

C 기초 프로그래밍 교육

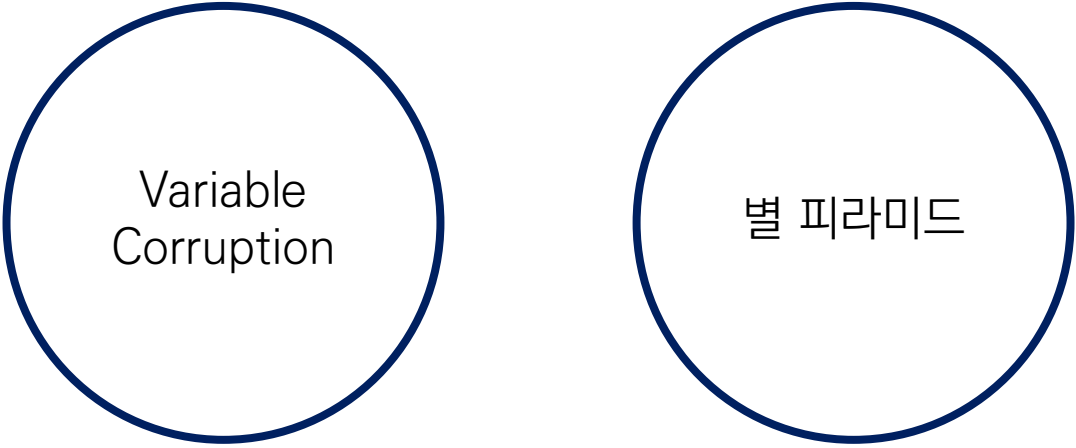
Week 4. pointer, function

교육 구성

INDEX

-
1. 과제 풀이
 2. 함수
 3. 포인터
-

1. 과제 풀이



Variable
Corruption

별 피라미드

과제 1 – Variable Corruption

```
// gcc -o Variable-Corruption Variable-Corruption.c -m32 -mpreferred-stack-boundary=2 -fno-stack-protector

#include <stdio.h>

int main()
{
    char type = 'A';
    char input[16];

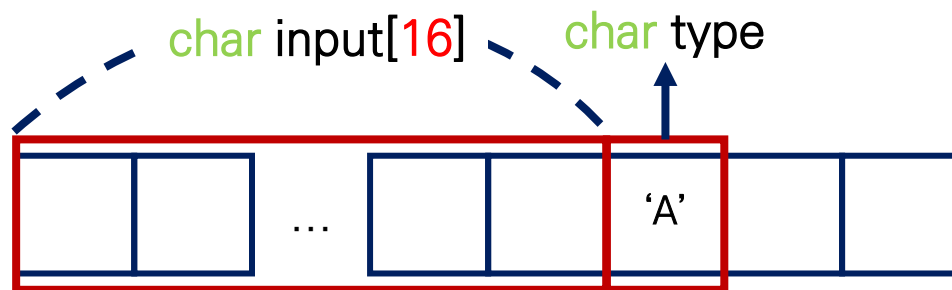
    printf("input : ");
    scanf("%s", input);

    if(type == 'A') {
        printf("type is %c, you lose!\n", type);
    } else if(type == 'B') {
        printf("type is %c, you WIN!!\n", type);
    }
}
```

```
minibee@cargos-edu:~/cedu/week3/hw$ ./Variable-Corruption
input : 
type is B, you WIN!!
```

적절한 입력 값을 넣어 “you WIN”이 출력되도록 만들기

과제 1 – Variable Corruption



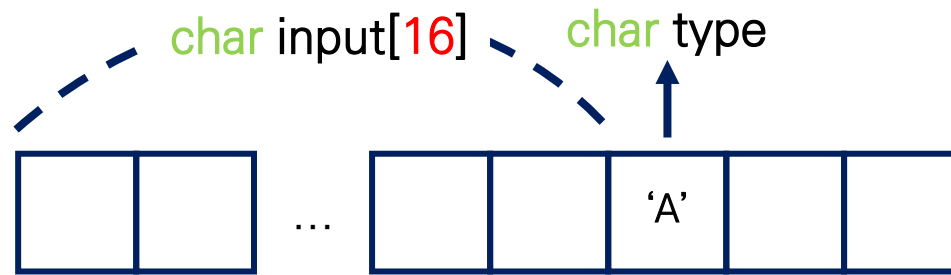
```
// gcc -o Variable-Corruption Variable-Corruption.c -m32 -mpreferred-stack-boundary=2 -fno-stack-protector
#include <stdio.h>

int main()
{
    char type = 'A';
    char input[16];

    printf("input : ");
    scanf("%s", input);

    if(type == 'A') {
        printf("type is %c, you lose!\n", type);
    } else if(type == 'B') {
        printf("type is %c, you WIN!!\n", type);
    }
}
```

