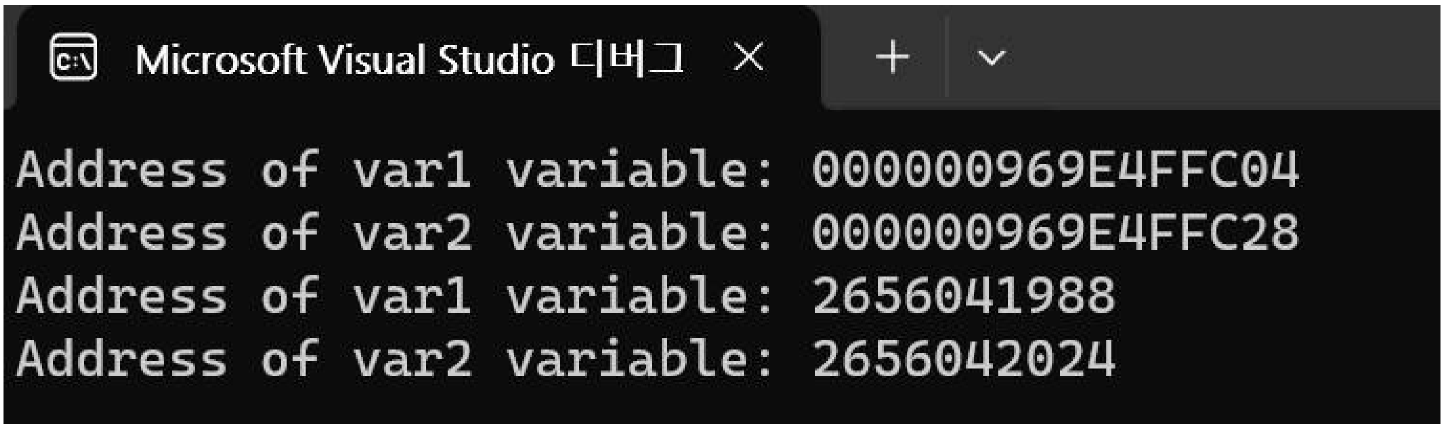
2019313550_박병현

실습 1.

```c
#include <stdio.h>

int main() {
	int var1;
	char var2[10];
	// %u: 부호없는 10진수로 출력, %p: 포인터의 주소를 출력
	printf("Address of var1 variable: %p\n", &var1);
	printf("Address of var2 variable: %p\n", &var2);

	printf("Address of var1 variable: %u\n", &var1);
	printf("Address of var2 variable: %u\n", &var2);

	return 0;
}
```

실습 1 실행 화면.



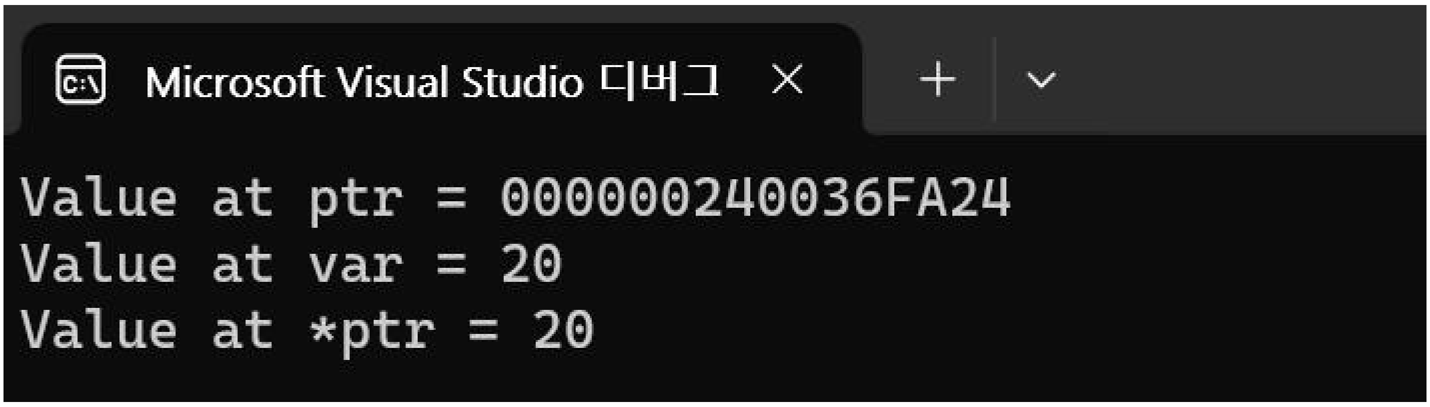
```
Address of var1 variable: 000000969E4FFC04
Address of var2 variable: 000000969E4FFC28
Address of var1 variable: 2656041988
Address of var2 variable: 2656042024
```

