Programming Practice

2018-10-04

Week 5

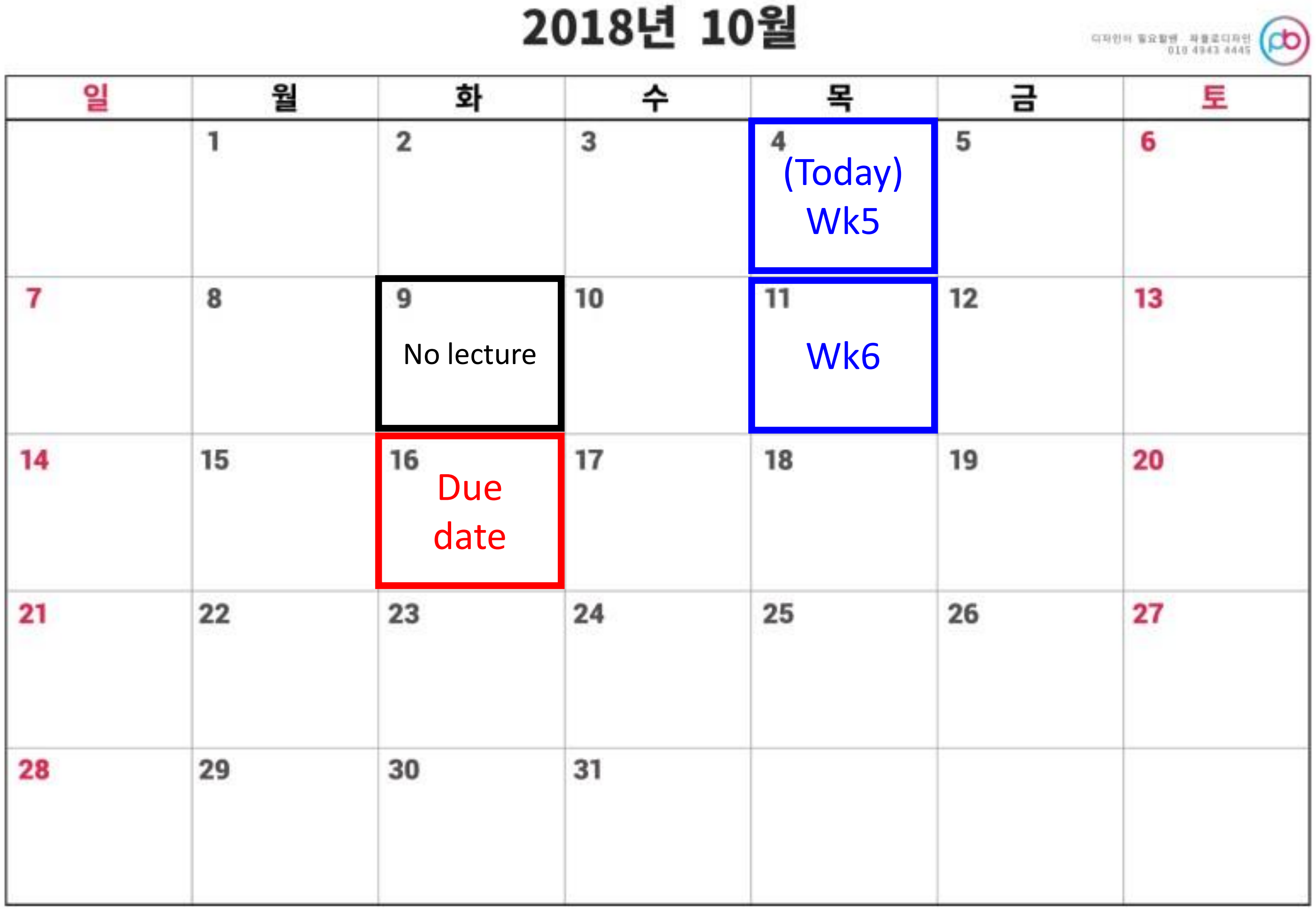
Notice

NOTICE

Due date:

