

문제 1번 [숙제 목록보기]

(전역변수를 사용하지 말고 풀어주세요)

`input()` 함수는

숫자 1개를 입력받고 그 숫자를 `return`하는 함수입니다.

`main()` 함수에서는

`input`함수를 3번 호출하여 숫자 3개를 입력받아주세요.

그리고 그 숫자 3개를 `cals`함수로 값을 보내주세요

ex)

```
a = input();
```

```
b = input();
```

```
c = input();
```

`cals()` 함수에서는

숫자 3개를 받고 그 합을 출력해주는 함수입니다

ex) `cals(a, b, c);`

[TIP]

1. KFC 함수로 값을 보내는 방법

- 함수 정의할 때

```
void KFC(int x, int y)
```

```
{
```

```
    //...
```

```
}
```

- 함수 호출할 때

```
ABC(3, 5); //숫자 3과 5를 ABC 함수로 보낸다.
```

*이 예제는 값 2개 보냈습니다. 호출할 때는 여러 값을 보낼 수 있습니다.

* `void KFC(int x, int y)` 에서 "`void`"의 의미는 `return` 할 값이 없을 때 사용합니다

2. KFC함수에서 값을 return 받는 방법

- 함수 정의할 때

```
int KFC()
{
    int x = 5;
    return x;
}
```

- 함수 호출할 때

```
int a = KFC(); //KFC함수의 return값을 변수 a에 저장한다
```

* int KFC() 에서 "int"의 의미는 숫자 형태의 값을 return 하겠다 라는 뜻입니다.

* 값을 보낼때와 달리, 값을 되돌려줄때는 오직 값 1개만 return할 수 있습니다.

3. KFC함수에서 값을 보내고, 결과를 받는 방법

- 함수 정의할 때

```
int KFC(int a, int b)
{
    return a + b;
}
```

- 함수 호출할 때

```
int t = KFC(1, 2); // 숫자 2개를 보내고, 숫자 1개를 return받는 코드
```

입력 예시

1 1 2

출력 예시

4

```
#include <iostream>
using namespace std;

int input()
{
    int num = 0;
    cin >> num;
    return num;
}

void cals(int num1, int num2, int num3)
{
    int sum = 0;
    sum = num1 + num2 + num3;
    cout << sum;
}

int main(void)
{
    int a = 0;
    int b = 0;
    int c = 0;
    a = input();
    b = input();
    c = input();
    cals(a, b, c);

    return 0;
}
```

문제 2번 [숙제 목록보기]

[전역변수를 사용하지 말고 풀어주세요]

다음과 같은 함수를 만들어주세요

sum(int a, int b) 함수는 숫자 2개를 받고 그 합을 return 하는 함수입니다.

comp(int a, int b) 함수는 숫자 2개를 받고 그 차를 return 하는 함수입니다.

print(int t1, int t2) 함수는 숫자 2개를 받고 그 받은 숫자 2개를 출력하는 함수입니다.

main함수에서는 숫자 2개를 입력 받으세요

그리고 sum함수, comp함수를 호출해서 입력 받은 두 수의 합과 차를 구해주세요

그리고 print함수에 구한 합과 차를 보내고, 그 결과를 출력 해 주세요

입력 예시

5 1

출력 예시

합:6

차:4

```
#include <iostream>
using namespace std;

int sum(int a, int b)
{
    int sum = 0;
    sum = a + b;
    return sum;
}

int comp(int a, int b)
{
    int sub = 0;
    sub = a - b;
    if (sub < 0)
        sub *= (-1);
    return sub;
}

void print(int t1, int t2)
{
    cout << "합:" << t1 << endl;
    cout << "차:" << t2;
}

int main(void)
{
    int num1 = 0;
    int num2 = 0;
    cin >> num1 >> num2;
    print(sum(num1, num2), comp(num1, num2));

    return 0;
}
```

문제 3번 [숙제 목록보기]