2019313550_박병현

실습 2.

```c
#include <stdio.h>

int main() {
	int var = 20;
	//declare pointer variable
	int* ptr;

	// note that data type of ptr and var must be same
	ptr = &var;

	// assign the address of a vatiable to a pointer
	printf("Value at ptr = %p \n", ptr);
	printf("Value at var = %d \n", var);
	printf("Value at *ptr = %d \n", *ptr);

	return 0;
}
```

실습 2 실행 화면.



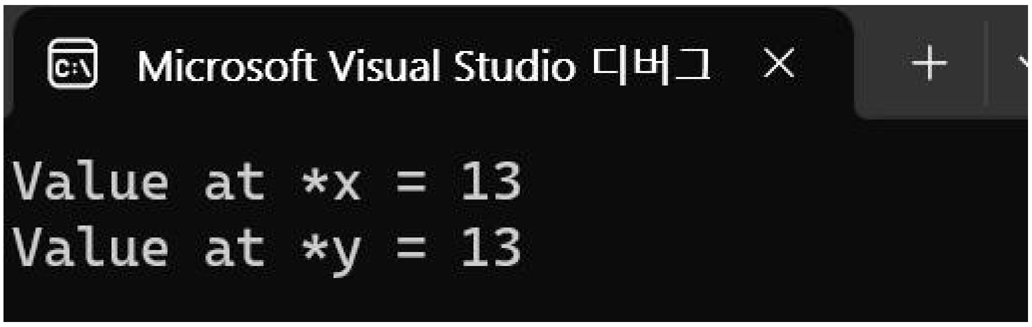
```
Value at ptr = 000000240036FA24
Value at var = 20
Value at *ptr = 20
```

2019313550_박병현

실습 3.

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

int main() {
        int* x; // Allocate the pointers x and y
        int* y; // (but not the pointees)

        x = malloc(sizeof(int)); // Allocate an in pointee,
        // and set x to point to it

        *x = 42; // Dereference x to store 42 in its pointee
        //*y = 13; // CRASH -- y doss not have a pointee yet
        y = x; // Pointer assignment sets y to point to x's pointee
        *y = 13; // Dereference y to store 13 in its (shared) pointee

        printf("Value at *x = %d \n", *x);
        printf("Value at *y = %d \n", *y);