과제 1 – Variable Corruption



```
// gcc -o Variable-Corruption Variable-Corruption.c -m32 -mpreferred-stack-boundary=2 -fno-stack-protector
#include <stdio.h>

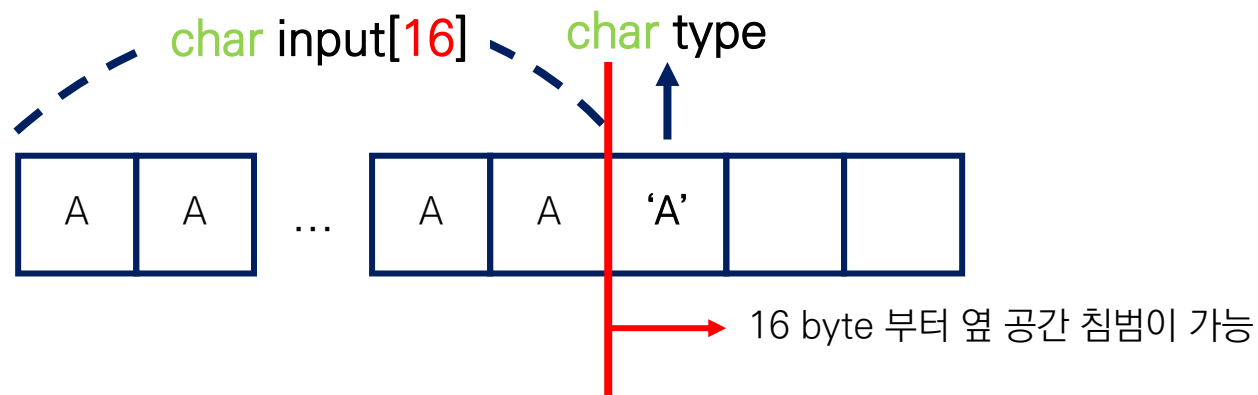
int main()
{
    char type = 'A';
    char input[16];

    printf("input : ");
    scanf("%s", input);

    if(type == 'A') {
        printf("type is %c, you lose!\n", type);
    } else if(type == 'B') {
        printf("type is %c, you WIN!!\n", type);
    }
}
```

Buffer Overflow 버그 발생!

과제 1 – Variable Corruption



```
// gcc -o Variable-Corruption Variable-Corruption.c -m32 -mpreferred-stack-boundary=2 -fno-stack-protector
#include <stdio.h>

int main()
{
    char type = 'A';
    char input[16];

    printf("input : ");
    scanf("%s", input);
    if(type == 'A') {
        printf("type is %c, you lose!\n", type);
    } else if(type == 'B') {
        printf("type is %c, you WIN!!\n", type);
    }
}
```

→ 16 byte 이상 입력 시..

과제 풀이

과제 1 – Variable Corruption

```
minibeef@argos-edu:~/cedu/week3/hw$ ./Variable-Corruption  
input : aaaaaaaaaaaaaaB  
type is B, you WIN!!
```


과제 2 - 별 피라미드

```
minibee@argos-edu:~/cedu/week3/hw$ ./star
높이를 입력하시오 : 4
  *
 ***
*****
minibee@argos-edu:~/cedu/week3/hw$ ./star
높이를 입력하시오 : 7
  *
 ***
*****
*****
*****
*****
*****
minibee@argos-edu:~/cedu/week3/hw$ ./star
높이를 입력하시오 : 9
  *
 ***
*****
*****
*****
*****
*****
*****
*****
*****
minibee@argos-edu:~/cedu/week3/hw$
```