Lab Wk5 & Wk6 (both)

10/16(Tue) 14:00



2018년 10월

일	월	화	수	목	금	토
	1	2	3	4 (Today) Wk5	5	6
7	8	9 No lecture	10	11 Wk6	12	13
14	15	16 Due date	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Practice Lecture

Floating-point Arithmetic

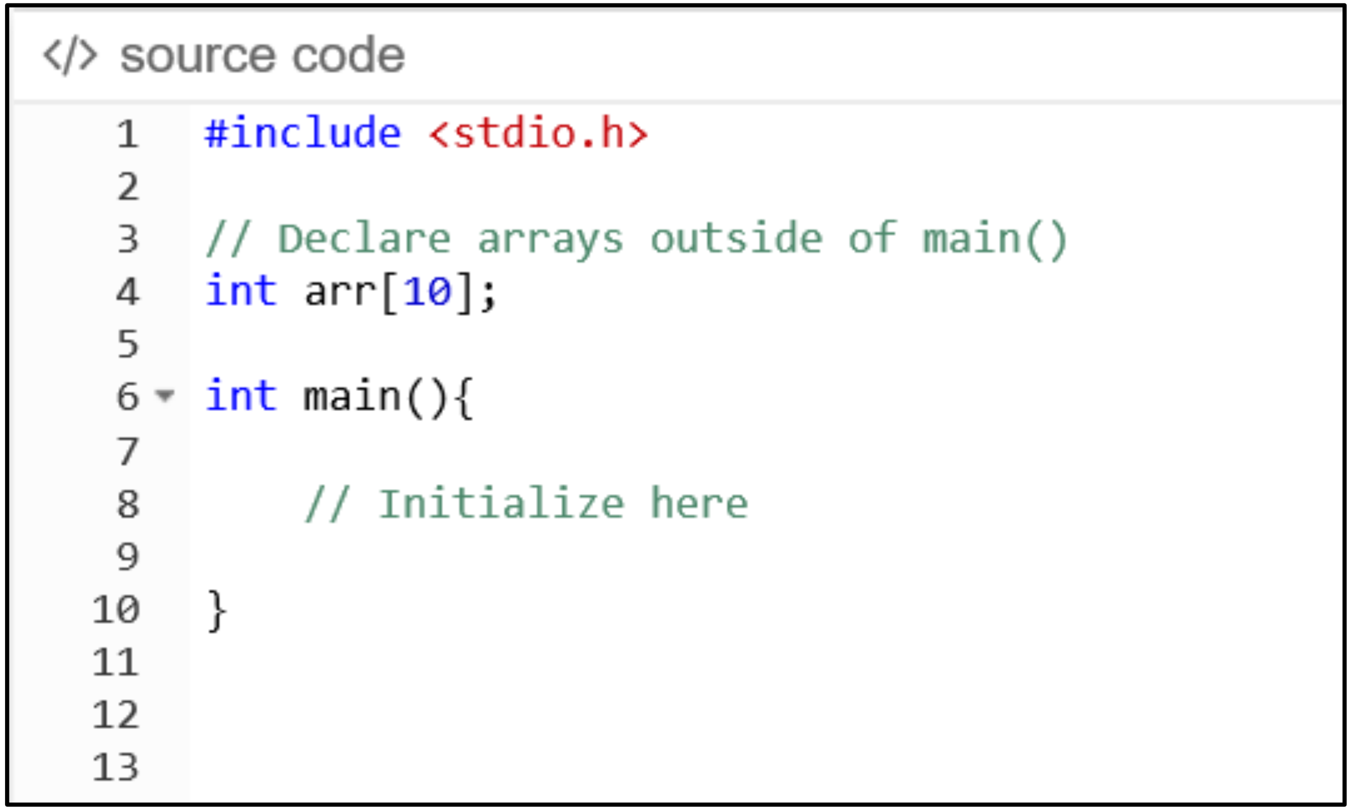
- Use `double`, rather than `float`

  (Only use `float` when e.g. memory limitations are very harsh.)

