



---

# R E P O R T

## C프로그래밍2 과제3

|     |                  |
|-----|------------------|
| 과목명 | C 프로그래밍 II       |
| 분반  | 2 분반             |
| 교수  | 정 구 철            |
| 학번  | 2020136129       |
| 이름  | 최 수 연            |
| 제출일 | 2020년 9월 30일 수요일 |

1. 'int a=10'을 선언하고 변수 a의 주소를 출력하시오.

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

```
int main(void)
```

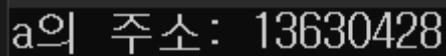
```
{
```

```
    int a = 10;
```

```
    printf("a의 주소: %u\n", &a);
```

```
    return 0;
```

```
}
```



a의 주소: 13630428

C:\Users\choi6\source

2. 'int a=10'을 선언하고 변수 a를 'int \*p'에 맵핑한 후, p를 사용하여 a의 주소(address)와 값(value)을 출력하시오

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

```
int main(void)
```

```
{
```

```
    int a = 10;
```

```
    int* p = NULL;
```

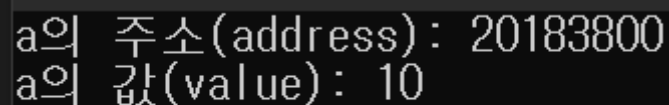
```
    p = &a;
```

```
    printf("a의 주소(address): %u\n", p);
```

```
    printf("a의 값(value): %d\n", *p);
```

```
    return 0;
```

```
}
```



a의 주소(address): 20183800  
a의 값(value): 10

C:\Users\choi6\source\renosh

3. 'double a=10, int b=3'을 정의하고 각각을 'double \*pa, int\*pb'에 맵핑한 다음 'a,b,pa,pb' 변수의 크기를 출력하시오

☐ Hint, sizeof 함수를 사용하시오.

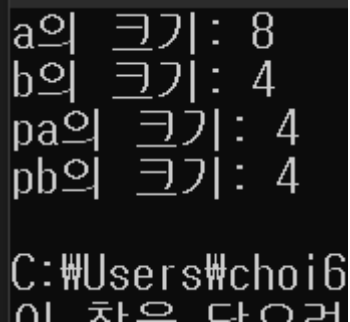
```
#include <stdio.h>

int main(void)
{
    double a = 10;
    int b = 3;
    double* pa = NULL;
    int* pb = NULL;

    pa = &a;
    pb = &b;

    printf("a의 크기: %u\n", sizeof(a));
    printf("b의 크기: %u\n", sizeof(b));
    printf("pa의 크기: %u\n", sizeof(pa));
    printf("pb의 크기: %u\n", sizeof(pb));

    return 0;
}
```



```
a의 크기: 8
b의 크기: 4
pa의 크기: 4
pb의 크기: 4

C:\Users\choi6
이 작은 다음
```

4. int \*p를 10000으로 초기화시킨 후 p의 값을 출력하시오

☐ warning이 발생하지 말아야함.

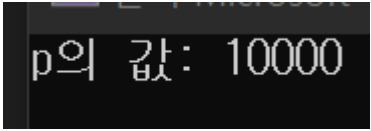
☐ Hint, '10000'은 데이터 타입이 정수지만, '(int \*)10000'는 데이터 타입이 포인터형 정수

```
#include <stdio.h>

int main(void)
{
    int* p = (int*)10000;

    printf("p의 값: %d\n", p);

    return 0;
}
```



p의 값: 10000

5. 'int a=10'을 선언하고 변수 a를 'int \*p'에 맵핑한 후, p를 사용하여 a의 값을 30으로 변경하고 "printf("value of a is %d",a);"를 사용하여 a의 값을 출력하시오.

```
#include <stdio.h>

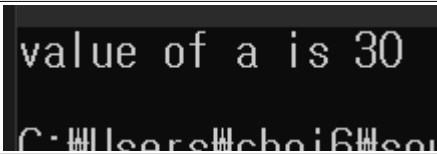
int main(void)
{
    int a = 10;
    int* p = NULL;

    p = &a;

    *p = 30;

    printf("value of a is %d\n", a);

    return 0;
}
```



value of a is 30

6. 'double a=10'을 정의하고 'double \*pa'에 맵핑한 후 "++\*pa, \*pa++,\*pa"을 순서대로 출력하시오.

