

C프로그래밍 (CSE2035) (실습18)



Ji-Hwan Kim, Ph.D.

Dept. of Computer Science and Engineering

Sogang University

Seoul, Korea

Tel: +82-2-705-8924

Email : kimjihwan@sogang.ac.kr



제출 형식

1. 각 문제에 대한 소스 코드를 압축하여 사이버캠퍼스에 업로드
 - 압축 파일명: "[실습#]학번_이름.zip" (#은 실습번호)
 - 각 소스코드 파일명: "cp실습번호_학번_p문제번호.c"
2. 업로드 후 제출 시, 제출 기한 내에 사이버캠퍼스에 제출
 - 화요일 실습의 경우: 이번주 수요일 오후 11시 59분까지
 - 목요일 실습의 경우: 이번주 금요일 오후 11시 59분까지
3. COPY 등의 문제 발생 시 실습 0점 및 각종 불이익을 줄 것



Practice 1

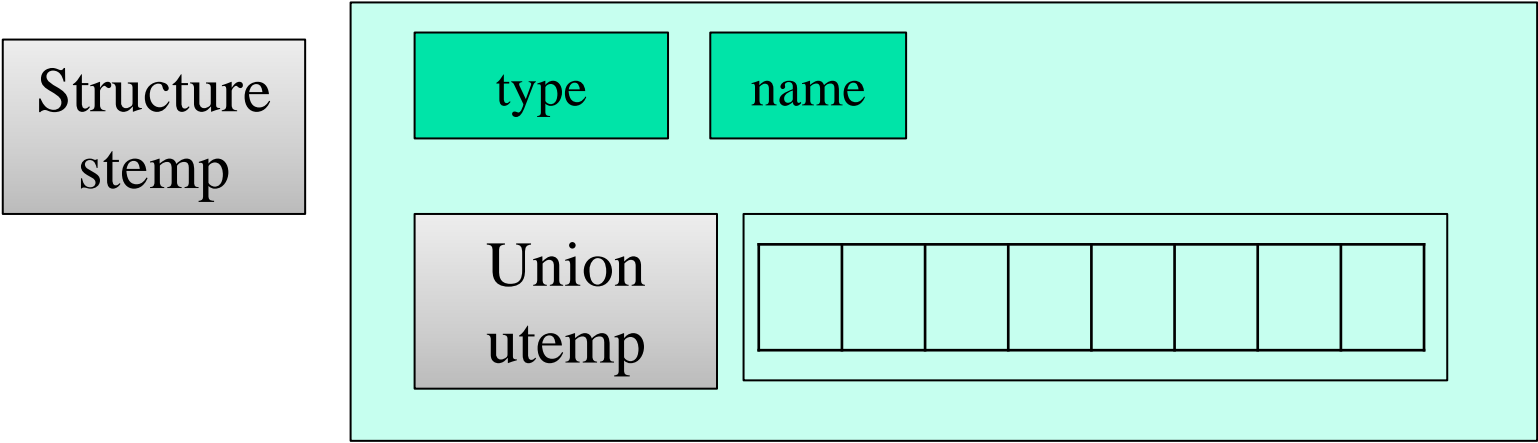
- 입력 받은 수에 대한 제곱을 계산하는 프로그램을 작성한다.

```

Structure
int type; // 입력 받은 수의 변수 타입 (1: integer 2: floating poiny 3: double)
char name[20]; // 변수 타입
union utemp u; // 입력 받은 수
    
```

```

Union  int i;
      float f;
      double d;
    
```





Practice 1

- 프로그램이 시작되면 반복 횟수를 입력 받아 해당 횟수만큼 계산을 반복한다.
 - Type은 1 : integer, 2: float, 3 : double 으로 정의한다.
- 입력 받은 숫자에 대한 제곱을 구하는 함수를 통하여 공용체 내에 다시 제곱의 결과를 저장하고, 이를 변수 타입에 맞게 출력하도록 한다.
 - 구조체의 name을 출력할 때, 10칸에 맞춰서 왼쪽 정렬을 하도록 한다.
- 함수의 원형은 다음과 같다.

```
union utemp{
    int i;
    float f;
    double d;
};
```