  (Same goes for other programming languages like C++, Java, etc.)

- Cf. `double` is short for “double-precision floating-point data type”

Declaring an array



```c
#include <stdio.h>

// Declare arrays outside of main()
int arr[10];

int main(){

    // Initialize here

}
```

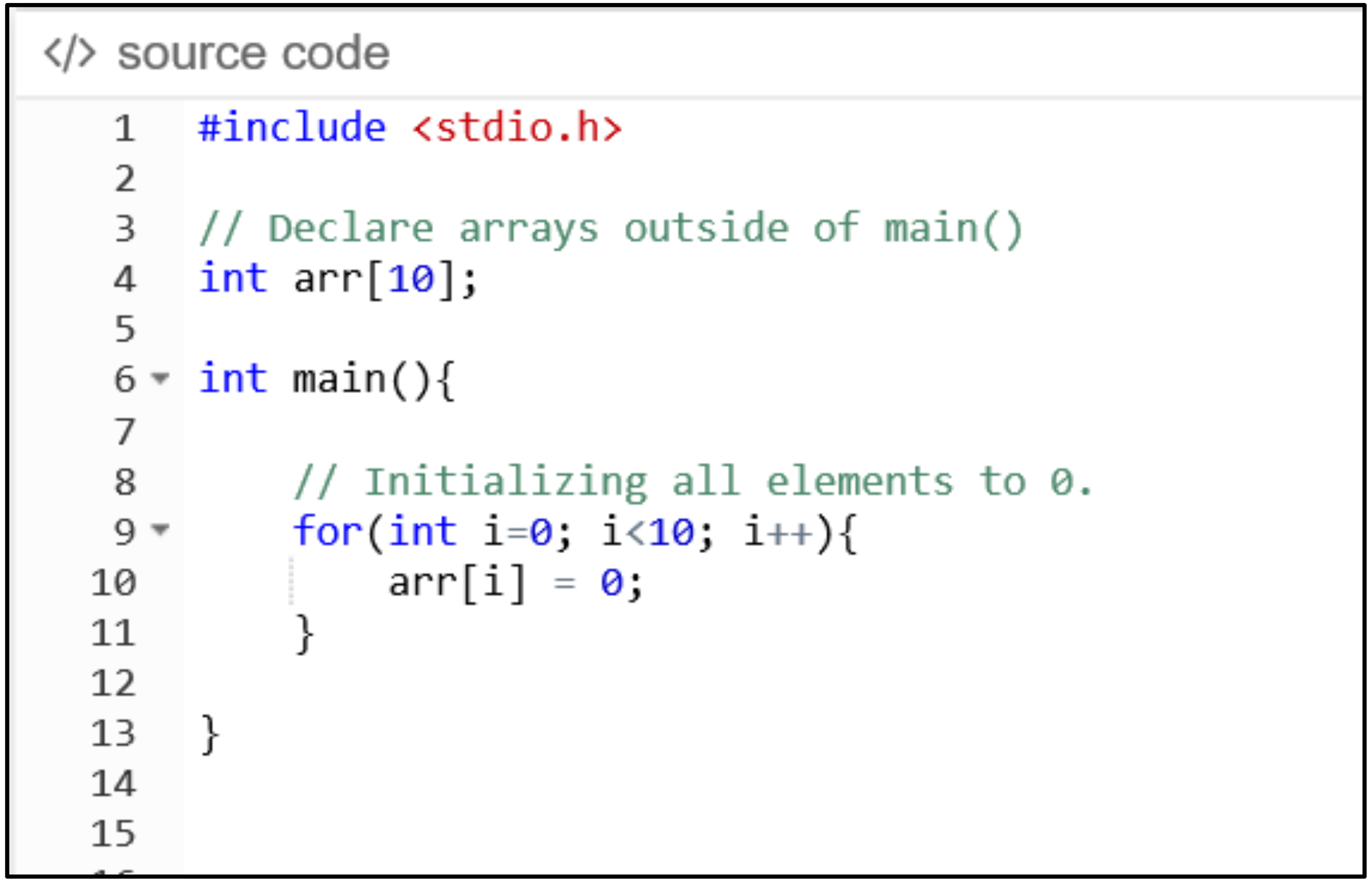
Declaring arrays (Ex.)

```c
int a[10];
char b[5];
double c[100];
```

- Declare outside of `main()` for now..