        return 0;
}
```

실습 3 실행 화면.



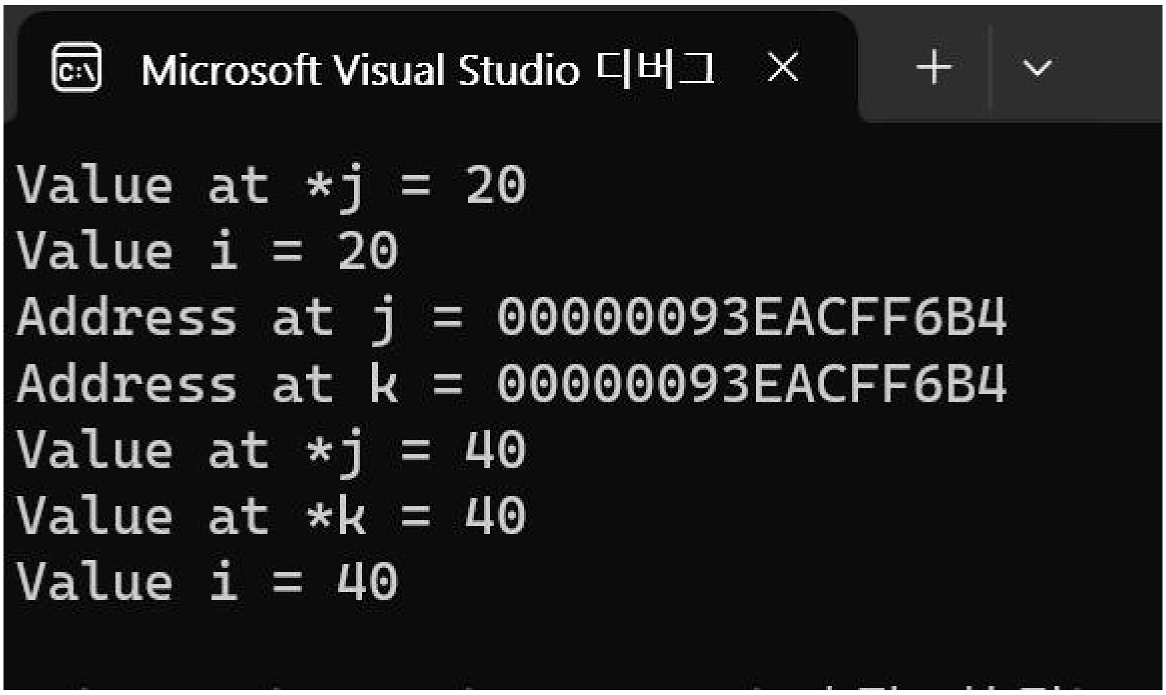
```
Value at *x = 13
Value at *y = 13
```

2019313550_박병현

실습 4.

```c
#include <stdio.h>

int main() {
    int i = 10;
    int* j = &i;
    int* k;

    /* Assign to what j points to: */
    *j = 20; /* Now i is 20. */
    printf("Value at *j = %d \n", *j);
    printf("Value i = %d \n", i);
    printf("Address at j = %p \n", j);

    /* Assign j to k: */
    k = j; /* Now k pints to i too. */
    printf("Address at k = %p \n", k);

    /* Assign to what j points to: */
    *j = *k + i; /* Now i is 40. */
    printf("Value at *j = %d \n", *j);
    printf("Value at *k = %d \n", *k);
    printf("Value i = %d \n", i);