입력 받은 높이만큼 별 피라미드 출력

힌트) 반복문

과제 2 - 별 피라미드

```
#include <stdio.h>

int main()
{
    int h;
    printf("높이를 입력하시오 : ");
    scanf("%d", &h);

    for(int i = 0; i < h; i++) { // 세로축 이동
        for(int j = h - 1; j > i; j--) { // 가로축 이동(공백)
            printf(" ");
        }

        for(int j = 0; j < 2 * i + 1; j++) { // 가로축 이동(별)
            printf("*"); // 1, 3, 5, 7, 9, ...
        }
        puts(""); // 줄바꿈
    }
}
```

〈정답〉

“패턴에 따른 적절한 for문 조건 세우기”

과제 2 - 별 피라미드

```
#include <stdio.h>

int main()
{
    int h;
    printf("높이를 입력하시오 : ");
    scanf("%d", &h);

    for(int i = 0; i < h; i++) { // 세로축 이동
        for(int j = h - 1; j > i; j--) { // 가로축 이동(공백)
            printf(" ");
        }

        for(int j = 0; j < 2 * i + 1; j++) { // 가로축 이동(별)
            printf("*"); // 1, 3, 5, 7, 9, ...
        }
        puts(""); // 줄바꿈
    }
}
```

〈정답〉

입력 : 4

```
      *
     ***
    *****
   ********
```

과제 2 - 별 피라미드

```
#include <stdio.h>

int main()
{
    int h;
    printf("높이를 입력하시오 : ");
    scanf("%d", &h);

    for(int i = 0; i < h; i++) { // 세로축 이동
        for(int j = h - 1; j > i; j--) { // 가로축 이동(공백)
            printf(" ");
        }

        for(int j = 0; j < 2 * i + 1; j++) { // 가로축 이동(별)
            printf("*"); // 1, 3, 5, 7, 9, ...
        }

        puts(""); // 줄바꿈
    }
}
```

〈정답〉

입력 : 4

```
○○○*
○○***
○*****
*****
```

공백 출력은 “높이-1”, 총이 쌓일수록 1씩 감소

과제 2 - 별 피라미드

```
#include <stdio.h>

int main()
{
    int h;
    printf("높이를 입력하시오 : ");
    scanf("%d", &h);

    for(int i = 0; i < h; i++) { // 세로축 이동
        for(int j = h - 1; j > i; j--) { // 가로축 이동(공백)
            printf(" ");
        }

        for(int j = 0; j < 2 * i + 1; j++) { // 가로축 이동(별)
            printf("*"); // 1, 3, 5, 7, 9, ...
        }

        puts(""); // 줄바꿈
    }
}
```

