

최범수 4일차 과제

1. HW_001

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    //동적할당을 위해 배열과 숫자 행, 열 입력받기
    int x, y;
    int** arr;
    int i, j;
    printf("입력 : ");
    scanf("%d %d", &x, &y);

    //숫자가 1일 때부터 계속 증가하게
    int num = 1;
    //동적할당
    arr = (int**)malloc(sizeof(int*) * x);
    //배열에 데이터 값 넣기
    for (i = 0; i < x; i++)
    {
        arr[i] = (int*)malloc(sizeof(int) * y);
    }
    //반복문을 이용해서 배열에 num값 넣기
    for (i = 1; i <= x; i++)
    {
        for (j = 1; j <= y; j++)
        {
            arr[i - 1][j - 1] = num;
            num++;
        }
    }
    /*
    for (i = 1; i <= x; i++)
    {
        for (j = 1; j <= y; j++)
        {
            arr[i - 1][j - 1];
        }
    }
    */
    //배열에 저장된 데이터 값 출력하는 부분
    for (i = 0; i < x; i++)
    {
        for (j = 0; j < y; j++)
        {
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
}
```

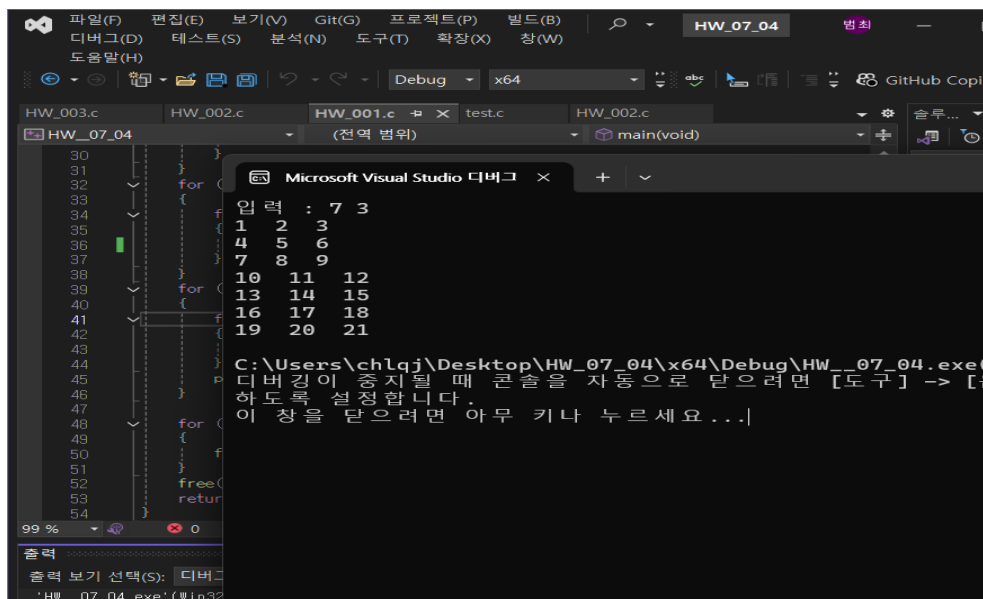
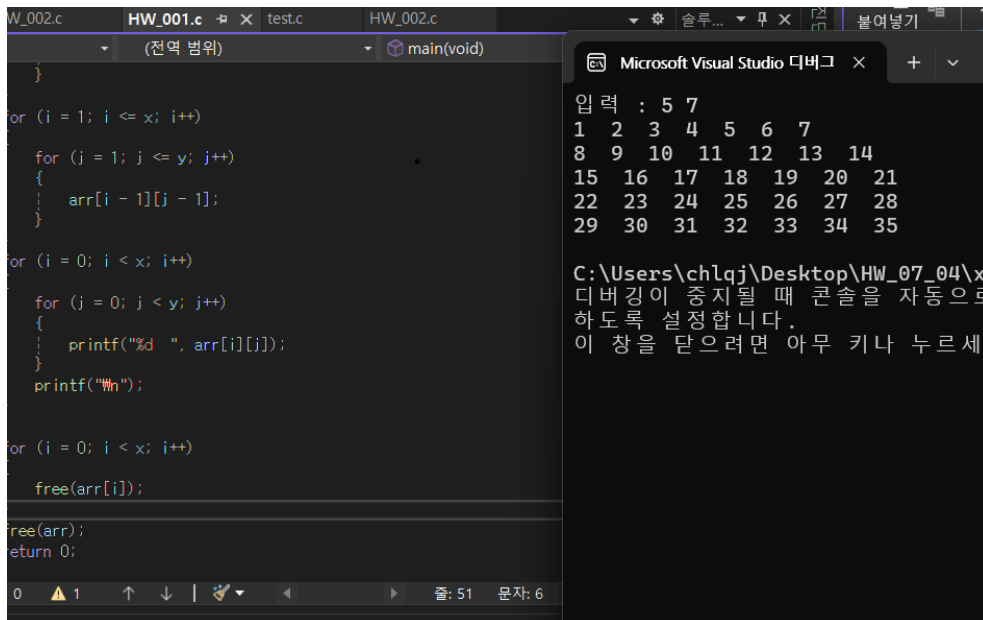
```