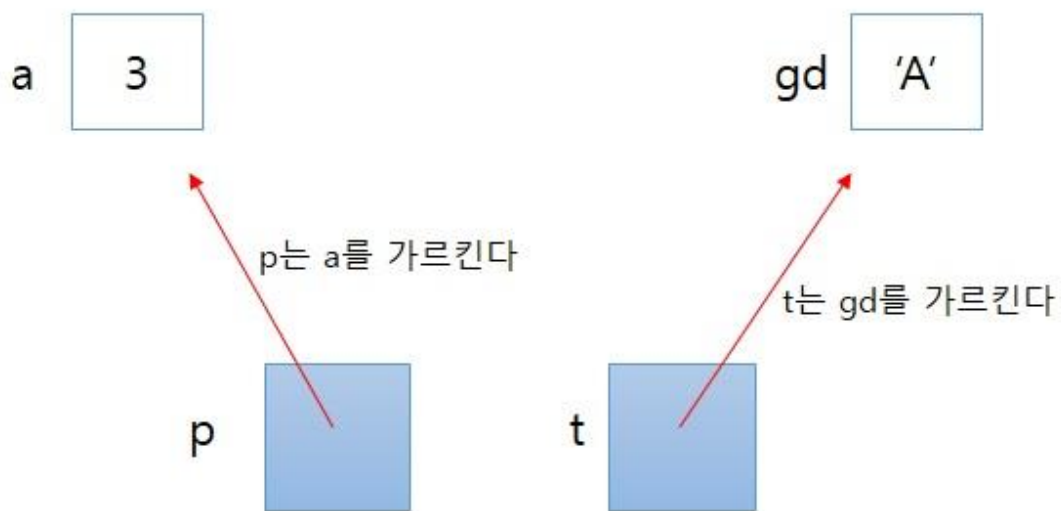
이번 문제를 통해 포인터 개념을 한번 더 확실히 익혀봅시다.

꼼꼼히 읽어보면서 아래 문제를 풀어보고, 스스로 포인터 연습도 해 보세요

아래 그림과 같이 int형(int type) a와 char형 gd 변수를 만들고

숫자1개, 문자1개를 입력 받아 각각의 변수 안에 넣어주세요

아래는 숫자3과 문자 'A'를 입력받았을 때 그림입니다



그리고

int pointer형 변수 p를 선언하고 변수 a를 가르키도록 해 주세요

char pointer형 변수 t를 선언하고 변수 gd를 가르키도록 해 주세요

이제

포인터 변수를 이용해서 가르키고 있는 곳의 값을 출력 해 주세요

아래와 같이 출력하시면 됩니다

```
cout << *p;
```

```
cout << *t;
```

[TIP] "가르킨다" 라는 의미 (Teaching이 아닙니다 / Pointing 입니다)

포인터 변수 에다가 다른 변수의 주소를 넣는 것을 "가르킨다" 라고 표현합니다.

예로들어

```
int x = 10;
```

```
int *p; // integer pointer형 p를 선언
```

```
p = &x; // p 에다가 x의 주소를 넣음
```

변수 이름 앞에다가 &(앰퍼센드)를 붙이면 변수의 메모리 주소값을 나타냅니다.

이 주소값을 p라는 변수에다가 집어 넣은 것이죠

그리고 이 동작을 한번에 처리할 수 도 있습니다.

```
int *p = &x; // integer pointer형 p를 선언하고, x를 가르키게 한 코드
```

[TIP] 포인터와 일반 변수의 차이

포인터 변수나 일반 변수나 똑같은 변수입니다. 어떤 값을 저장하는 변수입니다.

int형 변수는 정수형 숫자값을 저장하고

char형 변수는 문자값을 저장하고

integer pointer형 변수는 int형 변수의 주소값을 저장하는 변수입니다

포인터 변수는 일반 변수와 달리 특별한 기능을 가지고 있습니다.

포인터가 어떤 변수를 가르키고 있을 때 (변수의 주소를 저장하고 있을 때)

포인터 앞에 *을 붙이면

가르키고 있는 변수를 원격 조정이 가능해지죠

예로 들자면

```
int x = 10;
```

```
int *p = &x; //p는 x를 가르킨다
```

```
*p = 10; // 능력사용! p가 가르키고 있는 곳의 값에다가 10을 넣는다.
```

```
(*p) ++; // 능력사용! p가 가르키고 있는 곳의 값에다가 1 더한다.
```

```
cout << *p; //능력사용! p가 가르키고 있는 곳의 값을 출력한다.
```

```
cout << x; //변수 x 출력
```

Trace를 해보면서 연습해보세요

입력 예시

3 A

출력 예시

3 A

```
#include <iostream>
using namespace std;

int main(void)
{
    int a = 0;
    char gd = 'WO';
    cin >> a >> gd;

    int* p = nullptr;
    char* t = nullptr;
    p = &a;
    t = &gd;

    cout << *p << " ";
    cout << *t;

    return 0;
}
```

이름	값	형식
a	3	int
gd	65 'A'	char
p	0x000000f78139f5c4 {3}	int *
	3	int
t	0x000000f78139f5e4 <문자열에 잘못된 문자가 있습니다.>	char *
	65 'A'	char

문제 4번 [숙제 목록보기]