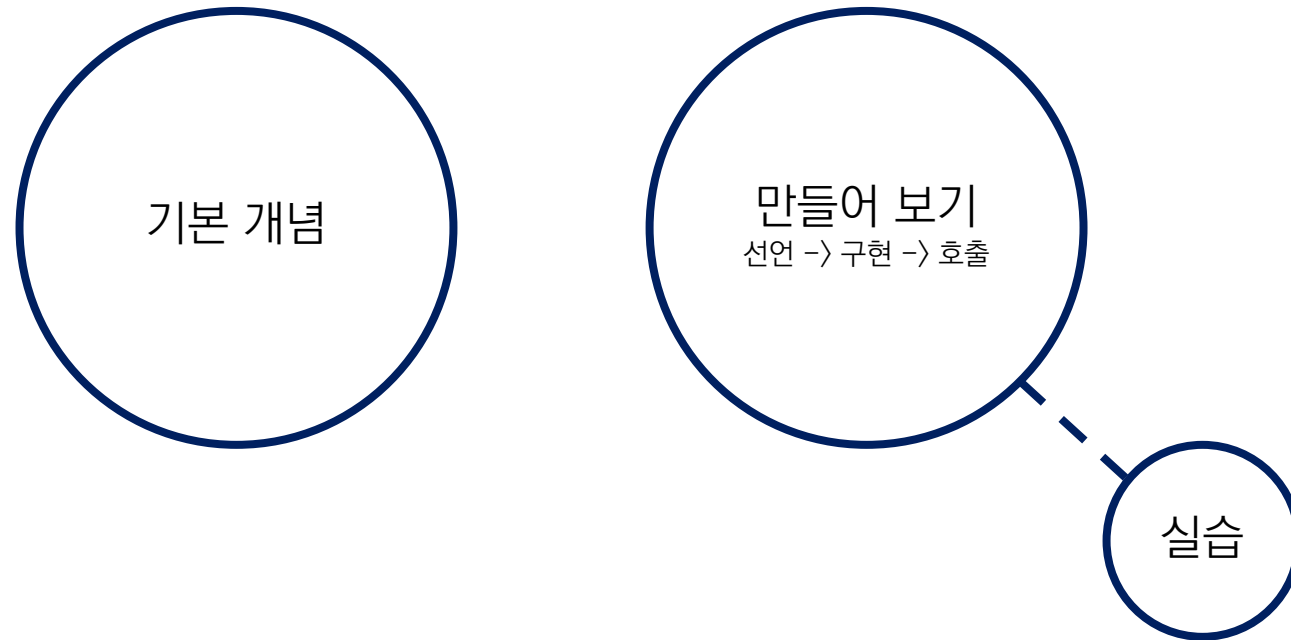
〈정답〉

입력 : 4

1층	*	1
2층	***	3
3층	*****	5
4층	*****	7

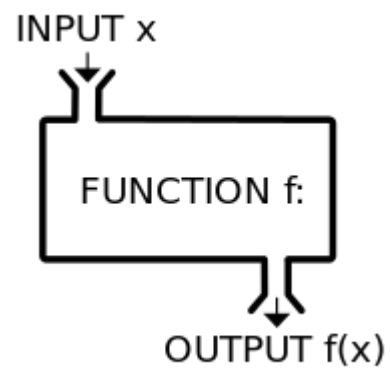
1층에 1개, 2층에 3개, 3층에 5개, ... n층에는 $2n-1$ 개

2. 함수



기본 개념

수학에서의 함수? $f(x)$



$$f(x) = x + 1;$$

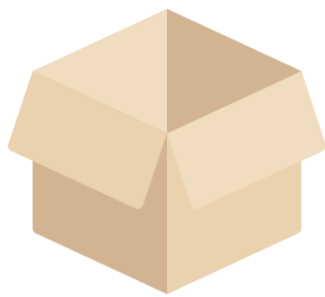
$$f(3) = 3 + 1 = 4$$

정의된 행동(1을 더한다)을 수행하고
결과를 내준다.

| 함수

기본 개념

프로그램에서의 함수(수학과 비슷)

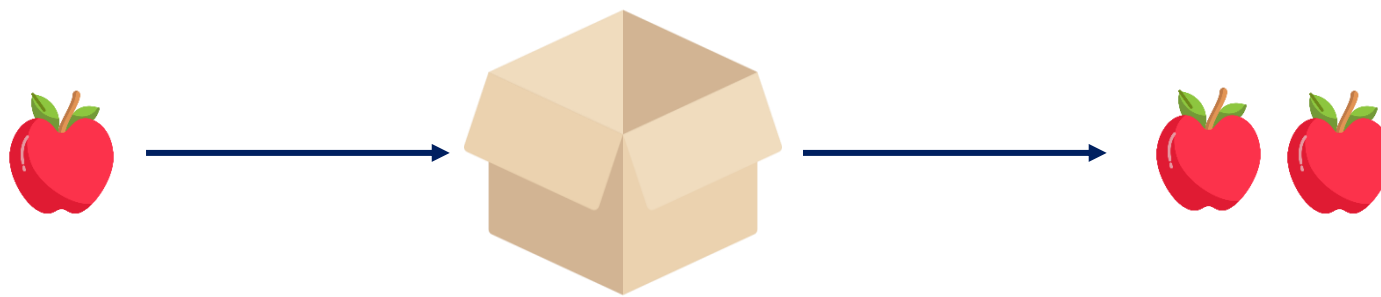


〈물건을 넣으면 무조건 두배가 되는 상자〉

함수

기본 개념

프로그램에서의 함수(수학과 비슷)