(global vs. local variables)

Initializing an array (1)

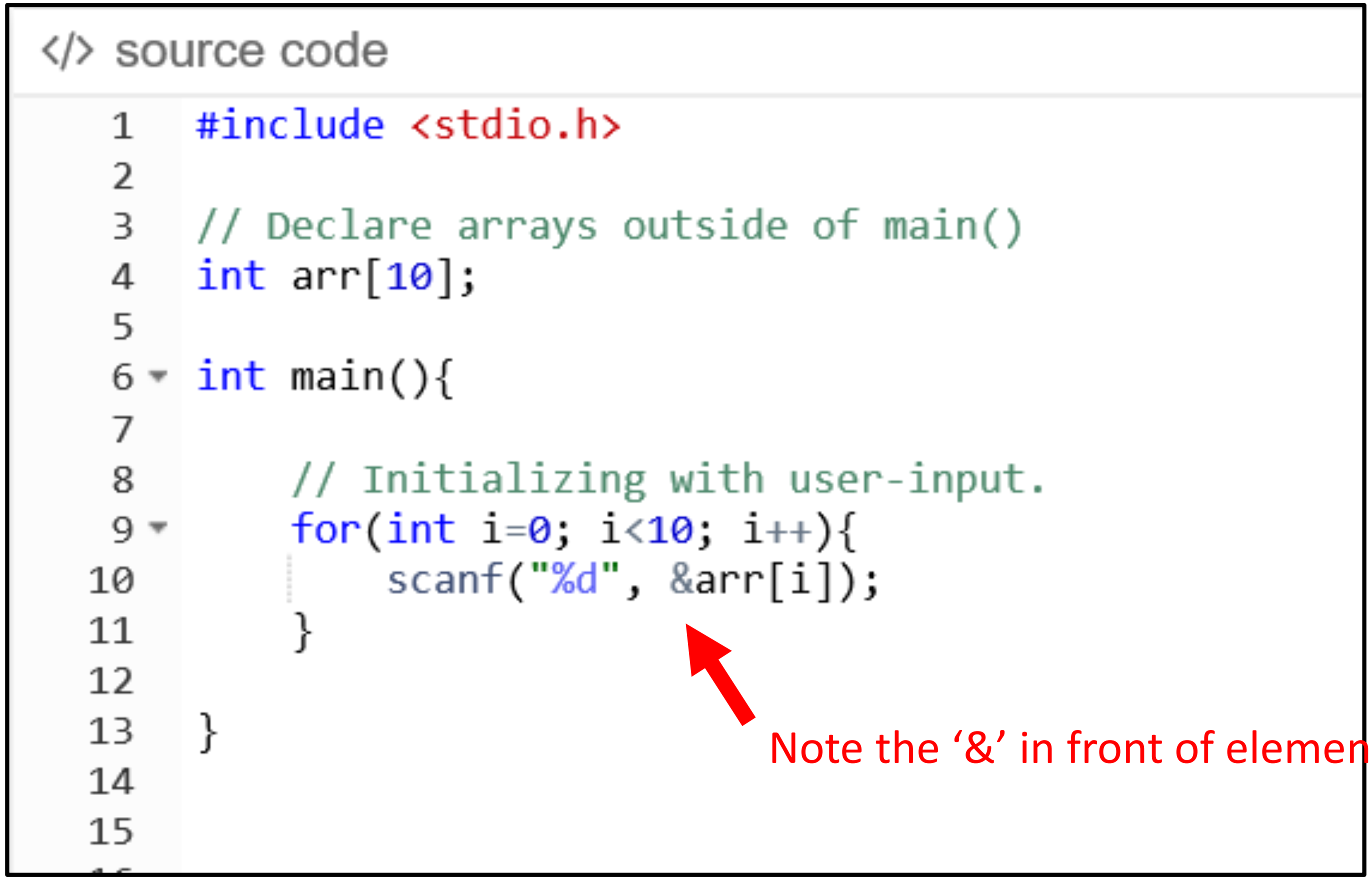


</> source code

```c
#include <stdio.h>

// Declare arrays outside of main()
int arr[10];

int main(){

    // Initializing all elements to 0.
    for(int i=0; i<10; i++){
        arr[i] = 0;
    }

}


```

- Use for-loop to write/read array elements

Initializing an array (2)



```c
#include <stdio.h>

// Declare arrays outside of main()
int arr[10];

int main(){

    // Initializing with user-input.
    for(int i=0; i<10; i++){
        scanf("%d", &arr[i]);
    }

}
```

Note the ‘&’ in front of element