}

//다 쓴 배열에 데이터 값 제거해 주기
for (i = 0; i < x; i++)
{
    free(arr[i]);
}
free(arr);
return 0;

```

}첫번째 과제가 실행되지 못해서 노력한 부분까지 설명하겠습니다. 우선 동적할당을 사용해서 배열에 1부터 열과 행을 곱한 최대값 까지를 저장했고 그 후 순서대로 출력하는 부분까지 완성했습니다. 그리고 동적할당에 쓴 메모리를 초기화하기 위해서 free를 이용해서 지웠습니다. 1번 과제에서 제가 발견한 규칙은 처음에 아래로 1칸, 오른쪽 위로 1칸, 왼쪽 위로 1칸, 왼쪽 아래로 1칸, 그 후 아래로 1칸 까지 처음에는 5번 움직이고 그 다음 화살표는 8칸, 그리고 12칸을 움직여서 각 움직임의 차이가 1씩 증가한다는 것을 발견하였고, 이를 통해서 배열을 출력한 뒤 만약 배열의 끝부분이면 같은 줄에서 오른쪽으로 출력된 부분이 없을 때까지 이동하게 만들어야 한다는 부분까지 구상하였습니다.



2. HW_002

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>

char stack[10][10]; // 스택 만듦
int top = -1; // 가장 위를 top

//스택에 넣는 부분
void push(char* tag)
{
    if (top < 9)
    {
        top++;
        int i = 0;
        while (tag[i] != '\0' && i < 9)
        {
            stack[top][i] = tag[i];
            i++;
        }
        stack[top][i] = '\0';
    }
    //그렇지 않으면 오류가 뜨게 설정
    else
    {
        printf("Error\n");
    }
}

//스택에서 제거하는 부분
char* pop()
{
    if (top >= 0)
    {
        return stack[top--];
    }
    else
    {
        return NULL;
    }
}

// 가장 위 부분 확인
char* top_top()
{
    if (top >= 0)
    {
        return stack[top];
    }
    else
    {

```

```

        return NULL;
    }
}

// 스택이 비어있는지 확인하는 함수
int No_stack() {
    return top == -1;
}

int main()
{
    int **arr;
    int num;

    arr = (int**)malloc(sizeof(int*) * num);
    for (int i = 0; i < num; i++)
    {
        arr[i] = (int*)malloc(sizeof(int) * num);
    }
}

```

두번째 과제 역시 동적할당을 사용하여 스택을 구현하는 부분까지 완성했지만 문자열을 입력 받는 문제 조건과는 달리 우선 정수를 입력 받아 출력하고 그 뒤에 문자열로 고쳐서 해결하려 했지만 풀이과정에서 어려움을 겪어 HTML 태그의 중첩 구조를 시각적으로 표현하지 못했습니다. 코드 설명을 하자면 스택을 만들고 그 후 제일 위 부분을 측정하고 그 뒤에 스택에 자료를 넣는 push부분과 스택에 자료를 제거하는 pop부분을 만들었습니다. 그리고 가장 위 부분을 확인할 때 배열보다 하나 작은 숫자를 이용해서 top을 정의했고 스택이 비어 있는지는 top이 0보다 크거나 같다고 전에 정의했는데 만약 스택이 없는 경우에는 - 즉, 음수를 출력하게 하여 확인하려 했습니다.

