

Java를 알고 C배우기

컴퓨터프로그래밍3

week 5-4 배열, 포인터-포인터연산

2022.1학기
충남대 조은선

& 연산자 복습

- ▶ & 연산자: 변수를 주면 변수의 위치(주소)를 주는 연산자

```
int num = 5;  
int * pnum = &num;           // 변수 num의 위치(주소)를 pnum에 보관해둔다
```

```
int num = 5;  
double * pnum = &num1;       // NO! 자료형 불일치!!
```

```
double num2 = 5.0;  
int * pnum2 = &num2;         // NO! 자료형 불일치!!
```

& 연산자의 역연산

▶ &의 역연산자가 있을까?

▶ & 는, 변수를 주면 주소를 주는 연산자 예) &num

▶ 거꾸로 해서, 주소를 주면 변수를 주는 연산자가 있을까? 예) 1024번지 → num ?

→ 답) 없다

→ 그러나 주소(l-value)를 주면 (변수이름을 주지는 않지만)

변수가 가진 값(r-value)을 꺼내 주는 연산자는 있다!

즉,

```
int num = 5;
```

```
printf("%d", op(&num) );      // 5를 출력하는 그런 어떤 op가 존재한다 → * 연산자
```

* 연산자

▶ * 연산자

- ▶ &의 역연산과 비슷
- ▶ 주소를 주면 그 주소에 들어있는 값을 꺼내 준다

```
int num = 10;  
printf("%p", &num);    // num변수의 주소 즉 1048 등을 출력  
printf("%d", *(&num)); // 10을 출력  
                        // 즉, num 변수의 주소를 주니 (&num),  
                        // num변수가 가진 값을 출력
```

참고1) 연산자 *&는 자기자신, 즉 상쇄된다!

참고2) 연산자 &*는 불가능하다 (상수값의 주소를 취하려는 시도)

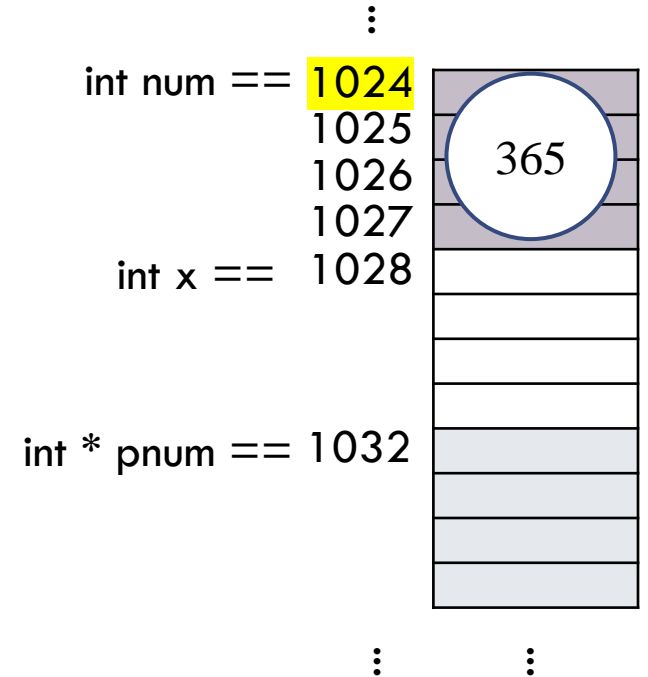
포인터 변수와 *, & - 예

```
int * pnum;
int x;
int num = 365;          // 1024 번지에 4byte를 할당하고 365를 넣음

printf("%d", num);       // 365를 출력
printf("%p", &num);      // 주소 1024를 출력
printf("%d", *&num);     // 365를 출력
printf("%p", &*&num);    // 오류! 상수 값의 주소를 취하려고 함

pnum = &num;
printf("%p", pnum);      // 주소 1024 출력
printf("%d", *pnum);     // 365를 출력
printf("%p", &*pnum);    // 오류! 상수 값의 주소를 취하려고 함

x = *pnum;               // 365가 x에 저장됨
printf("%d", x);         // 365를 출력
printf("%p", &x);        // 주소 1028 출력 (1024 아님)
printf("%d", *&x);      // 365를 출력
printf("%p", &*&x);      // 오류! 상수값의 주소를 취하려고 함
```



(* ptr변수)

- ▶ * 과 포인터 변수를 함께 써서 일반 변수처럼 사용할 수 있다

```
int num1=100, num2=100;  
int * pnum;
```

(*pnum)은
num1, num2같은 int 변수처럼 사용된다

```
pnum = &num1;  
(*pnum) += 30;           // (*pnum)은 num1의 별칭처럼 사용된다  
pnum = &num2;  
(*pnum) -= 30;           // (*pnum)은 num2의 별칭처럼 사용된다  
  
printf("num1:%d, num2:%d\n", num1, num2);
```

포인터 타입 규칙과 형변환

[illegible]

Quiz

- ▶ 아래는 1부터 10까지 더하는 코드이다.

```
int sum = 0;
int num;
for(num = 1; num < = 10; num++)
    sum += num;
```

이 코드에

int * psum, int* pnum; 를 추가하고,
for문에서 num과 sum 대신 (*psum) 과 (*pnum)를 써서 동작하도록 고쳐보시오.