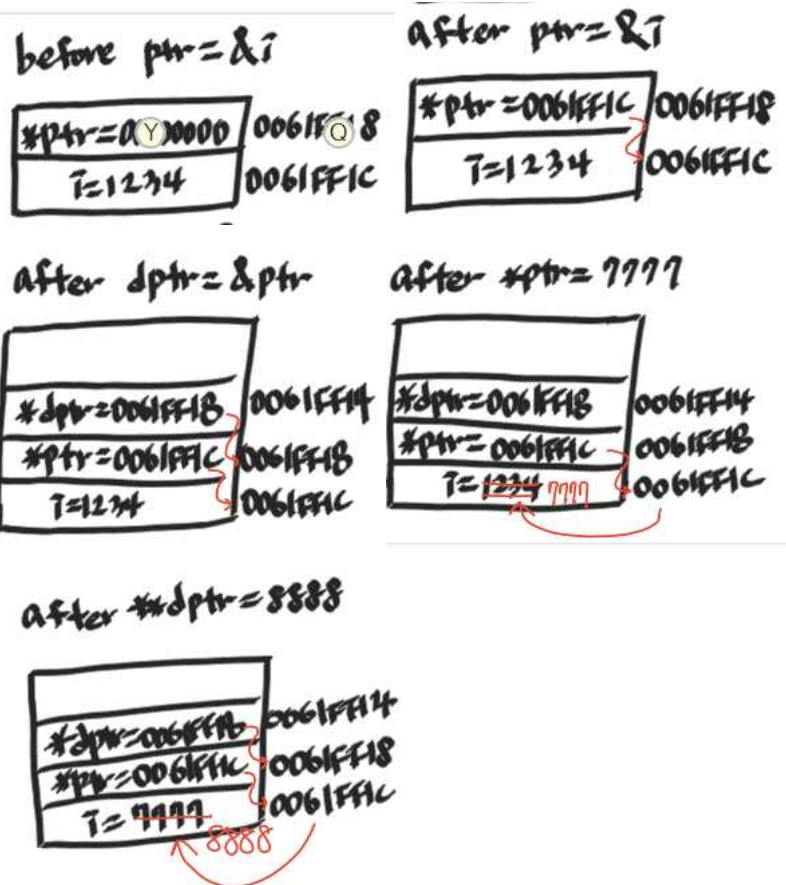


## homework #2

정보통계학과 2020045070 김가현



```

C lab2-1.c > main()
1  #include <stdio.h>
2  int main()
3  {
4      char charType;
5      int integerType;
6      float floatType;
7      double doubleType;
8
9      printf("----- [Gahyun KIM] [2020045070] -----");
10
11     printf("Size of char: %ld byte\n",sizeof(charType)); //문자형 변수 charType의 크기 출력 : 1 byte
12     printf("Size of int: %ld bytes\n",sizeof(integerType)); //정수형 변수 integerType의 크기 출력 : 4 byte
13     printf("Size of float: %ld bytes\n",sizeof(floatType)); //실수형 변수 floatType의 크기 출력 : 4 byte
14     printf("Size of double: %ld bytes\n",sizeof(doubleType)); //실수형 변수 doubleType의 크기 출력 : 8 byte
15     printf("-----\n");
16     printf("Size of char: %ld byte\n",sizeof(char)); //자료형 char의 크기 출력 : 1 byte
17     printf("Size of int: %ld bytes\n",sizeof(int)); //자료형 int의 크기 출력 : 4 byte
18     printf("Size of float: %ld bytes\n",sizeof(float)); //자료형 float의 크기 출력 : 4 byte
19     printf("Size of double: %ld bytes\n",sizeof(double)); //자료형 double의 크기 출력 : 8byte
20     printf("-----\n");
21     printf("Size of char*: %ld byte\n",sizeof(char*)); //자료형 char의 주소 값의 크기 출력 : 4 byte
22     printf("Size of int*: %ld bytes\n",sizeof(int*)); //자료형 int의 주소 값의 크기 출력 : 4 byte
23     printf("Size of float*: %ld bytes\n",sizeof(float*)); //자료형 float의 주소 값의 크기 출력 : 4 byte
24     printf("Size of double*: %ld bytes\n",sizeof(double*)); //자료형 double의 주소 값의 크기 출력 : 4 byte
25     return 0;
26 }
27

```

```

[----- [Gahyun KIM] [2020045070] -----]
Size of char: 1 byte
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
-----
Size of char: 1 byte
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
-----
Size of char*: 4 byte
Size of int*: 4 bytes
Size of float*: 4 bytes
Size of double*: 4 bytes
* Terminal will be reused by tasks, press any

