') {
    // 2. '>'로 시작 ~ '<'로 끝날 때 = '>'와 '<'로 시작하고 끝나지 않을
    때
    while (1) {
        if (htmlAry[j] == '>' || htmlAry[j] == '<') {
            /*tempWordSize++;*/
            isBreak = 1;
            wordCnt2++; // 단어 수에 포함시키기 2
            break;
        }else if (htmlAry[j] != '>' && htmlAry[j] != '<') {
            tempWordSize++;
            printf("%c", htmlAry[j]);
            isBreak = 0;
        }
        templdx++;
        j++;
    }
    /*printf("%d %d", templdx, j);*/
}
printf("Wn");

if (isBreak == 1) {
    continue;
}
templdx++;
j++;
tagIdx++;
}

free(htmlAry);

return 0;
}

```

[실행결과]

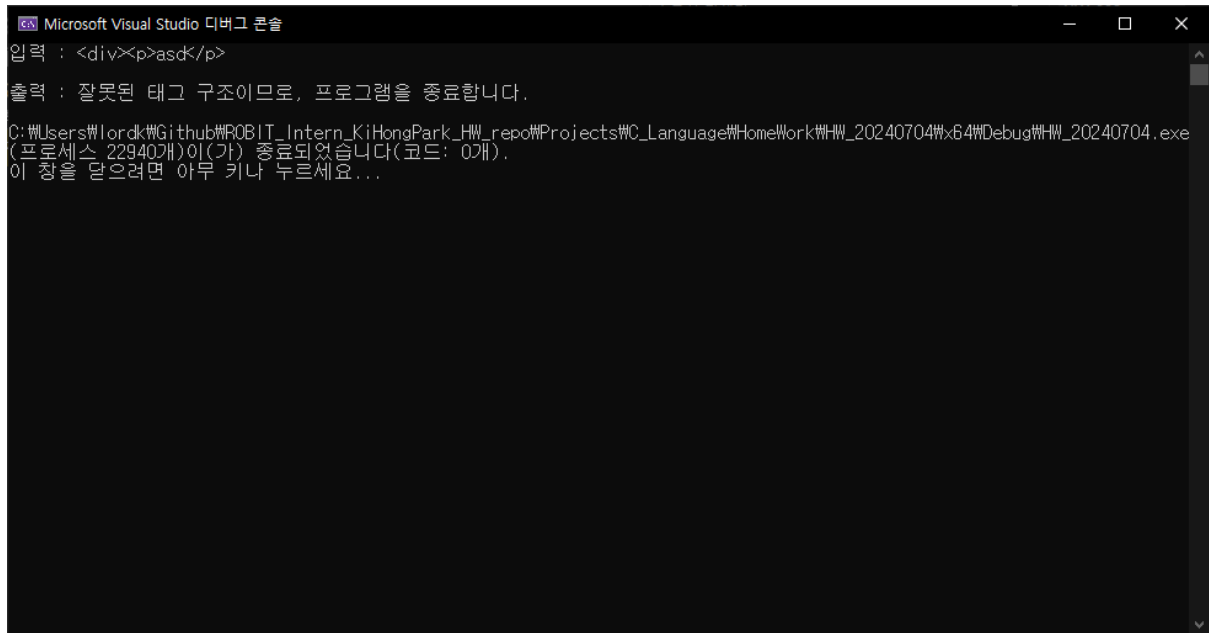
Test Case #1


```
Microsoft Visual Studio 디버깅 콘솔
입력 : <div><p>HELLO,<b>world</b>!</p></div>
출력 : <div>
      <p>
        HELLO,
        <b>
          world
        </b>
        !
      </p>
    </div>
C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\64\Debug\HW_20240704.exe
(프로세스 12908개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

