

## 6 장 포인터

---

### □ 개념 확인 학습

---

1. 포인터란 무엇인가요?
2. 포인터 변수 `int *p;`에서 `*p` 와 `p`의 차이점은 무엇인가요?
3. 포인터 변수의 크기는 어떻게 정해지나요?
4. 포인터의 형과 포인터가 가리키는 자료형은 왜 같아야 하나요?
5. 널 포인터(Null Pointer)란 무엇인가요?
6. 포인터 변수와 관련된 연산 식에 사용되는 연산자에는 어떤 것들이 있나요?
7. 포인터 연산과 일반적인 연산과 다른 점은 무엇인가요?
8. `(*p)++`; 의미는 무엇인가요?

9.  $*(p+1)$ 의 의미는 무엇인가요?
10. 포인터 변수를 이용하여 일반 변수의 내용을 접근하고자 할 때 반드시 필요한 작업은 무엇인가요?
11. 이차원 배열을 포인터로 접근하려면 포인터는 어떻게 선언되어야 하나요?
12. 포인터 배열이란 무엇인가요?
13. 포인터 배열을 사용할 때의 장점은 무엇인가요?
14. 다중 포인터란 어떤 의미인가요?

---

☐ 적용 확인 학습

---

1. 다음 서술 내용이 바르면 O, 그렇지 않으면 X를 표시 하세요.

- A. 일반 변수를 포인터를 사용하여 접근 하려면 반드시 자료형이 같은 포인터 변수에 주소를 저장한 후 사용해야 한다. ( )
- B. 배열의 이름은 배열의 주소이므로 배열을 포인터를 사용하여 접근 하려면 포인터 변수에 배열의 이름을 대입하면 된다. ( )
- C. 포인터 변수의 크기는 자료형에 따라 다르다. ( )
- D. ((void \*)0)란 아직 결정되지 않은 자료형의 주소를 나타낸다. ( )
- E. `int *p;`로 선언된 포인터 변수에서 `p++;`의 의미는 포인터 변수가 가리키고 있는 곳의 값을 1증가시키는 것이다. ( )
- F. `int *p;`로 선언된 포인터 변수에서 `p++;`을 수행하면 주소를 1byte 증가시킨다. ( )
- G. `int *p;` 선언된 포인터 변수에서 `*(p+1)`의 의미는 주소를 증가시켜 다음 원소를 가리킨 후, 그곳의 값을 참조한다. ( )
- H. 같은 자료형 타입의 배열과 포인터는 주소 치환이후 배열은 포인터처럼, 포인터는 배열처럼 사용할 수 있다. ( )
- I. 문자열 상수를 만들 때에는 배열을 이용하지 않고 포인터만을 이용해도 가능하다. ( )
- J. `int nums[3][5];`의 이차원 배열을 접근하기 위한 포인터의 선언은 `int(*p)[5];`의 형태를 가지며, 이 때 괄호가 반드시 필요한 것은 아니다. ( )

- K. 포인터 배열은 포인터들로 구성된 배열로 포인터 형의 변수가 배열의 요소가 되는 경우를 말한다. ( )
- L. 포인터 배열은 길이가 일정하지 않은 문자열 배열을 나타낼 때 메모리의 손실을 줄일 수 있는 장점이 있다. ( )
- M. 이중 포인터는 번지를 두 번 참조해서 원하는 자료가 저장된 주소를 찾아간다. ( )

2. 다음에 서술된 문장에 어울리는 프로그램 코드를 완성해 보세요.

- A. 정수형 변수에 사용자가 입력한 정수를 저장하고 포인터를 이용하여 출력한다.

```
int num, *p;
```

- B. 문자열형 변수에 사용자가 입력한 문자열을 저장하고 포인터를 이용하여 출력한다.

```
char str
```

- C. 정수배열 변수에 사용자가 입력한 정수 다섯 개를 저장하고 포인터를 이용하여 출력한다.

```
int num
```

- D. 문자배열 변수에 사용자가 입력한 문자 다섯 개를 저장하고 포인터를 이용하여 출력한다.

char str

- E. 문자열배열 변수에 사용자가 입력한 문자열 다섯 개를 저장하고 포인터를 이용하여 출력한다.

char str

---

#### □ 응용 프로그래밍

---

>> 응용 프로그래밍은 다음과 같은 구조로 작성합니다. 번호가 늘어 갈 경우 번호에 해당하는 함수를 만들어 사용합니다.