    return 0;
}
```

실습 4 실행 화면.



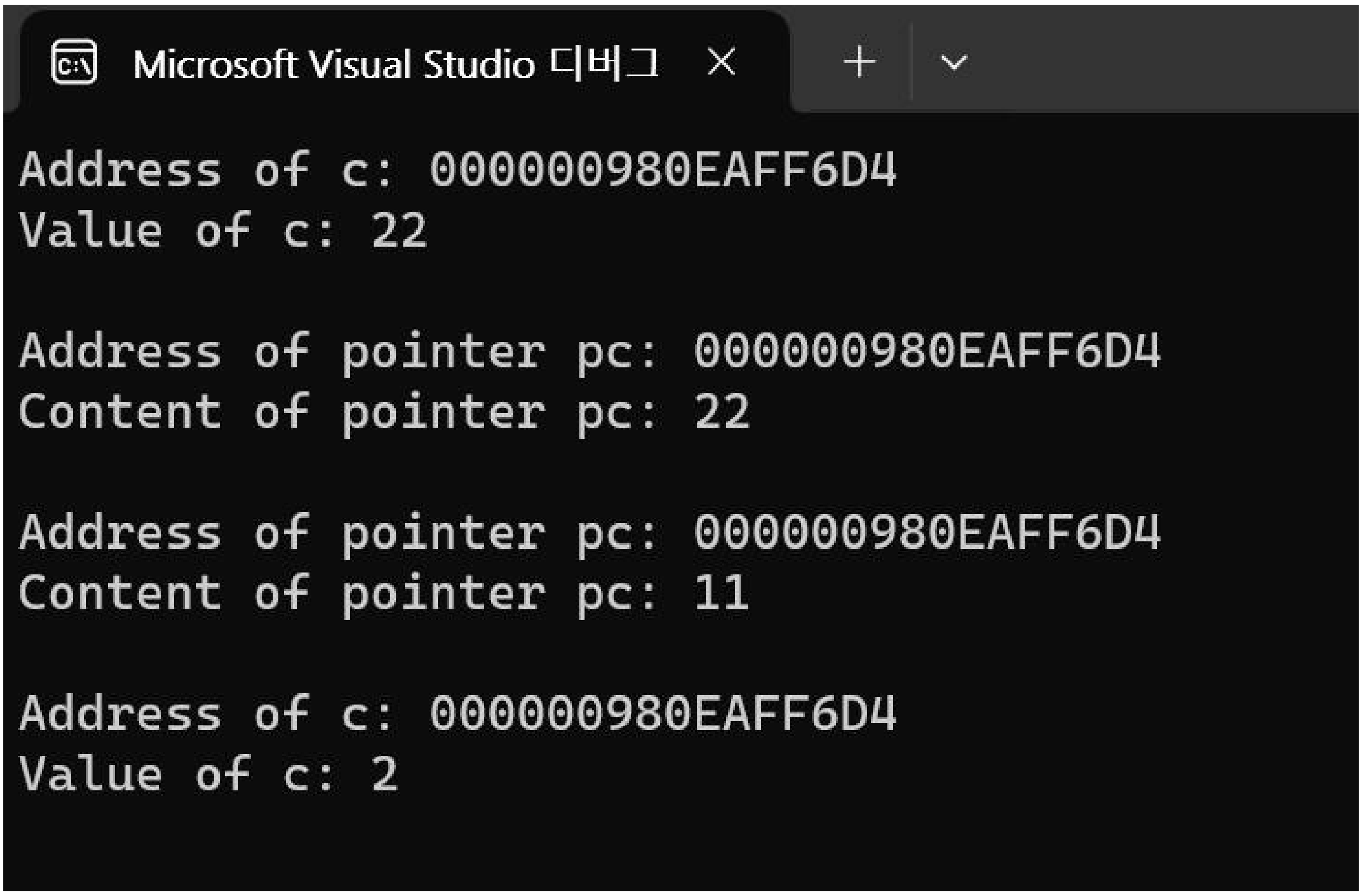
```
Value at *j = 20
Value i = 20
Address at j = 00000093EACFF6B4
Address at k = 00000093EACFF6B4
Value at *j = 40
Value at *k = 40
Value i = 40
```

2019313550_박병현

실습 5.

```c
#include <stdio.h>

int main() {
    int* pc, c;

    c = 22;
    printf("Address of c: %p\n", &c);
    printf("Value of c: %d\n\n", c); // 22

    pc = &c;
    printf("Address of pointer pc: %p\n", pc);
    printf("Content of pointer pc: %d\n\n", *pc); // 22

    c = 11;
    printf("Address of pointer pc: %p\n", pc);
    printf("Content of pointer pc: %d\n\n", *pc); // 11

