포인터

[1] 포인터의 개요

- 변수 : 수나 문자를 저장, 기억장소의 어느 위치에 대한 이름, 주소
- 포인터 : 변수의 주소

[2] C문제

```c
#include <stdio.h>
main() {
    int a = 50;
    int *b = &a;
    *b = *b + 20;
    printf("%d, %d\n", a, *b);
    char *s;
    s = "gilbut";
    for (int i = 0; i < 6; i += 2) {
        printf("%c, ", s[i]);
        printf("%c, ", *(s + i));
        printf("%s\n", s + i);
    }
}
```

i => 0, 2, 4

풀이

```c
#include <stdio.h>
main() {
    int a = 50;  // 정수형 a 를 선언 50으로 초기화
    int *b = &a; // 정수형 포인터변수 b를 선언, a의 주소로 초기화
    *b = *b + 20; // b가 가리키는 곳에 (*b)에 20을 더한다 => a의 값도 바뀐다
  // b가 가리키는 곳의 값에 20을 더해 다시 b가 가리키는 곳에 저장, 그곳은 변수 a의 주소
  // 변수 a의 값도 저정로 변경
    printf("%d, %d\n", a, *b);  // a= 70, *b = 70
    char *s;  // s는 주소를 기억하는 포인터 변수
    s = "gilbut";  // s는 gilbut을 기억하는 것이 아니라 주소를 기억
  // s[0]=g, s[1]=i, s[2]=l, s[3]=b, s[4]=u, s[5]=t
    for (int i = 0; i < 6; i += 2) { // i = 0, 2, 4
        printf("%c, ", s[i]); // s[0]=g , s[2]=l, s[4]=u
        printf("%c, ", *(s + i)); //*(s + 0)=g, *(s + 2)=l, *(s + 4)=u
        printf("%s\n", s + i);  // s + 0 = gilbut, s + 2 = lbut, s + 4 = ut
    // (s+i)즉 (s+0)의 위치부터 문자열의 끝('\0')까지 모든 문자를 하나의 문자열로 출력
    }
}
```

결과

```
70, 70
g, g, gilbut
```

```
l, l, lbut
u, u, ut
```

기출 따라잡기

문제 1

```c
#include <stdio.h>
main() {
    char *p = "KOREA";
    printf("%s\n", p);
    printf("%s\n", p + 3);
    printf("%c\n", *p);
    printf("%c\n", *(p + 3));
    printf("%c\n", *p + 2);
}
```

풀이

```c
#include <stdio.h>
main() {
    char *p = "KOREA";  // p라는 변수에 KOREA 문자의 위치의 주소를 기억
    printf("%s\n", p);  // KOREA
    printf("%s\n", p + 3); // EA
    printf("%c\n", *p);  // K , *P 문자열에 첫번째 요소[0]를 가리킴
    printf("%c\n", *(p + 3)); // E  ,
    printf("%c\n", *p + 2); // M
}
```

결과

```
KOREA
EA
K
E
M
```

문제 2

```c
#include <stdio.h>
int main() {
    int ary[3];
    int s = 0;
    *(ary + 0) = 1;
    ary[1] = *(ary + 0) + 2;
    ary[2] = *ary + 3;
    for (int i = 0; i < 3; i++)
        s = s + ary[i];
```

```
    printf("%d", s);
}
```

풀이

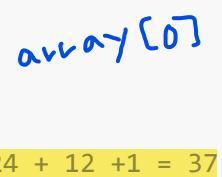
```
#include <stdio.h>
int main() {
    int ary[3];   // 크기가 3인 ary배열선언
    int s = 0;
    *(ary + 0) = 1; // ary배열 첫번째 요소ary[0]의 값은 1
    ary[1] = *(ary + 0) + 2;  // ary[1] = 3
    ary[2] = *ary + 3;    // ary[2] = 1 + 3 = 4
  // ary[0]=1, ary[1]=3, ary[2]=4
    for (int i = 0; i < 3; i++) // i = 0, 1, 2
        s = s + ary[i];  // 누적합계 1 + 3 + 4 = 8
    printf("%d", s);  // 8
}
결과 : 8
```

문제 3

```
#include <stdio.h>
int main() {
    int* array[3];
    int a = 12, b = 24, c = 36;
    array[0] = &a;
    array[1] = &b;
    array[2] = &c;
    printf("%d", *array[1] + **array + 1);
}
```

풀이

```
#include <stdio.h>
int main() {
    int* array[3];
    int a = 12, b = 24, c = 36;
    array[0] = &a;  // array[0] = 12
    array[1] = &b;  // array[1] = 24
    array[2] = &c;  // array[2] = 36
    printf("%d", *array[1] + **array + 1); // 24 + 12 +1 = 37
  // *array[1] : array[1]의 주소의 값= 24
  // *array => array변수의 주소의 값 => &array[0]
  // **array => *(&array[0]) => array[0] => 12
}
결과 : 37
```



문제 4

```c
#include <stdio.h>
int main( ) {
    int a[4] = { 0, 2, 4, 8 };
    int b[3];
    int* p;
    int sum = 0;
    for (int i = 1; i < 4; i++) {
        p = a + i;
        b[i - 1] = *p - a[i - 1];
        sum = sum + b[i - 1] + a[i];
    }
    printf("%d", sum);
}
```

풀이

```c
#include <stdio.h>
int main( ) {
    int a[4] = { 0, 2, 4, 8 };
    int b[3]; // 크기가 3인 b배열 배열선언
    int* p; // 포인트 정수형 p변수 선언
    int sum = 0;
    for (int i = 1; i < 4; i++) { // i = 1, 2, 3
        p = a + i;  // p = a + 1, a + 2, a + 3
        b[i - 1] = *p - a[i - 1]; // b[0] = *p -a[0], b[1]=*p-a[1], b[2]=*p-a[2]
        sum = sum + b[i - 1] + a[i];
    }
    printf("%d", sum);  // 22
}
결과 : 22
```

문제 5

```c
#include <stdio.h>
main() {
    char* a = "qwer";
    char* b = "qwtety";
    for (int i = 0; a[i] != '\0'; i++)
        for (int j = 0; b[j] != '\0'; j++)
            if (a[i] == b[j])
                printf("%c", a[i]);
}
```

풀이

```
#include <stdio.h>
main() {
    char* a = "qwer"; // 문자형 포인트 변수  a
    char* b = "qwtety"; // 문자형 포인트 변수 b
    for (int i = 0; a[i] != '\0'; i++)
        for (int j = 0; b[j] != '\0'; j++)
            if (a[i] == b[j])  // a[i]위치의 값이 b[i]의 문자와 같으면
                printf("%c", a[i]);   // a[i]위치의 값을 출력한다
}
결과 : qwe
```