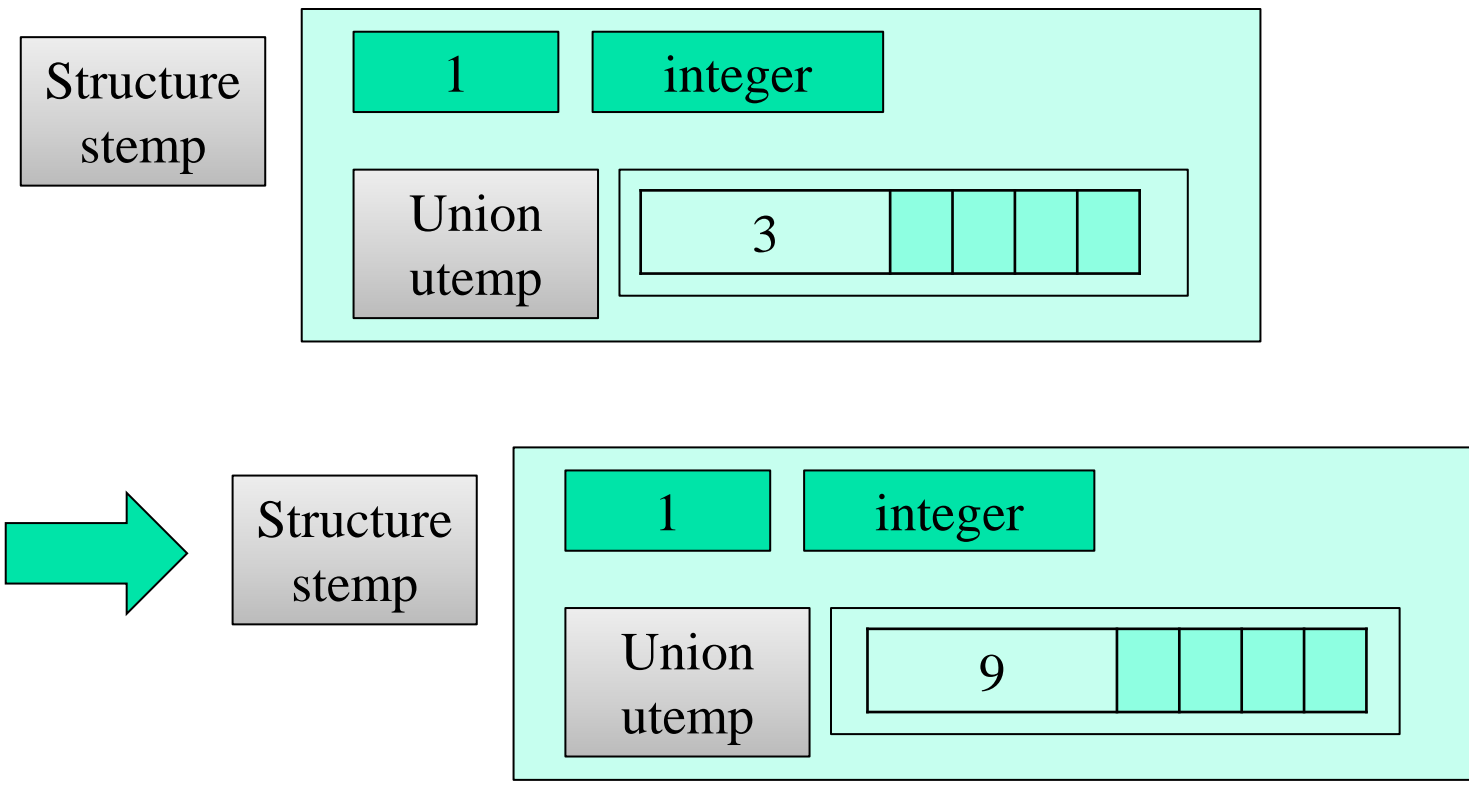
```
typedef struct {
    int type;
    char name[20];
    union utemp u;
}stemp;

void power(stemp *s);
```

Practice 1

- 입력 받은 수 y 에 대해 $y = x^2$ 을 계산할 수 있도록 한다.