- ☐ 연산은 printf에 내장할 것, 예)printf("( \*pa)++ is %lf\n",(\*pa)++);
- ☐ 순서를 지키지 않을시 오류가 날 수 있음
- ☐ hint, 마지막 \*pa는 예상했던 것과 다르게 나올 수 있음. 왜 그런지 생각해보자

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

```
int main(void)
```

```
{
```

```
    double a = 10;
```

```
    double* pa = NULL;
```

```
    pa = &a;
```

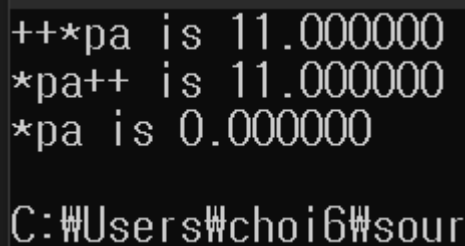
```
    printf("++*pa is %lf\n", ++ * pa);
```

```
    printf("*pa++ is %lf\n", *pa++);
```

```
    printf("*pa is %lf\n", *pa);
```

```
    return 0;
```

```
}
```



```
++*pa is 11.000000
*pa++ is 11.000000
*pa is 0.000000

C:\Users\choi6\source
```

마지막 \*pa가 예상했던 것과 다르게 나오는 이유는 앞의 \*pa++에서 pa를 증가시켰기 때문에 주소 값이 달라져서 기존의 pa에는 \*pa값이 존재하지 않게 된다. 따라서 0으로 초기화된 것이다.

[문제 7~10]

sns에 있어서 권한은 굉장히 중요하다. 내 게시물은 누구든 볼 수 있지만, 누구든 수정할 수는 없다. 이 관계가 무너지는 것을 해킹 혹은 sns 도용이라고 한다. 다음과 같은 예를 생각해보자.

- ☐ 'char \*me', 'char \*other'를 각각 본인과 다른 사람으로 정의한다.
- ☐ 포인터 변수 me와 other의 주소를 id라고 정의한다.
- ☐ char post='c'로 정의한다.

7. me, other의 id를 출력하시오

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

```
int main(void)
```

```
{
```

```
    char post = 'c';
```

```
    char* me = NULL;
```

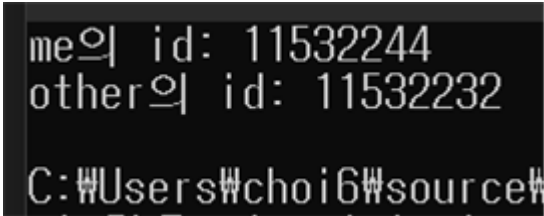
```
    char* other = NULL;
```

```
    printf("me의 id: %d\n", &me);
```

```
    printf("other의 id: %d\n", &other);
```

```
    return 0;
```

```
}
```



```
me의 id: 11532244
other의 id: 11532232
C:\Users\choi6\source#
```

8. me를 통해 post값을 'd'로 변경하고 other를 통해 post값을 출력하시오.

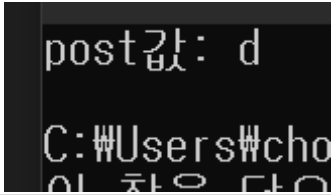
```
#include <stdio.h>

int main(void)
{
    char post = 'c';
    char* me = NULL;
    char* other = NULL;

    me = &post;
    *me = 'd';
    other = &post;

    printf("post값: %c\n", *other);

    return 0;
}
```



post값: d  
C:\Users\cho  
이 차은 님

9. 권한이 있는 사람이 post를 수정하면 “modified”란 알람을 출력하고 권한이 없는 사람이 post를 수정하면 “There is no permission”이란 알람을 출력하도록 한다. other을 통해 post를 'e'로 변경 시도하고 알람과 post값을 출력하시오.

□ other는 권한이 없는 사람이다.

□ hint. 포인터 변수도 변수이므로 포인터 변수 자체의 주소를 가지고 있음.

```
#include <stdio.h>

int main(void)
{
    char post = 'c';
    char* me = NULL;
    char* other = NULL;
    char y, z = post;
    int x;

    printf("'me' = 1, 'other' = 2\n");
    scanf_s("%d", &x);

    if (x == 1)
    {
        printf("변경할 문자를 입력하시오: ");
        scanf_s(" %c", &y);
        me = &post;
        *me = y;
        z = *me;
        printf("modified\n");
        printf("post값: %c\n", *me);
    }
    else if (x == 2)
    {
        printf("변경할 문자를 입력하시오: ");
        scanf_s(" %c", &y);
        other = &post;
        *other = y;
        printf("There is no permission\n");
        *other = z;
        printf("post값: %c\n", *other);
    }

    return 0;
}
```



```
'me' = 1, 'other' = 2
2
변경할 문자를 입력하시오: e
There is no permission
post값: c
```

10. char형 포인터의 포인터 'char\*\* hacker'를 해커라고 정의하자(char\* (\*hacker)라고 생각해도 됨). hacker를 통해 me의 id를 따내고 post 값을 'z'로 변경한 후 알람과 post값을 출력하시오.

```
#include <stdio.h>

int main(void)
{
    char post = 'c';
    char* me = NULL;
    char* other = NULL;
    char** hacker = &me;

    *hacker = &post;

    **hacker = 'z';

    printf("modified\n");
    printf("post값: %c\n", post);

    return 0;
}
```

```
modified
post값: z
```