

문제해결기법 3주차 과제

202033762 장민호

-코드

```
#include <stdio.h>

#define MAX 10

void printArr(int *arr)
{
    for (int i = 0; i < MAX; i++)
    {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main()
{
    int p1 = 10;
    int *p2 = &p1;
    // 포인터 변수에는 어떠한 데이터값의 주소를 담을 수 있다.

    printf("p1  = %d\n", p1);
    printf("*p2 = %d\n", *p2);

    printf("&p1 = %d\n", &p1);
    printf("p2  = %d\n", p2);
}
```

```
printf("&p2  = %d\n", &p2);
```

```
printf("=====\n");
```

*p2 = &p1; // 재미난 일이 일어난다. p1에 들어가있던 10은 온데간데없고 p1의 주소값이 들어가 있다. 또 p2의 데이터값에 p1의 주소값이 들어가 있다.

```
printf("&p1 = %d\n", &p1);
```

```
printf("p1  = %d\n", p1);
```

```
printf("*p2 = %d\n", *p2);
```

```
printf("p2  = %d\n", p2);
```

```
printf("&p2 = %d\n", &p2);
```

// 배열

```
int arr[MAX] = {10, 3, 5, 7, 9, 2, 4, 6, 8, 0};
```

```
int *arr2;
```

```
// arr = arr2; unvaild.
```

```
arr2 = arr; // 배열의 이름은 포인터이다.
```

// arr2 = 100; 포인터가 100번 주소를 가리키라는 의미이다. 100번 주소에 무엇이 있을지도 모르는데 가리키는 건 매우 위험한 수식이다.

```
printf("arr's address: %p\n", &arr);
```

```
printf("arr2's address: %p\n", &arr2);
```

```
printf("arr2[2] = %d\n", arr2[2]);  
printf("*arr2 = %d\n", *arr2); // arr2의 가장 앞에  
있는 숫자가 나오게 된다.  
printf("*arr + 1 = %d\n", *arr + 15);  
  
printArr(arr);  
printArr(arr2);  
printf("++*arr= %d\n", ++*arr2);  
// arr와 arr2는 같은 배열을 가리키고 있는 상태이므로 하나  
를 변경하면 나머지 하나도 같이 변경됨을 볼 수 있다.  
printArr(arr);  
printArr(arr2);  
  
printf("*arr + 2 = %d\n", *arr + 2);  
// arr 포인터 주소값 맨 앞에 있는 숫자 = 11  
printf("++*arr= %d\n", *(arr2 + 3));  
  
return 0;  
}
```

-실행결과

```
p1 = 10
*p2 = 10
&p1 = -962563812
p2 = -962563812
&p2 = -962563808
=====
&p1 = -962563812
p1 = -962563812
*p2 = -962563812
p2 = -962563812
&p2 = -962563808
arr's address: 0x7fffc6a07130
arr2's address: 0x7fffc6a07128
arr2[2] = 5
*arr2 = 10
*arr + 1 = 25
10 3 5 7 9 2 4 6 8 0
10 3 5 7 9 2 4 6 8 0
++*arr= 11
11 3 5 7 9 2 4 6 8 0
11 3 5 7 9 2 4 6 8 0
*arr + 2 = 13
++*arr= 7
```