```

```

52-2.c / main()
#include <stdio.h>
int main()
{
    int i;
    int *ptr;
    int **dptr;
    i = 1234;

    printf("----- [Gahyun KIM] [2020045070] -----]\n");

    printf("[checking values before ptr = &i] \n"); // ptr=&i하기 전의 값들 출력해보기
    printf("value of i == %d\n", i); //i 값 출력 : 1234
    printf("address of i == %p\n", &i); //i의 값 1234가 저장되어 있는 주소 출력
    printf("value of ptr == %p\n", ptr); //ptr의 실제 값 출력 : 초기화가 안되어 있으므로 NULL값 출력
    printf("address of ptr == %p\n", &ptr); //ptr의 주소 출력
    ptr = &i; /* ptr is now holding the address of i */
    printf("\n[checking values after ptr = &i] \n");
    printf("value of i == %d\n", i); //i의 값 출력 : 1234
    printf("address of i == %p\n", &i); //i의 값 1234가 저장되어 있는 주소 출력
    printf("value of ptr == %p\n", ptr); //ptr의 값 출력 즉, i의 주소가 출력
    printf("address of ptr == %p\n", &ptr); //ptr의 주소 출력
    printf("value of *ptr == %d\n", *ptr); //ptr에 저장된 주소가 가리키는 값을 출력. 즉, i의 값 1234가 출력됨
    dptr = &ptr; /* dptr is now holding the address of ptr */
    printf("\n[checking values after dptr = &ptr] \n");
    printf("value of i == %d\n", i); //i의 값 출력
    printf("address of i == %p\n", &i); //i의 값 1234가 저장되어 있는 주소 출력 (변하지 않음)
    printf("value of ptr == %p\n", ptr); //ptr의 값 출력 즉, i의 주소가 출력됨
    printf("address of ptr == %p\n", &ptr); //ptr의 주소 출력
    printf("value of *ptr == %d\n", *ptr); //ptr에 저장된 주소가 가리키는 값을 출력. 즉, i의 값 1234가 출력됨
    printf("value of dptr == %p\n", dptr); //dptr의 값 출력 즉, ptr의 주소 출력
    printf("address of dptr == %p\n", &dptr); //dptr의 주소 출력

29 printf("address of ptr == %p\n", &ptr); //ptr의 주소 출력
30 printf("value of *ptr == %d\n", *ptr); //ptr에 저장된 주소가 가리키는 값을 출력. 즉, i의 값 1234가 출력됨
31 printf("value of dptr == %p\n", dptr); //dptr의 값 출력 즉, ptr의 주소 출력
32 printf("address of dptr == %p\n", &dptr); //dptr의 주소 출력
33 printf("value of *dptr == %p\n", *dptr); //dptr에 저장된 주소가 가리키는 값-> ptr의 값 -> i의 주소 출력
34 printf("value of **dptr == %d\n", **dptr); //dptr에 저장된 주소가 가리키는 값이 가리키는 값 출력 즉 i의 값 1234가 출력됨.
35 *ptr = 7777; /* changing the value of *ptr */
36 printf("\n[after *ptr = 7777] \n"); //포인터 변수 ptr를 사용해 ptr에 저장된 주소가 가리키는 값을 7777로 바꿈 즉, i값을 바꿨음.
37 printf("value of i == %d\n", i); //바뀐 i의 값 7777 출력
38 printf("value of *ptr == %d\n", *ptr); //ptr에 저장된 주소가 가리키는 값 즉 i의 값 7777 출력
39 printf("value of **dptr == %d\n", **dptr); //dptr에 저장된 주소가 가리키는 값이 가리키는 값 즉,i의 값 7777
40 **dptr = 8888; /* changing the value of **dptr */
41 printf("\n[after **dptr = 8888] \n"); //이중 포인터 dptr를 사용해 i의 값을 8888로 바꿈
42 printf("value of i == %d\n", i); //i의 값 출력 8888
43 printf("value of *ptr == %d\n", *ptr); //ptr에 저장된 주소가 가리키는 값 출력 즉, i값 8888이 출력됨.
44 printf("value of **dptr == %d\n", **dptr); //이중포인터 dptr이 가리키는 값이 가리키는 값 출력. 즉, i값 8888
45 return 0;
46 }

```

```
[----- [Gahyun KIM] [2020045070] -----]
[checking values before ptr = &i]
value of i == 1234
address of i == 0061FF1C
value of ptr == 00000000
address of ptr == 0061FF18

[checking values after ptr = &i]
value of i == 1234
address of i == 0061FF1C
value of ptr == 0061FF1C
address of ptr == 0061FF18
value of *ptr == 1234

[checking values after dptr = &ptr]
value of i == 1234
address of i == 0061FF1C
value of ptr == 0061FF1C
address of ptr == 0061FF18
value of *ptr == 1234
value of dptr == 0061FF18
address of dptr == 0061FF14
value of *dptr == 0061FF1C
value of **dptr == 1234

[after *ptr = 7777]
value of i == 7777
value of *ptr == 7777
value of **dptr == 7777

[after **dptr = 8888]
value of i == 8888
value of *ptr == 8888
value of **dptr == 8888
% Terminal will be reused by tasks, press
```

Git

<https://github.com/yahooit1/24homework2>