- You can initialize array with user-input by using `scanf()`

Tips for using array & scanf (1)

Declare array length to be big enough!

- If problem says there will be at most 1000 number of inputs:

```
int arr[1000];
```

or

```
char arr[1005]; // +5 just in case
```

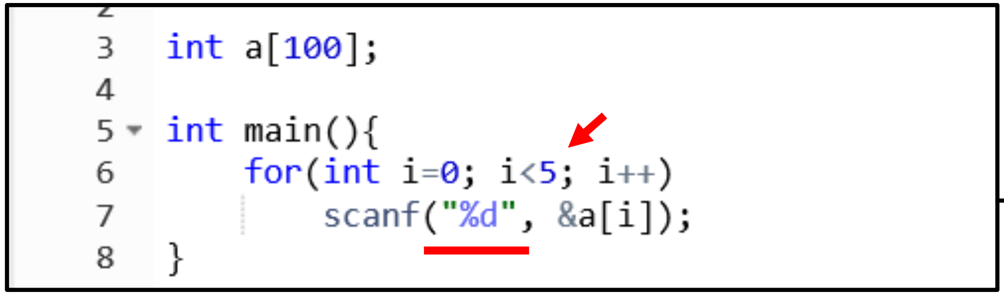
etc.

Tips for using array & scanf (2)

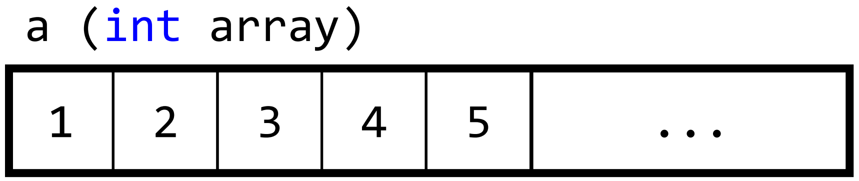
There are differences between int and char

[input]

```
1
2 3
4 5
```

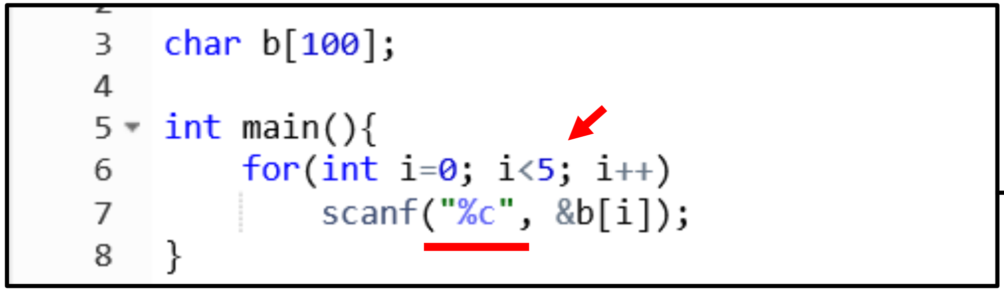


```
int a[100];

int main(){
    for(int i=0; i<5; i++)
        scanf("%d", &a[i]);
}
```