문자 변수 a,b,c를 만들고 문자 3개를 입력 받으세요

그리고 char pointer형 변수 3개를 만들고 각각 문자 변수들을 가르켜주세요. (pointing)

pointer의 능력을 사용해서

각 문자의 다음 알파벳을 출력 해 주세요

[HINT] pointer 능력 사용해서 가르키고 있는 변수 값 원격으로 바꾸기

```
int *p = &x;
```

```
*p = 10;
```

이렇게 하면 변수 x에 값이 10이 들어가게 됩니다

[HINT2] 가르키는 곳에 ++ 하는 방법

```
int a = 10;
```

```
int *p = &a;
```

```
(*a)++;
```

주의 : *a++; 하면 안됩니다. (*a)++; 이렇게 해야 합니다.

*a++; 이것은 *(a++); 이런 뜻이며, a 주소 값에 1을 더한다는 뜻입니다.

입력 예시

A B C

출력 예시

B C D

```
#include <iostream>
using namespace std;

int main(void)
{
    char a = 'WO';
    char b = 'WO';
    char c = 'WO';
    cin >> a >> b >> c;

    char* pA = &a;
    char* pB = &b;
    char* pC = &c;

    cout << ++*pA << " ";
    cout << ++(*pB) << " ";
    cout << ++(*pC);

    return 0;
}
```


▲ pA	0x00000059dc8ff574 <문자열에 잘못된 문자가 있습니다.>	Q 보기 ▼	char *
	97 'a'		char
▲ pB	0x00000059dc8ff594 <문자열에 잘못된 문자가 있습니다.>	Q 보기 ▼	char *
	98 'b'		char
▲ pC	0x00000059dc8ff5b4 <문자열에 잘못된 문자가 있습니다.>	Q 보기 ▼	char *
	99 'c'		char

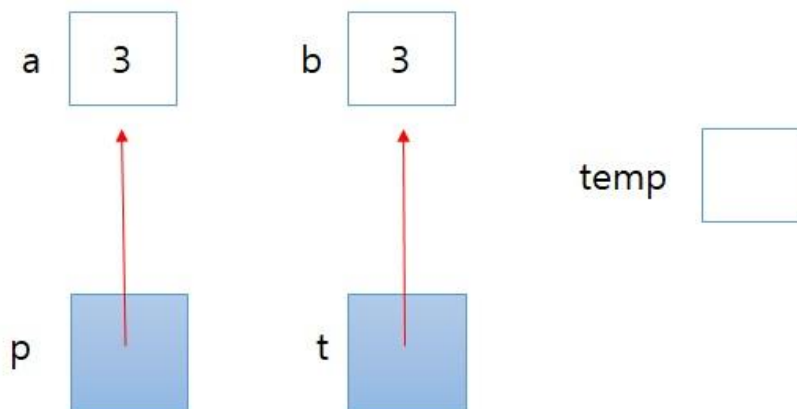
문제 5번 [숙제 목록보기]

int형 변수 a, b 에 숫자 2개를 입력 받으세요

그리고 int pointer형 변수 p, t를 선언하고 각각 a, b를 가르켜주세요.(pointing)

이제 변수 a와 b를 Swap하려고 합니다.

*p와 *t를 이용해서 두 값을 SWAP 하고 변수 *p와 *t를 출력 해 주세요



[HINT]

```
int temp;
```

temp변수와 *p, *t 를 가지고 SWAP을 할 수 있습니다

입력 예시

3 6

출력 예시

6 3

```
#include <iostream>
using namespace std;

int main(void)
{
    int a = 0;
    int b = 0;
    cin >> a >> b;

    int* p = &a;
    int* t = &b;

    int temp = 0;
    temp = *p;
    *p = *t;
    *t = temp;

    cout << *p << " " << *t;

    return 0;
}
```

문제 6번 [숙제 목록보기]

3	4	1	3	2	7	3
---	---	---	---	---	---	---

위 배열을 하드코딩 해 주세요

그리고 숫자 하나를 입력받아주세요

그 숫자가 존재하면 "발견" 출력

존재하지 않으면 "미발견" 출력

counting방식으로 풀지 말고, flag방식으로 문제를 풀어주세요

[TIP] Counting VS Flag

Counting방식은 내가 원하는 값이 몇개 있는지 세는 방식입니다

Flag방식은 스위치 변수를 하나 두고, 원하는 값이 발견되면 스위치를 켜는 방식입니다.