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

```
void ex_02(void) {
```

```
    //응용 프로그래밍 2번은 ex_02() 내부에 작성합니다
```

```
}
```

```
void ex_03(void) {
```

```
    //printf("응용 프로그래밍 3번은 ex_03() 내부에 작성합니다.
```

```
}
```

```
int main() {
```

```
    ex_02();
```

```
    ex_03();
```

```
    return 0;
```

```
}
```

1. double형 source 1차원 배열을 다음과 같이 초기화한 후, 포인터를 사용하여 target 배열에 복사하고, 두 배열의 내용을 출력하세요. 배열크기는 sizeof()로 계산합니다.

double source [] = {1.21, 3.43, 5.65, 7.87, 8.98}; 저장하고 출력하세요.

```
source = 1.21, 3.43, 5.65, 7.87, 8.98,
target = 1.21, 3.43, 5.65, 7.87, 8.98,
```

2. double source [] = {1.21, 3.43, 5.65, 7.87, 8.98}; 초기화 된 배열의 모든 원소에 사용자로부터 입력받은 값을 더해 출력하는 프로그램을 포인터를 이용하여 작성하세요.

```
source = 1.21, 3.43, 5.65, 7.87, 8.98,
source 각 원소에 더할 값 입력 : 0.1
target = 1.31, 3.53, 5.75, 7.97, 9.08,
```

3. 사용자로부터 문자열을 1개를 입력 받아 포인터를 이용하여 그 문자열의 길이를 출력하세요.

```
문자열입력 : apple
입력된 문자열 길이=5
```

4. 사용자로부터 문자열을 입력 받아 포인터를 이용하여 그 문자열의 길이를 출력하는 작업을 반복하세요. 단, <Enter키>만을 입력 받으면 반복을 종료합니다.

```
문자열입력 : apple
입력된 문자열 길이=5
문자열입력 : banana
입력된 문자열 길이=6
문자열입력 : orange
입력된 문자열 길이=6
문자열입력 : <Enter키>
```

5. 문자열과 문자를 입력 받아 문자열에서 문자가 몇 개 인지 출력하는 작업을 반복하세요. 단, 포인터를 이용해서 작성하며, <Enter키>만을 입력 받으면 반복을 종료합니다.

```
문자열입력 : coffee
문자입력 : f
    coffee에서 f는 2개.
문자열입력 : ice cream
문자입력 : h
    ice cream에서 h는 0개.
문자열입력 : <Enter키>
```

6. 문자열 2개를 입력 받아 포인터를 사용하여 입력 받은 두 문자열을 연결하세요.

```
문자열입력 : coffee
문자열입력 : ice cream
sa=coffee, sb=ice cream
sa=sa+sb=coffeeice cream, sb=ice cream
```

7. stra 배열에 소문자 문자열을 입력 받아 strb배열에 대문자로 변환하여 저장하는 작업을 포인터를 이용하여 작성하세요.

```
문자열입력 : coffee
stra=coffee, strb=COFFEE
```

8. 2차원 문자열 배열에 첫 번째 줄에 소문자 문자열을 입력 받아, 두 번째 줄에 대문자로 변환하여 저장하는 프로그램을 포인터를 이용하여 작성하세요.

```
문자열입력 : coffee
str[0]=coffee, str[1]=COFFEE
```

9. 문자열 2개를 입력 받아 두 문자열이 같은지 다른지 비교하는 프로그램을 포인터를 이용하여 작성하세요.

```
문자열입력 : apple
문자열입력 : apples
apple, apples, 다름.
```

10. 2차원 문자열 배열에 문자열 2개를 입력 받아 두 문자열이 같은지 다른지 비교하는 작업을 포인터를 이용하여 작성하세요. (단, 1차원 배열 사용금지)

```
문자열입력 : apple
문자열입력 : apples
apple, apples, 다름.
문자열입력 : apple
문자열입력 : apple
apple, apple, 같음.
문자열입력 : <Enter키>
```

11. 정수 4개를 저장하는 from배열을 선언하고 사용자로부터 정수를 입력 받아 다음과 같이 수행되는 프로그램을 포인터를 이용하여 작성하세요.

```
정수입력 : 2
정수입력 : 4
정수입력 : 5
정수입력 : 6

합=17
가장 큰 수= 6
가장 작은 수=2
```



12. 위 11번의 일차원 배열 from을 to배열에 포인터를 이용하여 복사하고 두 배열의 내용을 출력하세요.

```
정수입력 : 2
정수입력 : 4
정수입력 : 5
정수입력 : 6

from배열: 2, to배열:2
from배열: 4, to배열:4
from배열: 5, to배열:5
from배열: 6, to배열:6
```

13. 한 단어를 입력 받아, 단어를 역순으로 출력하는 프로그램을 포인터를 이용하여 작성하세요.

```
한 단어를 입력하세요 : rainy
입력한 단어를 반대로 출력합니다 : yniar
```