Test Case #2

```
Microsoft Visual Studio 디버깅 콘솔
입력 : <div><p>HELLO,<b>world</b><b>world</b>!</p></div>
출력 : <div>
      <p>
        HELLO,
        <b>
          world
        </b>
        <b>
          world
        </b>
        !
      </p>
    </div>
C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\64\Debug\HW_20240704.exe
(프로세스 9544개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

Test Case #3



3. HW_003

** 이름 정렬하는 기능까지 구현하려고 시도하였으나, 해당 기능은 끝내 구현하지 못 했습니다.

밤을 세우며 여러 방안을 모색해도 이름 정렬 관련해서는 기능을 완벽하게 구현하지 못 했습니다. 이름 제외 성적의 순서를 완벽하게 구현한 소스코드를 첨부해드립니다.

이름 정렬을 시도한 코드는 다음 commit log에서 확인하실 수 있습니다.

https://github.com/kinesis19/ROBIT_Intern_KiHongPark_HW_repo/commit/88f063529577291953929c03f9c22a8b5847402f

[소스코드]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

double GettingDecimalPoint(int num1, int num2, int num3);

int main() {

    // 변수 선언하기.
    int inputNum;
    int i, j;

    // 형식 출력하기.
    printf("입력 : ");
    scanf_s("%d", &inputNum);
```

```

/* 학생 이름 배열, 성적 배열
* 이름의 길이는 안내되어 있지 않으므로, 임의로 지정함.
*/
char* chAry = (char*)calloc(inputNum, sizeof(char) * 100); // 이름 저장 배열
int* pAry1 = (int*)calloc(inputNum, sizeof(int)); // 성적 1 저장 배열
int* pAry2 = (int*)calloc(inputNum, sizeof(int)); // 성적 2 저장 배열
int* pAry3 = (int*)calloc(inputNum, sizeof(int)); // 성적 3 저장 배열
double* pArySort = (double*)calloc(inputNum, sizeof(double)); // 성적 정렬 배열
int* nameCntAry = (int*)calloc(inputNum, sizeof(int)); // 이름의 시작 index를 알기
위한 배열

```

```

int templdx = 0;
for (i = 0; i < inputNum; i++) {
    printf(" ");
    if (chAry[templdx] == NULL) {
        scanf("%s %d %d %d", &chAry[templdx], &pAry1[i], &pAry2[i],
&pAry3[i]);

```

```

        while (1) {
            if (chAry[templdx] == NULL) {
                nameCntAry[i] = templdx;
                break;
            }
            else {
                templdx++;
            }
        }
    }
}

```

```

// 배열에 할당하기.
for (i = 0; i < inputNum; i++) {
    double examAvg = GettingDecimalPoint(pAry1[i], pAry2[i], pAry3[i]);
    // 성적 정렬 배열에 값 할당하기 - 초기 할당
    pArySort[i] = examAvg;
}

```

```

// 배열 정렬하기.
for (i = 0; i < inputNum; i++) {
    // 성적 정렬하기
    for (j = 0; j < inputNum; j++) {
        if (j + 1 < inputNum && pArySort[j] < pArySort[j + 1]) {
            double tempNum;
            tempNum = pArySort[j];
            pArySort[j] = pArySort[j + 1];
            pArySort[j + 1] = tempNum;
        }
    }
}

```

```

// 정렬된 배열 출력하기

```

```

int tempCnt = 0;
printf("출력 : Wn");
for (i = 0; i < inputNum; i++) {
    printf(" ");
    for (j = tempCnt; j < nameCntAry[i]; j++) {
        printf("%c", chAry[j]);
    }
    while (1) {
        if (tempCnt < nameCntAry[i]) {
            tempCnt++;
        }
        else {
            break;
        }
    }
    printf(" ");
    // 출력하기
    if ((int)(pArySort[i] * 10) % 10 == 0) {
        printf("%dWn", (int)pArySort[i]);
    }
    else {
        printf("%.1lfWn", pArySort[i]);
    }
}

free(chAry);
free(pAry1);
free(pAry2);
free(pAry3);
free(pArySort);
free(nameCntAry);

return 0;
}

double GettingDecimalPoint(int num1, int num2, int num3) {
    /*
    * 소숫점 한 자리 출력이 형식임.
    * 그냥 계산하면 소숫점이 잘못 출력 되는 경우가 존재함.
    * 따라서, 소숫점 둘째 자리까지 구하고, 둘째 자리의 숫자만 평균에서 빼주도록 구현함.
    * e.x) 평균 : 50.66
    * 1. 50.66
    * 2. 506.6 <- 1
    * 3. 506.0
    * 4. 50.6 -> 1
    * 조정한 값 : 50.6 (-0.06)
    * 소수점 구하는 공식도 스스로 유추해 적용해서 뿌듯함.
    * 모든 과제는 규칙대로 인공지능 미사용, 인터넷 검색X, 레퍼런스 참고X,
    * 오직 강의 자료만을 참고하여 수행함.
    */
    double result = 0;
    result = (num1 + num2 + num3) / 3.0;
}