물론 Flag대신 Counting방식으로도 코딩할 수 있지만,

몇 개인지 셀때는 Counting

존재여부를 검사할때는 Flag

가독성을 위해 Flag를 쏘 때와 Counting을 할 때를 구분해서 써 주세요

입력 예시

3

출력 예시

발견

```
#include <iostream>
using namespace std;

int main(void)
{
    int arr[7] = { 3, 4, 1, 3, 2, 7, 3 };
    int input = 0;
    cin >> input;

    bool isExit = false;
    for (int i = 0; i < 7; ++i)
    {
        if (arr[i] == input)
        {
            isExit = true;
            break;
        }
    }
    if (isExit)
        cout << "발견";
    else
        cout << "미발견";

    return 0;
}
```

문제 7번 [숙제 목록보기]

7칸짜리 int형 배열을 만들어주세요.

그리고 이 배열에 숫자 7개를 입력받아주세요

예로들어 4 1 5 2 3 2 2 를 입력받았다고하면

4	1	5	2	3	2	2
---	---	---	---	---	---	---

이렇게 채워질 수 있습니다.

이 상태에서 MAX값과 MIN값을 구해서 출력 해 주세요

입력 예시

4 1 5 2 3 2 2

출력 예시

MAX=5

MIN=1

```
#include <iostream>
using namespace std;

int main(void)
{
    int arr[7] = {};
    for (int i = 0; i < 7; ++i)
        cin >> arr[i];
    int max = arr[0];
    int min = arr[0];

    for (int i = 0; i < 7; ++i)
    {
        if (arr[i] > max)
            max = arr[i];
        if (arr[i] < min)
            min = arr[i];
    }

    cout << "MAX=" << max << endl;
    cout << "MIN=" << min << endl;
    return 0;
}
```

문제 8번 [숙제 목록보기]

다음 문장을 배열에 하드코딩 해 주세요

"StructPointer"

그리고 문자 하나를 입력 받으세요

그 문자가 만약 존재하면 "발견" 출력

존재하지않으면 "미발견" 출력

입력 예시

P

출력 예시

발견

```
#include <iostream>
using namespace std;

int main(void)
{
    char arrCh[13] = { 'S', 't', 'r', 'u', 'c', 't', 'P', 'o', 'i', 'n', 't', 'e', 'r' };
    char input = 'W';
    cin >> input;

    for (int i = 0; i < 13; ++i)
    {
        if (arrCh[i] == input)
        {
            cout << "발견";
            return 0;
        }
    }
    cout << "미발견";

    return 0;
}
```

문제 9번 [숙제 목록보기]

for문을 돌려 문자 8개를 입력 받아주세요

만약 d T e a A B f a 를 입력했다면 아래와 같이 배열이 만들어집니다

d	T	e	a	A	B	f	a
---	---	---	---	---	---	---	---

그리고 big, small이라는 8칸 짜리 배열 2개를 더 만들어주세요

만들어진 배열을 탐색하면서 대문자는 **big 배열**에 채워주세요

만들어진 배열을 탐색하면서 소문자는 **small 배열**에 채워주세요

big배열

T	A	B					
---	---	---	--	--	--	--	--

small 배열

d	e	a	f	a			
---	---	---	---	---	--	--	--

big 배열과 small 배열을 출력 해 주시면 됩니다.

입력 예시

d T e a A B f a

출력 예시

big=TAB

small=deafa

```
#include <iostream>
using namespace std;

int main(void)
{
    char arrCh[8] = {};
    for (int i = 0; i < 8; ++i)
    {
        cin >> arrCh[i];
    }

    char big[8] = {};
    char small[8] = {};
    int cntBig = 0;
    int cntSmall = 0;

    for (int k = 0; k < 8; ++k)
    {
        if (arrCh[k] >= 'A' && arrCh[k] <= 'Z')
        {
            big[cntBig] = arrCh[k];
            ++cntBig;
        }
        else if (arrCh[k] >= 'a' && arrCh[k] <= 'z')
        {
            small[cntSmall] = arrCh[k];
            ++cntSmall;
        }
    }
    cout << "big=";
    for (int m = 0; m < cntBig; ++m)
    {
        cout << big[m];
    }
    cout << endl;

    cout << "small=";
    for (int n = 0; n < cntSmall; ++n)
    {
        cout << small[n];
    }

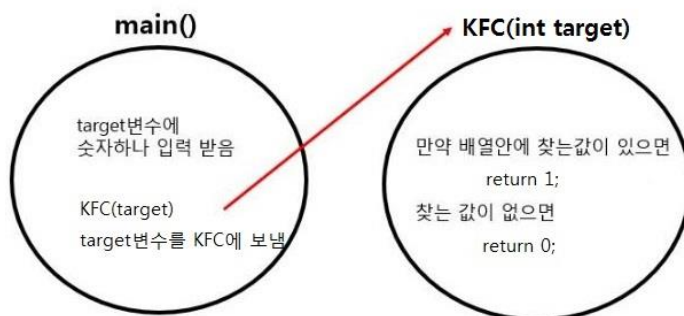
    return 0;
}
```

문제 10번 [숙제 목록보기]

전역배열 vect[2][5] 를 하드코딩 해주세요.

