

CHAPTER 10

동적 메모리 할당의 이해
및 응용 실습

실습 1

문제 1)

배열의 길이 n 을 입력받고 n 개의 정수를 배열에 저장 후 출력하기

Microsoft Visual Studio 디버그 콘솔

- □ X

원하는 배열의 길이 입력: 4

4개의 정수 입력: 1 2 3 4

입력 받은 정수: 1 2 3 4

실습 1

정답

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

// 배열의 길이 n을 입력받고 n개의 정수를 배열에 저장 후 출력하기
int main() {
    int n; // length
    int* arr; // pointer for dynamic array

    // input length
    printf("원하는 배열의 길이 입력: ");
    scanf("%d", &n);
    // allocate dynamic memory
    arr = (int*)malloc(sizeof(int) * n);

    printf("n개의 정수 입력: ");
    // input values
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("입력 받은 정수: ");
    // print values
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);

    // deallocating the memory
    free(arr);
}
```

실습 2

문제 2)

배열의 길이 n 을 입력받고 n 개의 정수를 배열에 저장 후 총합을 구하기



```
Microsoft Visual Studio 디버그 콘솔
Enter number of elements: 3
Enter elements: 1 2 3
Sum = 6
```

실습 2

정답

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

// n개의 정수를 배열에 저장 후 총합을 구하기

int main() {
    int n, i, * ptr, sum = 0;

    printf("Enter number of elements: ");
    scanf("%d", &n);

    ptr = (int*)malloc(n * sizeof(int));

    // if memory cannot be allocated
    if (ptr == NULL) {
        printf("Error! memory not allocated.");
        exit(0);
    }
    // input values and update sum
    printf("Enter elements: ");
    for (i = 0; i < n; ++i) {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }

    printf("Sum = %d", sum);

    // deallocating the memory
    free(ptr);

    return 0;
}
```

실습 3

문제 3)

배열의 길이 $n1$ 을 입력받고 동적할당한 배열의 주소값을 출력 후, 새로운 길이 $n2$ 를 입력받아 배열을 $n2$ 의 길이로 재할당 후 변경된 배열의 주소값을 출력하시오

Microsoft Visual Studio 디버그 콘솔

```
길이 입력: 3  
할당된 메모리의 주소값 출력:  
0000019EE4A9D510  
0000019EE4A9D514  
0000019EE4A9D518  
  
새로운 길이 입력: 5  
새로 할당된 메모리 주소 입력:  
0000019EE4A9D790  
0000019EE4A9D794  
0000019EE4A9D798  
0000019EE4A9D79C  
0000019EE4A9D7A0
```

실습 3

정답

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

//동적할당한 배열의 주소값을 출력하고
//새로운 길이를 입력받아서 변경된 길이로 배열의 길이를 변경 후 변경된 배열의 주
소값 출력
int main() {
    // ptr: array, i: index, n1: first length, n2: second length
    int* ptr, i, n1, n2;
    printf("길이 입력: ");
    scanf("%d", &n1);

    ptr = (int*)malloc(n1 * sizeof(int));

    printf("할당된 메모리의 주소값 출력:\n");
    for (i = 0; i < n1; ++i)
        printf("%pc\n", ptr + i);

    printf("\n새로운 길이 입력: ");
    scanf("%d", &n2);

    // relocating the memory
    ptr = (int*)realloc(ptr, n2 * sizeof(int));

    printf("새로 할당된 메모리 주소 입력:\n");
    for (i = 0; i < n2; ++i)
        printf("%pc\n", ptr + i);

    free(ptr);

    return 0;
}
```

실습 4

문제 4)

크기가 n 인 배열에 1부터 n 까지 값을 저장 후, n 보다 작은 새로운 길이 $n1$ 을 입력받고 $n1$ 개의 값을 배열에서 순서대로 출력하시오

Microsoft Visual Studio 디버그 콘솔

```
Enter number of elements: 5  
The elements of the array are: 1, 2, 3, 4, 5,
```

```
Enter the new size of the array: 3  
The elements of the array are: 1, 2, 3,
```


실습 4

정답

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

// 크기가 n인 배열에 n개의 값을 저장 후 배열의 사이즈를 줄인 후 배열의 값을 모두 출력

```
int main()
{
    // This pointer will hold the
    // base address of the block created
    int* ptr;
    int n, i;

    // Get the number of elements for the array

    printf("Enter number of elements: ");
    scanf("%d", &n);
    // Dynamically allocate memory using calloc()
    ptr = (int*)calloc(n, sizeof(int));

    // Check if the memory has been successfully
    // allocated by malloc or not
    if (ptr == NULL) {
        printf("Memory not allocated.\n");
        exit(0);
    }
    else {

        // Memory has been successfully allocated
        //printf("Memory successfully allocated using calloc.\n");