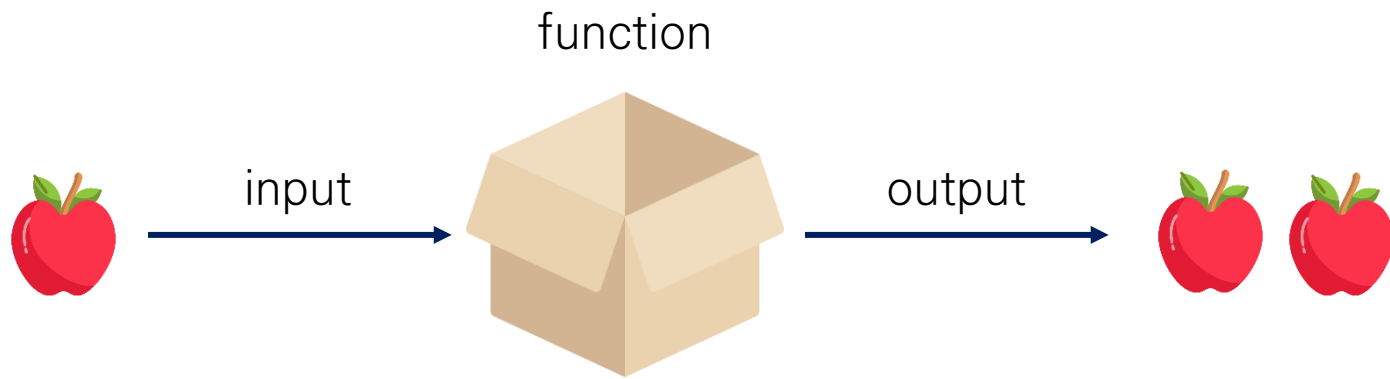


〈물건을 넣으면 무조건 두배가 되는 상자〉

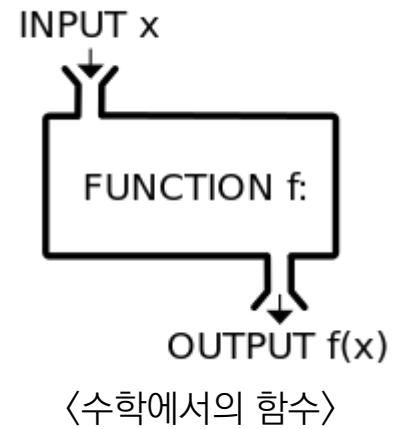
함수

기본 개념

프로그램에서의 함수(수학과 비슷)



〈물건을 넣으면 무조건 두배가 되는 상자〉



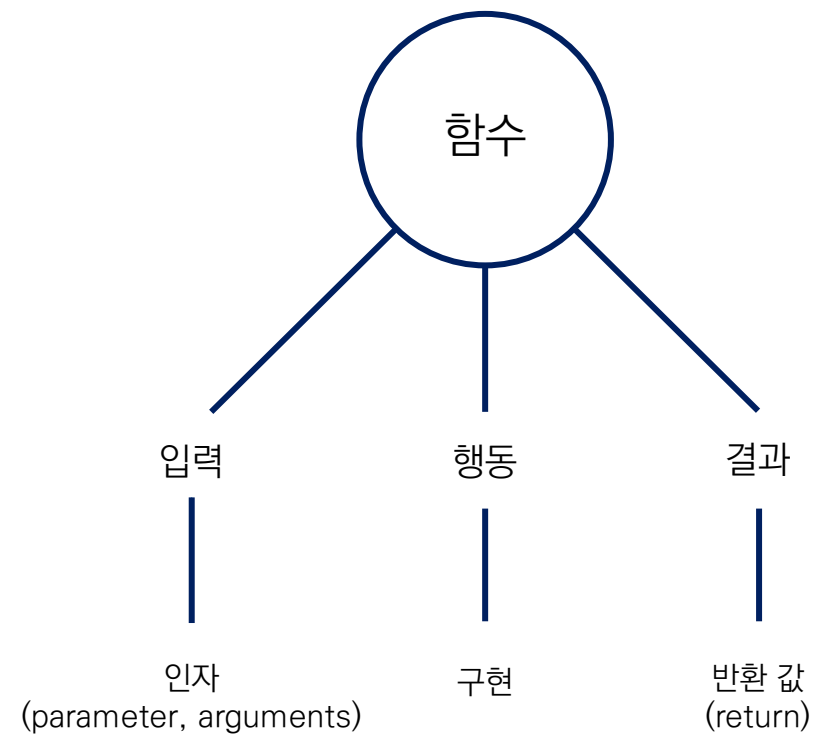
| 함수

기본 개념

“ 정해진 행동을 수행하되, 입력과 결과가 존재 ”