    *pc = 2;
    printf("Address of c: %p\n", &c);
    printf("Value of c: %d\n\n", c); //2
    return 0;
}
```

실습 5 실행 화면.



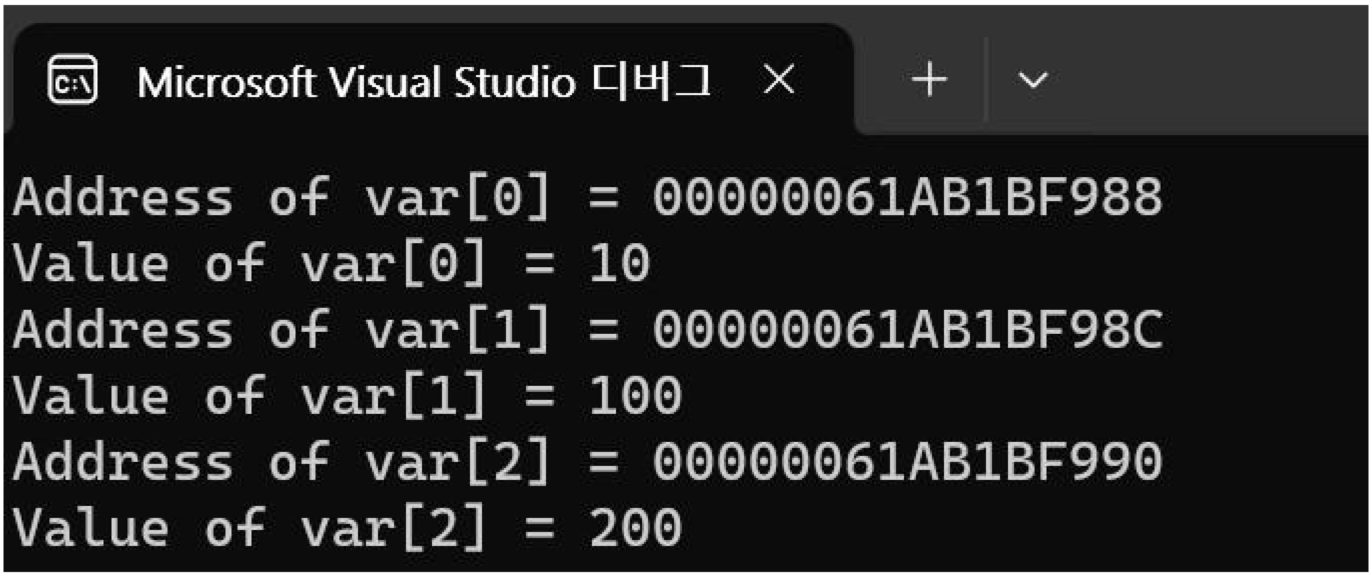
```
Address of c: 000000980EAFF6D4
Value of c: 22

Address of pointer pc: 000000980EAFF6D4
Content of pointer pc: 22

Address of pointer pc: 000000980EAFF6D4
Content of pointer pc: 11

Address of c: 000000980EAFF6D4
Value of c: 2
```

2019313550_박병현

실습 6.

```
#include <stdio.h>

const int MAX = 3;

int main() {
    int var[] = { 10, 100, 200 };
    int i, * ptr;

    /* let us have array address in pointer */
    ptr = var;