        // Get the elements of the array
        for (i = 0; i < n; ++i) {
            ptr[i] = i + 1;
        }
    }
}
```

```
// Get the new size for the array

printf("\n\nEnter the new size of the array: ");
scanf("%d", &n);
// Dynamically re-allocate memory using realloc()
ptr = (int*)realloc(ptr, n * sizeof(int));

// Memory has been successfully allocated
// printf("Memory successfully re-allocated using realloc.\n");

// Print the elements of the array
printf("The elements of the array are: ");
for (i = 0; i < n; ++i) {
    printf("%d, ", ptr[i]);
}

free(ptr);

return 0;
```

실습 5

문제 5)

문자열의 최대 길이 n 을 입력받고 길이가 n 이하인 문자열을 입력받고 출력하기 (힌트: 문자열의 끝은 '\0' 이다.)

A screenshot of the Microsoft Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio 디버그 콘솔". The console output shows three lines: "Enter limit of the text: 20", "Enter text: Kwangwoon SW !!", and "Inputted text is: Kwangwoon SW !!". The text is displayed in a light green monospace font on a black background. The window has standard Windows controls (minimize, maximize, close) in the top right corner.

```
Microsoft Visual Studio 디버그 콘솔
Enter limit of the text: 20
Enter text: Kwangwoon SW !!
Inputted text is: Kwangwoon SW !!
```

실습 5

정답

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

// 문자열의 최대 길이 n을 입력받고 길이가 n 이하인 문자열을 입력받고 출력하기

int main()
{
    int n;
    char* text;

    // input the length of text
    printf("Enter limit of the text: ");
    scanf("%d", &n);

    /*allocate memory dynamically*/
    text = (char*)malloc(n * sizeof(char));

    // input text
    printf("Enter text: ");
    getchar(); /*clear input buffer*/
    gets(text);

    printf("Inputted text is: %s\n", text);

    /*Free Memory*/
    free(text);

    return 0;
}
```

실습 6

문제 6)

크기가 n 인 배열에 1부터 n 까지 값을 저장 후, n 보다 큰 새로운 길이 $n1$ 을 입력받고 $n1$ 개의 값을 배열에서 순서대로 출력하시오
(힌트: 기존의 배열 뒤에 값을 추가)



```
Select Microsoft Visual Studio 디버그 콘솔
Enter number of elements: 5
The elements of the array are: 1, 2, 3, 4, 5,
Enter the new size of the array: 10
The elements of the array are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

실습 6

정답

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

// 크기가 n인 배열에 n개의 값을 저장 후 배열의 사이즈를 늘린 후 늘어난 만큼의 값을 원래 배열 뒤에 추가로 출력

```
int main()
{
    // This pointer will hold the
    // base address of the block created
    int* ptr;
    int n1, n2, i; // n1: first length, n2: second length, i: index

    // Get the number of elements for the array

    printf("Enter number of elements: ");
    scanf("%d", &n1);
    // Dynamically allocate memory using calloc()
    ptr = (int*)calloc(n1, sizeof(int));

    // Check if the memory has been successfully
    // allocated by malloc or not
    if (ptr == NULL) {
        printf("Memory not allocated.\n");
        exit(0);
    }
    else {
        // Memory has been successfully allocated
        // printf("Memory successfully allocated using calloc.\n");

        // Get the elements of the array
        for (i = 0; i < n1; ++i) {
            ptr[i] = i + 1;
        }

        // Print the elements of the array
        printf("The elements of the array are: ");
        for (i = 0; i < n1; ++i) {
            printf("%d, ", ptr[i]);
        }
    }
}
```

```

    // Get the new size for the array
    ;
    printf("\n\nEnter the new size of the array: ");
    scanf("%d", &n2);
    // Dynamically re-allocate memory using realloc()
    ptr = (int*)realloc(ptr, n2 * sizeof(int));

    // Memory has been successfully allocated
    // printf("Memory successfully re-allocated using realloc.\n");

    // Get the new elements of the array
    for (i = n1; i < n2; ++i) {
        ptr[i] = i + 1;
    }

    // Print the elements of the array
    printf("The elements of the array are: ");
    for (i = 0; i < n2; ++i) {
        printf("%d, ", ptr[i]);
    }

    free(ptr);

    return 0;
}
```

실습 7

문제 7)

크기가 n인 배열을 선언하고 n개의 정수를 저장 후

`void plus_one(int* arr, int n)` // n은 배열의 길이, arr은 동적배열
함수를 만들어서 배열의 값을 1씩 증가시킨 후 출력



```
Microsoft Visual Studio 디버그 콘솔
배열의 길이 입력: 3
배열의 값 입력: 1 2 3
2 3 4
```

실습 7

정답

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

// 함수를 하나 만들어서 배열의 모든 값을 +1 한 결과를 출력해보자