3. HW_003

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>
#include <stdlib.h>

int main() {
    int num;
    printf("입력 : ");
    scanf("%d", &num);

    char* name[30];
    for (int i = 0; i < num; i++) {
        name[i] = (char*)malloc(30 * sizeof(char)); //위에서 배열을 30까지 만들어서 여기도 30
    }

    float* score = (float*)malloc(num * 3 * sizeof(float));
    float* aver = (float*)malloc(num * sizeof(float));

    // 학생 이름과 성적 3가지 입력 3개씩 들어오니까 3의 배수, +1, +2의 칸에 저장
    for (int i = 0; i < num; i++) {
        scanf("%s %f %f %f", name[i], &score[i * 3], &score[i * 3 + 1], &score[i * 3 + 2]);
        aver[i] = (score[i * 3] + score[i * 3 + 1] + score[i * 3 + 2]) / 3.0;
    } //성적 3개 평균을 aver에 저장

    // 학생 정렬
    for (int i = 0; i < num - 1; i++)
    {
        for (int j = 0; j < num - i - 1; j++)
        {
            if (aver[j] < aver[j + 1]) //평균이 작으면 위치 바꿔주고 이름도 바꿔주기 저장할
배열도 위치 바꿈
            {

                float tmp_aver = aver[j];
                aver[j] = aver[j + 1];
                aver[j + 1] = tmp_aver;
                // 평균 저장 위치 변경

                char* tmp_name = name[j];
                name[j] = name[j + 1];
                name[j + 1] = tmp_name;
                // 이름 위치 변경

                for (int k = 0; k < 3; k++)
                { // 점수 위치 변경 점수는 3개라서 for문 반복
                    float tmp_score = score[j * 3 + k];
                    score[j * 3 + k] = score[(j + 1) * 3 + k];
                    score[(j + 1) * 3 + k] = tmp_score;
                }
            }
        }
    }
}
```

```

    }
}

}

printf("출력 :Wn");
for (int i = 0; i < num; i++)
{
    printf("Wt%s %.1fWn", name[i], aver[i]);

    free(name[i]);
}
free(score);
free(aver);

return 0;
}

```

The screenshot shows the Microsoft Visual Studio IDE with a C program being debugged. The code in the editor defines a program that takes a number of students and their scores, calculates the average, and prints the results. The debug console shows the input and output of the program.

Code in Editor:

```

1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4  #include <stdlib.h>
5
6
7  int main() {
8      int num;
9      printf("입력 : ");
10     scanf("%d", &num);
11
12
13
14     char* name[30];
15     for (int i = 0; i < num; i++) {
16         name[i] = (char*)malloc(30 * sizeof(char)); //위에서
17     }
18
19
20     float* score = (float*)malloc(num * 3 * sizeof(float));
21     float* aver = (float*)malloc(num * sizeof(float));
22
23     // 학생 이름과 성적 3가지 입력 3개씩 들어오니까 3의 배수
24     for (int i = 0; i < num; i++) {
25         scanf("%s %f %f", name[i], &score[i * 3], &score[

```

Debug Console Output:

```

입력 : 5
Amelia 40 63 49
Grace 76 23 65
Claire 34 54 23
Bella 12 54 12
Ellie 43 67 23
출력 :
Grace 54.7
Amelia 50.7
Ellie 44.3
Claire 37.0
Bella 26.0

```

Path and Message:

```

C:\Users\chlqj\Desktop\HW_07_04\x64\Debug\HW
디버깅이 중지될 때 콘솔을 자동으로 닫으려
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...

```