3	2	6	2	4
1	4	2	6	5

그리고 아래 그림과 같이 코딩을 해 주세요.



main에서는 KFC함수를 호출하고, 그 결과를 return 받습니다.

return 받은 값이 1이라면 "값이 존재합니다" 를 출력

return 받은 값이 0이라면 "값이 없습니다"를 출력

[TIP] return 명령어에 대해서1

return 명령어는 함수를 종료시키면서 호출했던 곳으로 돌아가는 명령어입니다

void 함수에서는 return 명령어가 없어도 되지만, 일부러 넣는 경우들이 있습니다. (함수를 강제로 종료할 때)

예로들어

```
void KFC( )
{
    cout << "#"; //실행 됨
    return; //실행 됨
    cout << "@"; //실행안되는 코드
}
```


여기서 KFC함수를 호출하면 #@가 뜨지않고, #만 뜨고 KFC함수가 바로 종료되게 됩니다.

그리고 KFC 함수는 void로 시작하는 함수이기 때문에 호출 한 함수로 돌려줄 값이 없습니다.

[TIP2] return 명령어에 대해서2

return 명령어는 무조건 맨 뒤에 있어야 하는 것이 아닙니다.

소스코드 중간 어디에서나 집어넣을 수 있고, 여러개 집어넣어도 됩니다.

함수를 즉시 종료 시킬 때 사용합니다.

```
int isCheck(int g)
{
    if (g == 1)
    {
        return 0;
    }

    cout << "#";

    return 100;
}
```

이 소스코드는

isCheck라는 함수에 숫자1을 보낼경우 --> return 0을 수행하고 바로 함수를 끝냅니다.

isCheck라는 함수에 숫자5를 보낼경우 --> #을 출력하고, return 100을 수행하고 바로 함수를 끝냅니다.

입력 예시

2

출력 예시

값이 존재합니다

```

#include <iostream>
using namespace std;

int vect[2][5] =
{
    {3,2,6,2,4},
    {1,4,2,6,5}
};

int KFC(int target)
{
    for (int y = 0; y < 2; ++y)
    {
        for (int x = 0; x < 5; ++x)
        {
            if (vect[y][x] == target)
                return 1;
        }
    }
    return 0;
}

int main(void)
{
    int input = 0;
    cin >> input;
    int result = 0;
    result = KFC(input);
    if (result == 1)
        cout << "값이 존재합니다.";
    else if (result == 0)
        cout << "값이 없습니다.";
    else
    {
        // nothing
    }

    return 0;
}

```

문제 11번 [숙제 목록보기]

아래와 같이 2차 배열을 하드 코딩 해 주세요

1	3	6	2
4	2	4	5

6	3	7	3
1	5	4	6

숫자 1개를 입력 받아주세요

입력받은 숫자보다 더 큰 값을 찾아서, 16칸 짜리 select 배열에 값을 채우는 문제입니다

만약 숫자 2를 입력 받았다면 select 배열에는 아래와 같이 채워집니다

3	6	4	4	5	6	3	7	3	5	4	6				
---	---	---	---	---	---	---	---	---	---	---	---	--	--	--	--

만약 숫자 4를 입력 받았다면 select 배열에는 아래와 같이 채워집니다

6	5	6	7	5	6										
---	---	---	---	---	---	--	--	--	--	--	--	--	--	--	--

입력 예시

4

출력 예시

6 5 6 7 5 6

```
#include <iostream>
using namespace std;

int main(void)
{
    int arrNum2D[4][4] =
    {
        {1,3,6,2},
        {4,2,4,5},
        {6,3,7,3},
        {1,5,4,6}
    };
    int select[16] = {};
    int cntSelect = 0;
    int input = 0;
    cin >> input;

    for (int y = 0; y < 4; ++y)
    {
        for (int x = 0; x < 4; ++x)
        {
            if (arrNum2D[y][x] > input)
            {
                select[cntSelect] = arrNum2D[y][x];
                ++cntSelect;
            }
        }
    }

    for (int i = 0; i < cntSelect; ++i)
        cout << select[i] << " ";
    return 0;
}
```