Example) type : 1 (integer) value : 3





Practice 1

- 프로그램 실행 예시

```
gr120160221@cspro1:~/exercise/18$ ./18_2
Number of Iterations : 3
Type [1 for int, 2 for float, 3 for double] : 1
Enter an integer value : 4
Type [1 for int, 2 for float, 3 for double] : 2
Enter a float value : 3.2
Type [1 for int, 2 for float, 3 for double] : 3
Enter a double value : 2.664
-----Result-----
integer : 16
float   : 10.2400
double  : 7.0969
```



Practice 2

- 강의 시간에 배운 **linked list insertion**을 실제로 구현해 보자. **Node**를 **list**의 처음/끝 부분에 삽입하는 함수를 만들도록 하자.

```
typedef struct node* nptr;
typedef struct node
{
    char data;
    nptr link;
}NODE;

nptr head = NULL;
```

- List의 head에 삽입하는 함수를 이용하여 p, p, a을 삽입하고, list의 끝에 삽입하는 함수를 이용하여 l, e를 삽입하라. 그리고 그 결과를 출력하라.
- 구현 함수 :

InsertNodeAtFront(char) / InsertNodeAtBack(char) / PrintList()



Practice 2

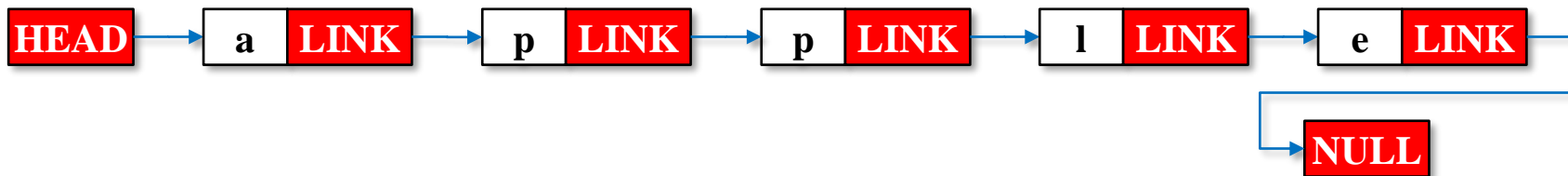
- **void InsertNodeAtFront(char a)**
 - 새로운 node를 생성하여 list의 맨 앞에 추가한다.
 - Parameter: a = 새로운 node의 data값

- **void InsertNodeAtBack(char b)**
 - 새로운 node를 생성하여 list의 맨 뒤에 추가한다.
 - Parameter: b = 새로운 node의 data값

- **void PrintList()**
 - Linked list의 모든 값을 출력한다.

Practice 2

- Insert operation은 아래와 같은 순서로 이루어진다.





Practice 2

- Main함수를 아래와 같다.

```
int main()
{
    InsertNodeAtFront('p');
    InsertNodeAtFront('p');
    InsertNodeAtFront('a');
    InsertNodeAtBack('l');
    InsertNodeAtBack('e');
    PrintList();

    return 0;
}
```

- 프로그램 수행 결과

```
gr120160221@csp:~/exercise/20$ ./20_2
apple
```



Practice 3

■ 수업시간에 배운 **linked list** 구조를 이용하여, 10개의 숫자가 들어올 때마다 순서대로 **Linked List** 구조로 저장하고 **data**를 기준으로 한 **linked list**를 오름차순으로 정렬하는 프로그램을 작성한다.

■ 제한 조건

- 정렬은 모든 **Node**가 완성된 이후 **Sort** 함수 단 한번의 호출로 진행한다.
- 함수의 원형과 **main** 에서 입력이 들어가는 배열은 다음과 같다.

```
typedef struct node{  
    int data;  
    struct node * link;  
}NODE;  
typedef NODE* nptr;  
  
void Insert(nptr* head, int data);  
void Sort(nptr* head);  
void PrintAll(nptr head);
```

```
int A[10] = { 3, 9, 8, 2, 5, 10, 7, 1, 4, 6 };
```



Practice 3

■ 출력 예시

출력은 입력이 끝난 후 모든 노드를 한 번 출력하고, 정렬한 이후 한 번 더 출력한다.

```
3 9 8 2 5 10 7 1 4 6
1 2 3 4 5 6 7 8 9 10
```