함수

만들어 보기



함수

만들어 보기

$$f(x) = x + 1$$

```
int plus_one(int x)
{
    return x + 1;
}
```

인자
(parameter, arguments)

함수

만들어 보기

$$f(x) = x + 1$$



```
int plus_one(int x)
```

```
{
```

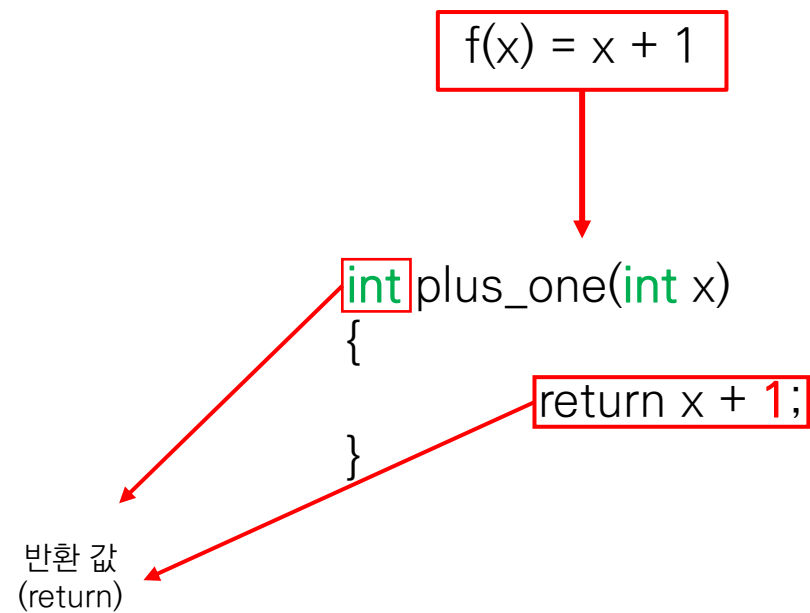
```
    return x + 1;
```

```
}
```

구현

함수

만들어 보기



| 함수

만들어 보기

〈 선언 〉

```
[결과 타입] [함수 이름]([인자 1], [인자 2], ...)  
{  
    [구현]  
    return [결과 값]  
}
```


| 함수

만들어 보기

〈 호출 〉


[함수이름]([인자]);

만들어 보기

```
#include <stdio.h>

int plus_one(int x)
{
    return x + 1;
}

int main()
{
    int function_result = plus_one(4);
    printf("%d\n", function_result);
}
```



```
minibee@argos-edu:~/cedu/week4$ ./ex1
5
```

덧셈기

```
#include <stdio.h>

int adder( ) 인자
{
    결과
}

int main()
{
    int num1, num2;

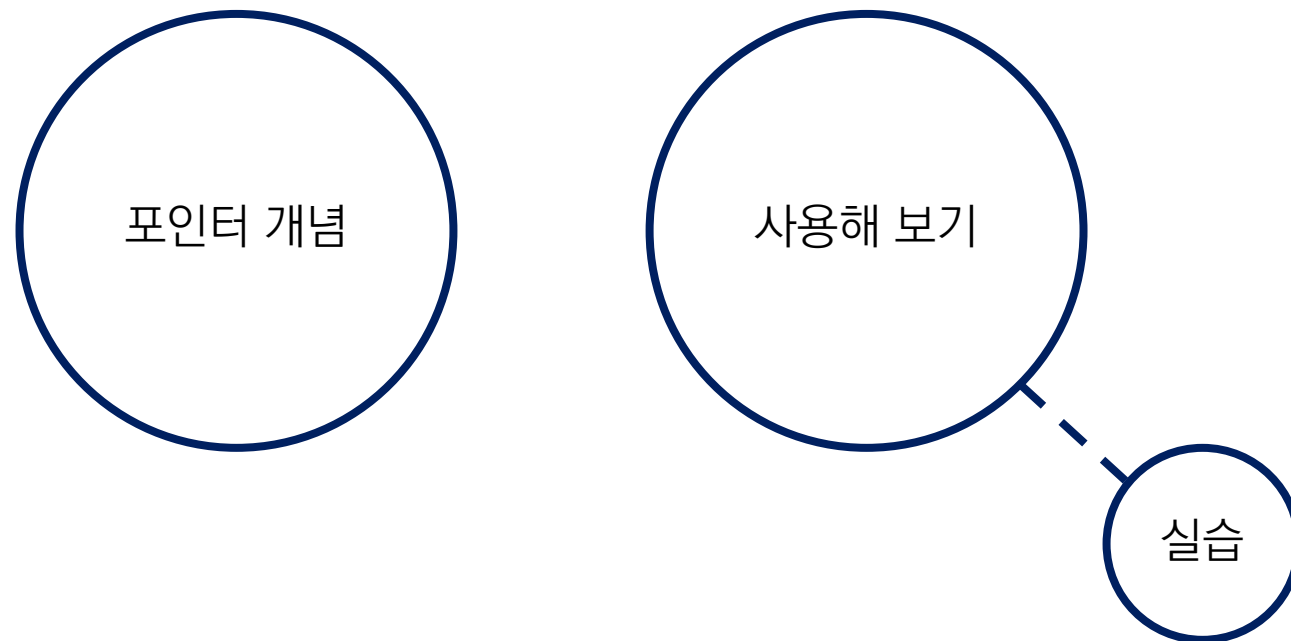
    printf("입력 1 : ");
    scanf("%d", &num1);

    printf("입력 2 : ");
    scanf("%d", &num2);

    printf("%d + %d = %d\n", num1, num2, );
}
```

```
minibee@argos-edu:~/cedu/week4$ ./pr1
입력 1 : 3
입력 2 : 4
3 + 4 = 7
```

