자료구조 2주차 과제

2019038059
소프트웨어학과
윤태경

github 주소

https://github.com/taegung/data-structures.git

lab2-1.c

```c
#include <stdio.h>
int main()
{
char charType;//char형 charType 변수 선언
int integerType;//int형 integerType 변수 선언
float floatType;//float형 floatType 변수 선언
double doubleType;//double형 doubleType 변수 선언
printf("[----- [윤태경] [2019038059] -----]");
printf("Size of char: %ld byte\n",sizeof(charType));// charType 변수의 크기 출력
printf("Size of int: %ld bytes\n",sizeof(integerType));//integerType 변수의 크기 출력
printf("Size of float: %ld bytes\n",sizeof(floatType));//floatType 변수의 크기 출력
printf("Size of double: %ld bytes\n",sizeof(doubleType));// doubleType 변수의 크기 출력
printf("----------------------------------------\n");
printf("Size of char: %ld byte\n",sizeof(char));//char자료형 크기 출력
printf("Size of int: %ld bytes\n",sizeof(int));//int 자료형 크기 출력
printf("Size of float: %ld bytes\n",sizeof(float));//float 자료형 크기 출력
printf("Size of double: %ld bytes\n",sizeof(double));//double 자료형 크기 출력
printf("----------------------------------------\n");
printf("Size of char*: %ld byte\n",sizeof(char*));//포인터 char자료형의 크기 출력
printf("Size of int*: %ld bytes\n",sizeof(int*));//포인터 int 자료형의 크기 출력
printf("Size of float*: %ld bytes\n",sizeof(float*));//포인터 float 자료형의 크기 출력
printf("Size of double*: %ld bytes\n",sizeof(double*));//포인터 double 자료형의 크기 출력
return 0;}
```

```
[----- [윤태경] [2019038059] -----]
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

터미널이 작업에서 다시 사용됩니다. 닫으려면 아무 키나 누르세요.
```

lab2-2.c

```c
#include <stdio.h>
int main()
{
int i;
int *ptr;
int **dptr;
i = 1234;
printf("[----- [윤태경] [2019038059] -----]\n");
printf("[checking values before ptr = &i] \n");
printf("value of i == %d\n", i);//i의 값 출력
printf("address of i == %p\n", &i);//i의 주소 출력
printf("value of ptr == %p\n", ptr);//포인터ptr의 값 출력
printf("address of ptr == %p\n", &ptr);//포인터ptr 주소 추력
ptr = &i; /* ptr is now holding the address of i */  //포인터 ptr에 i의 주소 입력
printf("\n[checking values after ptr = &i] \n");
printf("value of i == %d\n", i);//i 의 값을 출력
printf("address of i == %p\n", &i);//i의 주소 출력
printf("value of ptr == %p\n", ptr);//포인터 ptr 값출력 포인터 ptr값이 i의 주소
printf("address of ptr == %p\n", &ptr);//포인터 ptr의 주소 출력
printf("value of *ptr == %d\n", *ptr);//포인터 ptr이 가리키는 i의 값출력
dptr = &ptr; /* dptr is now holding the address of ptr */  //포인터 dptr에 포인터 ptr의 주소 입력
printf("\n[checking values after dptr = &ptr] \n");
printf("value of i == %d\n", i);//i의 값을 출력
printf("address of i == %p\n", &i);//i의 주소 출력
printf("value of ptr == %p\n", ptr);//포인터 ptr 값 출력
printf("address of ptr == %p\n", &ptr);//포인터 ptr 주소 출력
printf("value of *ptr == %d\n", *ptr);//포인터 ptr이 가리키는 i의 값 출력
printf("value of dptr == %p\n", dptr);//포인터 dptr 값 출력  포인터 dptr 값은 포인터 ptr의 주소
printf("address of dptr == %p\n", &dptr);//포인터 dptr 주소 출력
printf("value of *dptr == %p\n", *dptr);//포인터 dptr이 가리키는 포인터 ptr값 출력
printf("value of **dptr == %d\n", **dptr);//포인터 *dptr 가르키는 i의 값 출력
*ptr = 7777; /* changing the value of *ptr */ //i의 값 7777 수정
```

```c
printf("\n[after *ptr = 7777] \n");
printf("value of i == %d\n", i);//i의 값 출력
printf("value of *ptr == %d\n", *ptr);//포인터 ptr이 가리키는 i의 값 출력
printf("value of **dptr == %d\n", **dptr);//포인터 *dptr 가르키는 i의 값 출력
**dptr = 8888; /* changing the value of **dptr */ //i의 값 8888로 수정
printf("\n[after **dptr = 8888] \n");
printf("value of i == %d\n", i);//i의 값 출력
printf("value of *ptr == %d\n", *ptr);;//포인터 ptr이 가리키는 i의 값 출력
printf("value of **dptr == %d\n", **dptr);//포인터 *dptr 가르키는 i의 값 출력
return 0;}
```

```
> Executing task: cmd /C c:\Users\82109\Desktop\example\example <

[----- [윤태경] [2019038059] -----]
[checking values before ptr = &i]
value of i == 1234
address of i == 0061FF1C
value of ptr == 00390000
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

터미널이 작업에서 다시 사용됩니다. 닫으려면 아무 키나 누르세요.
```

Memory Layout으로 도식화