```
void plus_one(int* arr, int n) {
    for (int i = 0; i < n; i++)
        arr[i] += 1;
}
```

```
int main() {
    int* arr;
    int n;

    printf("배열의 길이 입력: ");
    scanf("%d", &n);
    arr = (int*)malloc(n * sizeof(int));
```

```
    printf("배열의 값 입력: ");
    for (int i = 0; i < n; i++)
        scanf("%d", &arr[i]);
```

```
    // plus one each value of array
    plus_one(arr, n);
```

```
    for (int i = 0; i < n; i++)
        printf("%d ", arr[i]);
```

```
    // deallocationg dynamic memory
    free(arr);
    return 0;
```

```
}
```

실습 8

문제 8)

$n \times n$ 크기의 0으로 초기화된 행렬을 만들고 반지름 r 을 입력받은 후 행렬의 중심으로부터 r 이내의 값들을 1로 변경 하기

Hint: math.h 라이브러리의 pow와 sqrt 함수 사용

`double pow(double base, double power);` // base: 밑, power: 지수

`double sqrt(double arg);` // arg: 값

Microsoft Visual Studio 디버그 콘솔

```
input your width: 9
input your radius: 3
0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 0 0
0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0
```


실습 8

정답

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

// Make HEIGHTxWIDTH matrix and draw a circle
int main() {
    int** mat;
    int r;
    int width, height;
    float distance = 0;

    printf("input your height and width: ");
    scanf("%d", &width); // input width , test: 9

    height = width; // n x n matrix

    printf("input your radius: ");
    scanf("%d", &r);

    // allocate each row of matrix
    mat = (int**)malloc(height * sizeof(int*));

    int centerX, centerY;
```

```
// calculate center point
centerX = (height + 1) / 2 - 1;
centerY = (width + 1) / 2 - 1;

// allocate each column of matrix
for (int i = 0; i < width; i++)
    mat[i] = (int*)malloc(width * sizeof(int));

// initialize all values of matrix to 0
for (int i = 0; i < height; i++)
    for (int j = 0; j < width; j++)
        mat[i][j] = 0;

// fill circle on matrix
for (int i = 0; i < height; i++) {
    for (int j = 0; j < width; j++) {
        // calculate distance using radius and center point
        distance = sqrt(pow(centerX - j, 2) + pow(centerY - i, 2));
        // printf("(%d,%d) %f\n",i,j,distance);
        if (distance <= r)
            mat[i][j] = 1;
    }
}

// print matrix
for (int i = 0; i < height; i++) {
    for (int j = 0; j < width; j++) {
        printf("%d ", mat[i][j]);
    }
    printf("\n");
}

for (int i = 0; i < height; i++)
    free(mat[i]);
free(mat);
return 0;
}
```

실습 9

문제 9)

$n \times n$ 크기의 0으로 초기화된 행렬을 만들고 달팽이 모양 배열을 출력해보시오.



```
Microsoft Visual Studio 디버그 콘솔
배열의 사이즈를 입력해주세요: 5
1  2  3  4  5
16 17 18 19 6
15 24 25 20 7
14 23 22 21 8
13 12 11 10 9
```

실습 9

정답

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

// 행렬의 크기 N을 입력받고 NxN 크기의 달팽이 행렬을 출력해보세요.
int main() {
    int** arr; // 2차원 배열
    int size; // 배열의 행, 열 사이즈

    printf("배열의 사이즈를 입력해주세요: ");
    scanf("%d", &size);

    // 2차원 배열의 값을 0으로 동적할당 후 초기화
    arr = (int**)malloc(sizeof(int*) * size);
    for (int i = 0; i < size; i++) {
        arr[i] = (int*)calloc(size, sizeof(int));
    }

    int row = 0; // row 위치
    int col = -1; // col 위치
    int step = 1; // row, col 위치 이동
    int num = 0; // 저장할 값
    int len = size; // 출력을 위해 size 저장

    while (1) {
        // 열을 이동하면서 값 저장
        for (int i = 0; i < size; i++) {
            num += 1;
            col += step;
            arr[row][col] = num;
        }
        size -= 1;
        // 값 저장이 완료되면 탈출
        if (size < 1)
            break;
        // 행을 이동하면서 값 저장
        for (int i = 0; i < size; i++) {
            num += 1;
            row += step;
            arr[row][col] = num;
        }
    }
}
```

```
    }
    // 증감 방향을 반대로
    step = -step;
}

// 2차원 배열 달팽이 배열 출력하기
for (int i = 0; i < len; i++) {
    for (int j = 0; j < len; j++) {
        printf("%4d", arr[i][j]);
    }
    printf("\n");
}
return 0;
}
```