```

```

        result = (int)(result * 10) / 10.0;

        return result;
}

```

[실행결과]

Test Case #1



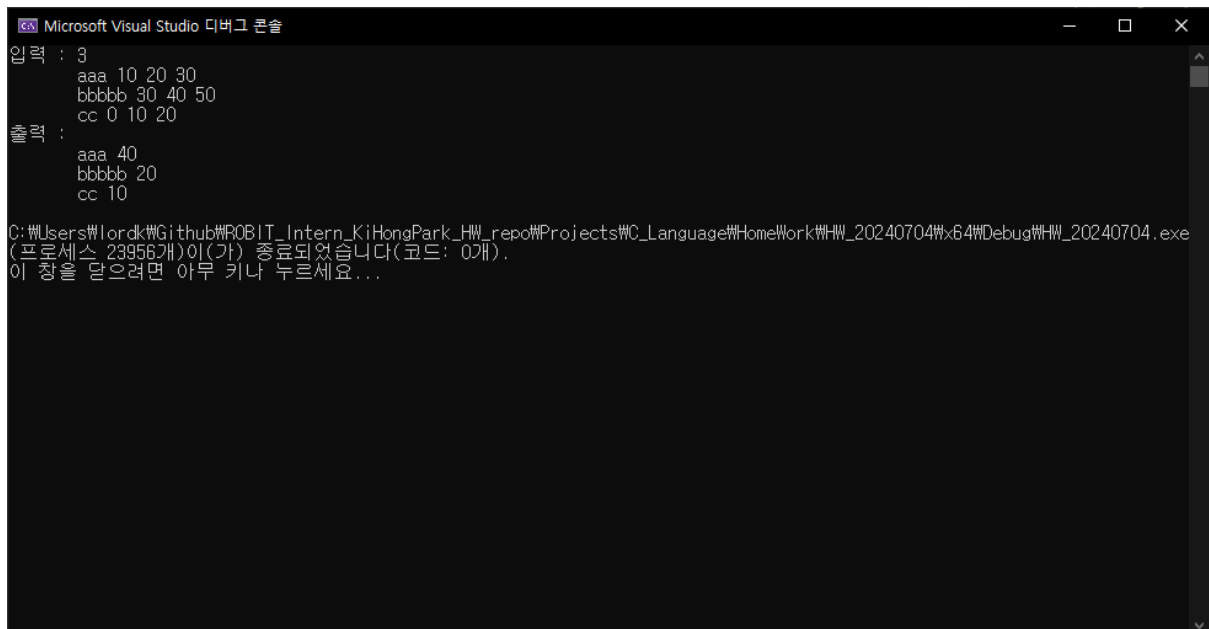
The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```

입력 : 5
      Amelia 40 65 49
      Grace 76 23 65
      Claire 34 54 23
      Bella 12 54 12
      Ellie 43 67 23
출력 :
      Amelia 54.6
      Grace 51.3
      Claire 44.3
      Bella 37
      Ellie 26
C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\Debug\HW_20240704.exe
(프로세스 3740개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

Test Case #2



The screenshot shows the Microsoft Visual Studio Debug Console with the following output:

```

입력 : 3
      aaa 10 20 30
      bbbb 30 40 50
      cc 0 10 20
출력 :
      aaa 40
      bbbb 20
      cc 10
C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\Debug\HW_20240704.exe
(프로세스 23956개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...

```

Test Case #3

```
Microsoft Visual Studio 디버그 콘솔
입력 : 2
AAAAA 9 8 7
BB 3 4 9
출력 :
AAAAA 8
BB 5.3
C:\Users\lordk\Github\ROBIT_Intern_KiHongPark_HW_repo\Projects\C_Language\HomeWork\HW_20240704\x64\Debug\HW_20240704.exe
(프로세스 26224개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```