3. 포인터



포인터

포인터 개념

point + -er



가르키는



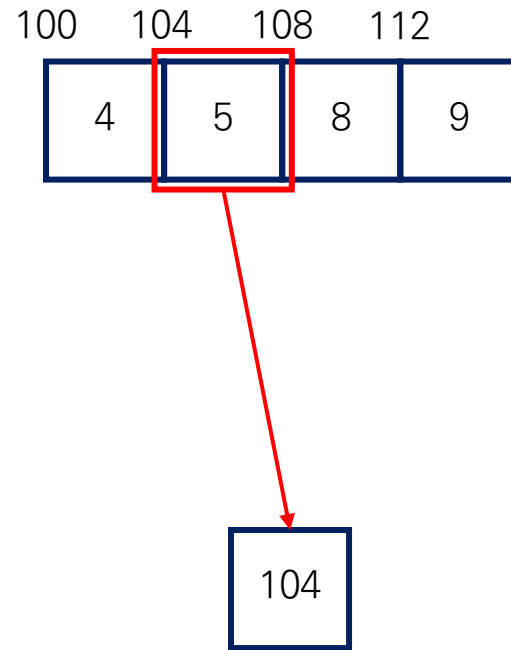
녀석

뭘 가르키느냐? -> 변수의 주소

포인터

포인터 개념

“값”을 저장하는게 아닌 “주소”를 저장하는 변수 : 포인터 변수



포인터 변수

| 포인터

포인터 개념

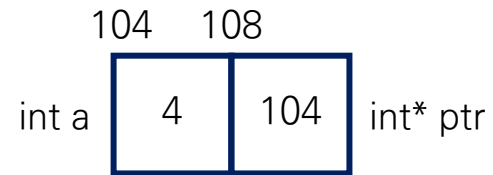
굳이 왜? 주소를 저장하는 변수가 필요한가

-> Call by Value(과제에서 만나용 ㅎㅎ)

포인터 사용해 보기

데이터 타입에 *이 붙은 형태
→ 나는 포인터 변수다!

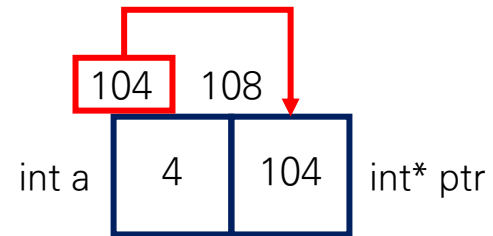
```
int a = 4;  
int* ptr = &a
```



포인터

포인터 사용해 보기

```
int a = 4;  
int* ptr = &a
```



포인터 사용해 보기

```
#include <stdio.h>

int main()
{
    int a = 4;
    int* ptr = &a;
    printf("%p\n", ptr);
}
```

```
minibeef@argos-edu:~/cedu/week4$ ./pr2
0x7ffdf69981ec
```