    for (i = 0; i < MAX; i++) {
        printf("Address of var[%d] = %p\n", i, ptr);
        printf("Value of var[%d] = %d\n", i, *ptr);
        /* move to the next location */
        ptr++;
    }
    return 0;
}
```

실습 6 실행 화면.



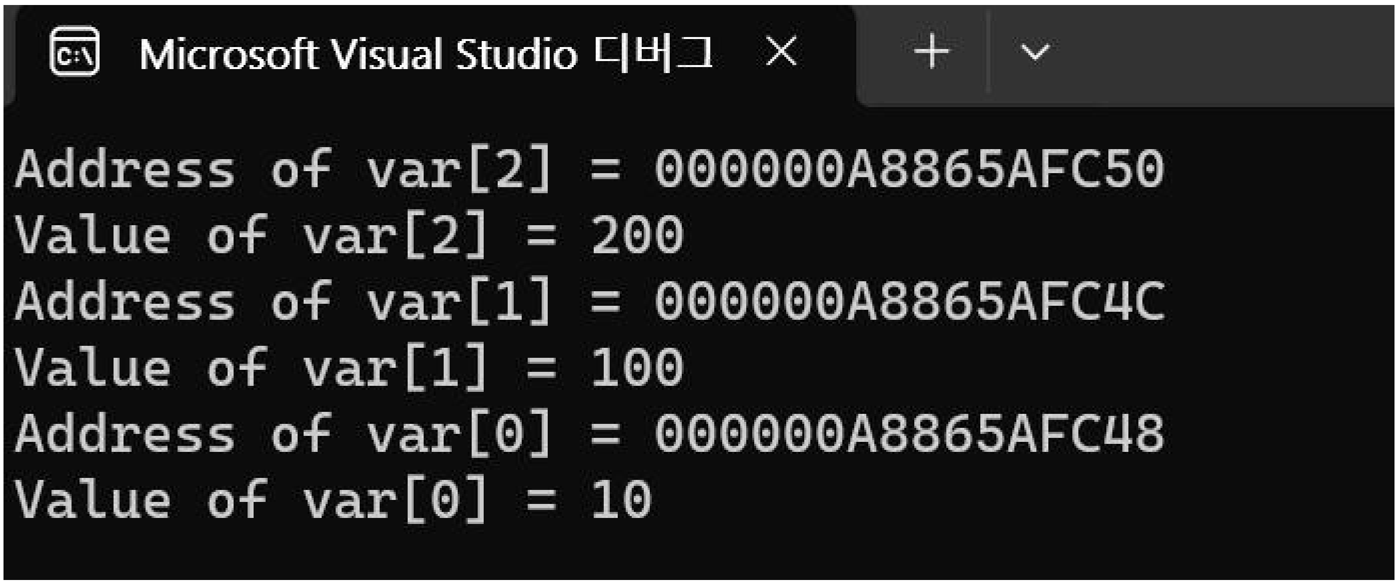
```
Address of var[0] = 00000061AB1BF988
Value of var[0] = 10
Address of var[1] = 00000061AB1BF98C
Value of var[1] = 100
Address of var[2] = 00000061AB1BF990
Value of var[2] = 200
```

2019313550_박병현

실습 7.

```
#include <stdio.h>

const int MAX = 3;

int main() {
    int var[] = { 10, 100, 200 };
    int i, * ptr;

    /* let us have array address in pointer */
    ptr = &var[MAX - 1];

    for (i = MAX; i > 0; i--) {
        printf("Address of var[%d] = %p\n", i - 1, ptr);
        printf("Value of var[%d] = %d\n", i - 1, *ptr);
        /* move to the previous location */
        ptr--;
    }
    return 0;
}
```

실습 7 실행 화면.



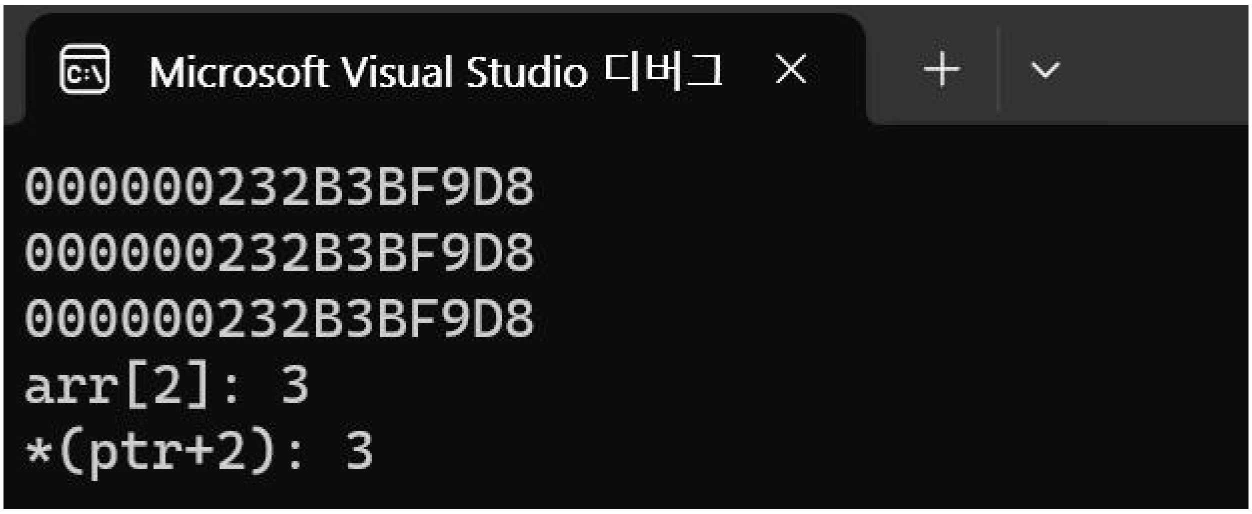
```
Address of var[2] = 000000A8865AFC50
Value of var[2] = 200
Address of var[1] = 000000A8865AFC4C
Value of var[1] = 100
Address of var[0] = 000000A8865AFC48
Value of var[0] = 10
```

2019313550_박병현

실습 9.