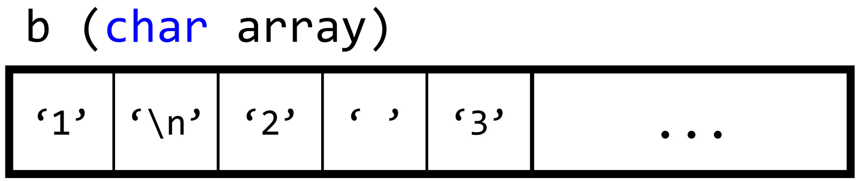


a (int array)

1	2	3	4	5	...



```
char b[100];

int main(){
    for(int i=0; i<5; i++)
        scanf("%c", &b[i]);
}
```



b (char array)

‘1’	‘\n’	‘2’	‘ ’	‘3’	...

Tips for using array & scanf (2)

In order to ignore an unwanted character (spaces/newlines/etc.) while getting input:

- Save to a temporary `char` and ignore it yourself.

```
char tmp;
scanf(“%c”, &tmp);   // scans a single character, and saves to tmp.
```

- Using “`%*c`” ignores a single character of the input for you.

```
scanf(“%*c”);        // scans a single character, but does not save it.
```

The C library: Math.h

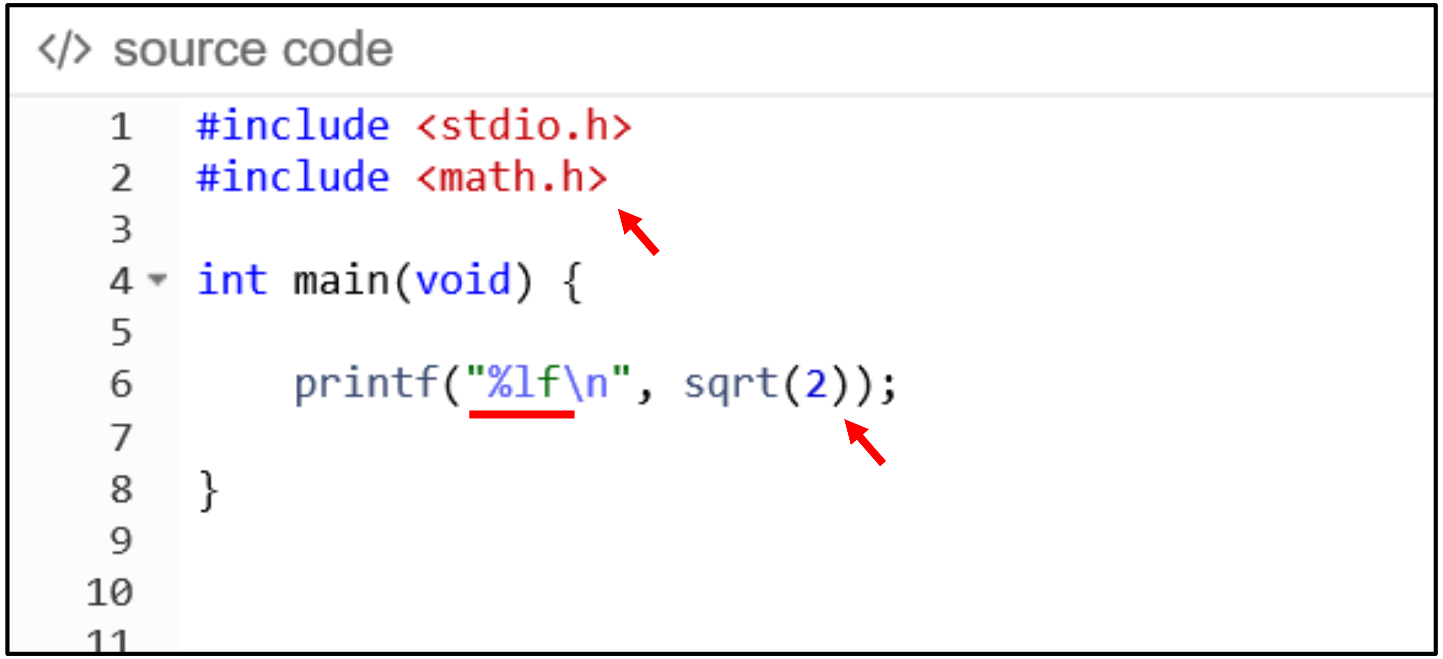
```
#include <math.h>
```

- Functions:

```
double pow(double x, double y)        // returns x^y
double sqrt(double x)                 // returns x^(1/2)
double ceil(double x)                 // round up to nearest integer value
double floor(double x)                // round down to nearest integer value
```

etc.

The C library: Math.h



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

int main(void) {

    printf("%lf\n", sqrt(2));

}
```

[output]

```
1.414214
```

<Compile>

```
gcc filename.c -lm
gcc filename.c -o filename -lm
```

To use Math.h, you MUST include the –lm option when compiling on your terminal!

Homework Problems

1. Print In Reverse
2. Count Alphabet
3. Maximum Sum Subarray
4. Closest Distance Points

Problem. **1**

Print In Reverse

Description

Write a program that gets a number of integers as input, and prints them in the reverse order they were given.

The first line of the input will state the number of integers that will be given : N $(1 \leq N \leq 10^5)$.

The second line contains the N integers, all of which are within the range of `int`.

Print the N integers in the reverse order they were given, with single spaces in between.

Input

The first line contains a single integer N $(1 \leq N \leq 10^5)$.

The second line contains N integers.

Output

Print the N integers in reverse order they were given.

Sample

```
[input]
5
10 50 20 40 30
```

```
[output]
30 40 20 50 10
```

Problem. **2**

Count Alphabet

Description

Write a program that counts how many of each alphabet occurs in the input. The given input will only consist of alphabet characters. (NO spaces or any other characters.)

When counting, consider lower/upper cases to be the same alphabet. (i.e. Count 'a' and 'A' as the same alphabet.)

Print each alphabet and its count, line by line, in alphabetical order. Only print the alphabets that occur at least once.

Input

First line contains a single integer N $(1 \leq N \leq 10^5)$.

Second line contains N alphabet characters.

Output

Print line by line, each alphabet and its count. (As lower case; Single space between alphabet and count; Alphabetical order.)

Sample