포인터 사용해 보기

```
#include <stdio.h>

int main()
{
    int a = 4;
    int* ptr = &a;
    printf("%p\n", ptr);
}
```

```
minibeef@argos-edu:~/cedu/week4$ ./pr2
0x7ffdf69981ec
```

포인터 사용해 보기

* : 역참조 - 주소로 가서 값을 보겠다.

```
#include <stdio.h>

int main()
{
    int a = 4;
    int* ptr = &a;
    printf("%d\n", *ptr);
}
```

```
minibee@argos-edu:~/cedu/week4$ ./pr2
4
```

포인터 덧셈

```
#include <stdio.h>

int main()
{
    int a = 4;
    int b = 5;
    int* [redacted]
    int* [redacted]

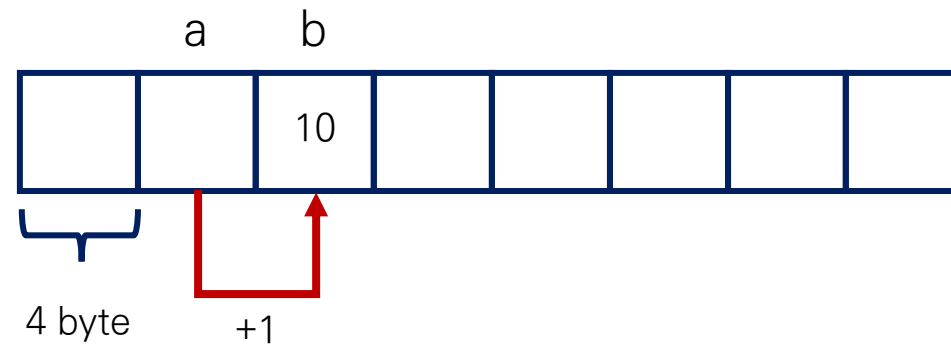
    printf("%d\n", [redacted]);
}
```

```
minibee@argos-edu:~/cedu/week4$ ./pr3
9
```

포인터 사용해 보기 - 포인터 연산

```
#include <stdio.h>

int main()
{
    int a;
    int b = 10;
    scanf("%d", &a + 1);
    printf("%d\n", b);
}
```



포인터 사용해 보기 - 포인터 연산

산술 연산 vs 주소 연산

(주소 아닌 숫자) $10 + 1 \Rightarrow 11$



산술 연산

(int형 변수의 주소) $10 + 1 \Rightarrow 14$



주소 연산

(double형 변수의 주소) $10 + 1 \Rightarrow 18$



주소 연산

포인터 사용해 보기 - 포인터 연산

```
#include <stdio.h>

int main()
{
    int a = 4;
    int* pa = &a;
    printf("%d %d %d\n", pa, pa + 1, pa + 2);
}
```

```
minibeef@argos-edu:~/cedu/week4$ ./pr4
1919788268 1919788272 1919788276
```

4byte 차이

wrong swap

```
void wrong_swap(int x, int y)
{
    int temp = x;
    x = y;
    y = temp;
}

int main()
{
    int a = 3;
    int b = 4;

    printf("before swap %d %d\n", a, b);

    wrong_swap(a, b);
    printf("after swap %d %d\n", a, b);
}
```

```
minibeef@argos-edu:~/cedu/week4/hw$ ./wrong_swap
before swap 3 4
after swap 3 4
```

두 값의 위치를 바꾸는 함수

wrong swap

```
void wrong_swap(int x, int y)
{
    int temp = x;
    x = y;
    y = temp;
}

int main()
{
    int a = 3;
    int b = 4;

    printf("before swap %d %d\n", a, b);

    wrong_swap(a, b);
    printf("after swap %d %d\n", a, b);
}
```

```
minibeef@argos-edu:~/cedu/week4/hw$ ./wrong_swap
before swap 3 4
after swap 3 4
```

두 값의 위치를 바꾸는 함수

과제

wrong swap

```
minibeef@argos-edu:~/cedu/week4/hw$ ./wrong_swap  
before swap 3 4  
after swap 4 3
```

포인터를 사용하여 위와 같이 출력되도록 만들기

끝