```c
#include <stdio.h>

int main() {
    int arr[5] = { 1, 2, 3, 4, 5 };
    int* ptr = arr;

    printf("%p\n", ptr);
    printf("%p\n", arr);
    printf("%p\n", &arr[0]);

    printf("arr[2]: %d\n", arr[2]);
    printf("*(ptr+2): %d\n", *(ptr + 2));

    return 0;
}
```

실습 9 실행 화면.



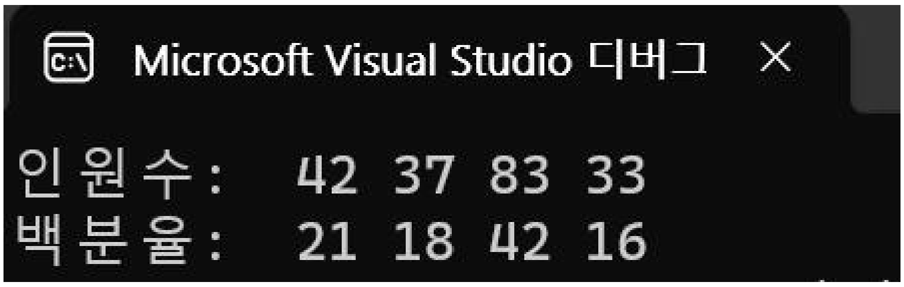
```
000000232B3BF9D8
000000232B3BF9D8
000000232B3BF9D8
arr[2]: 3
*(ptr+2): 3
```

2019313550_박병현

실습 11.

```
#include <stdio.h>
#define N 4

void print_arr(int* arr); // void print_arr(int arr[N]);
void percentage(int* arr); // void percentage(int arr[N]);

int main() {
    int count[N] = { 42, 37, 83, 33 };
    printf("인원수: ");
    print_arr(count); // count 배열을 전달해 출력하기
    percentage(count); // count 배열을 전달해 백분율로 변환하기
    printf("\n백분율: ");
    print_arr(count); // count 배열을 전달해 출력하기

    return 0;
}

void print_arr(int* arr) { // void print_arr(int arr[N]);
    int i;
    for (i = 0; i < N; i++)
        printf("%3d", *(arr + i));
}
void percentage(int* arr) { // void percentage(int arr[N]);
    int i, total = 0;
    for (i = 0; i < N; i++)
        total += *(arr + i);
    for (i = 0; i < N; i++)
        *(arr + i) = (int)((double)*(arr + i) / total * 100);
    // arr[i] = (int) ((double) arr[i] / total * 100);
}
```

실습 11 실행 화면.



```
인원수:  42 37 83 33
백분율:  21 18 42 16
```