```
[input]
7
cIsCool


[output]
c 2
i 1
l 1
o 2
s 1
```

Problem. **3**

Maximum Sum Subarray

Description

Given a number of integers as input, find a subarray whose sum of its elements is the maximum. (See next slide for what is and is not a ‘subarray’.)

N $(1 \leq N \leq 100)$ integers will be given.
Each integer will be in the range of [-100000, 100000].

Print the sum of the subarray with the maximum sum.

Input

First line contains a single integer N $(1 \leq N \leq 100)$.

Second line contains N integers, each in range [-100000, 100000].

Output

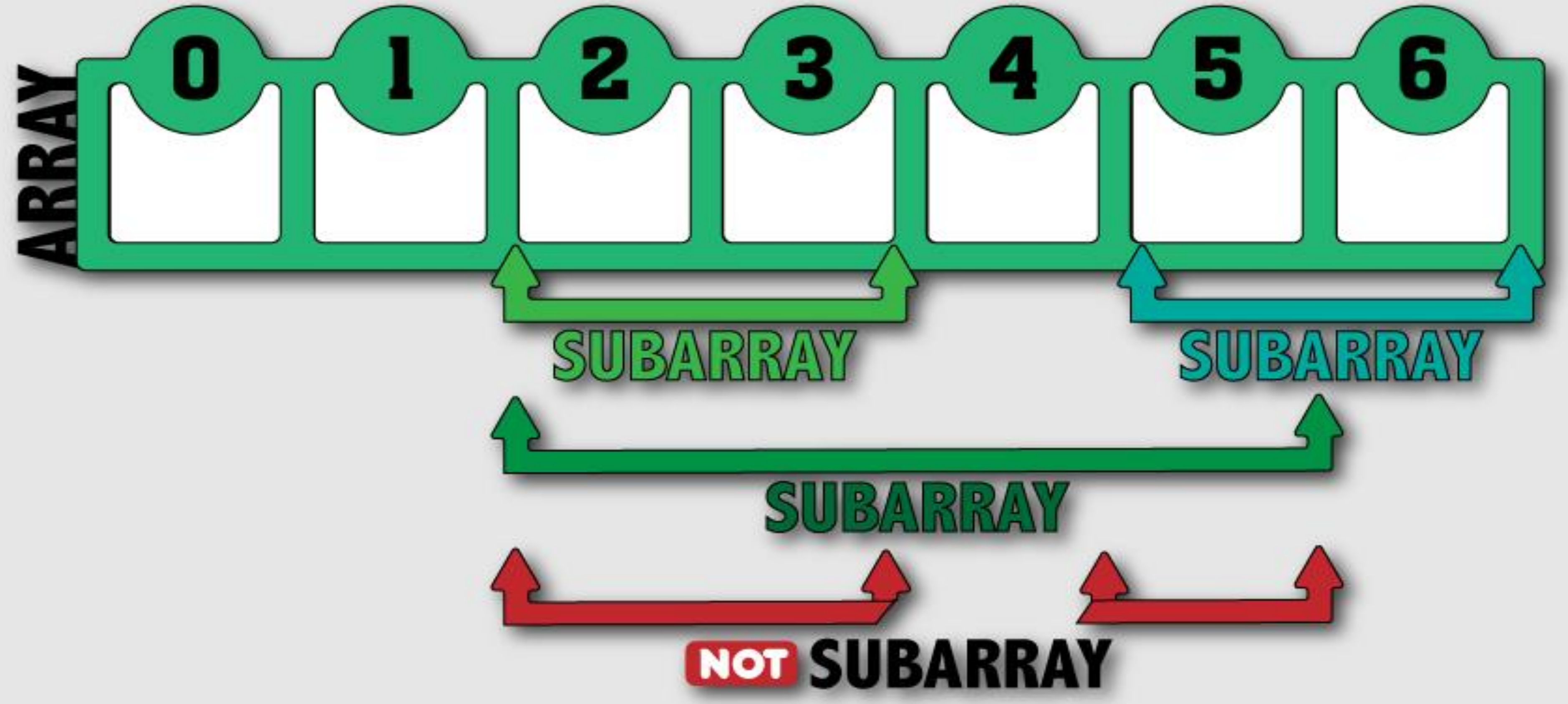
Print the sum of the maximum sum subarray.

Sample

```
[input]          [input]
4                5
2 -1 2 -1        0 0 0 -1 -1


[output]         [output]
3                0
```

Problem. **3**

Maximum Sum Subarray (cont.)



* Length of 0 is not considered a subarray

Problem. **3**

Maximum Sum Subarray (cont.)

<array>

1	-2	3	-4

<subarray>	<sum>	<current max>
1	= 1	1
1 -2	= -1	1
1 -2 3	= 2	2
1 -2 3 -4	= -2	2
-2	= -2	2
-2 3	= 1	2
-2 3 -4	= -3	2
3	= 3	3
3 -4	= -1	3
-4	= -4	3

Problem. **4**

Closest Distance Points

Description

Given a number of points on a coordinate plane(좌표 평면), find a pair of points with the closest distance. (See next slides for details.)

N $(2 \leq N \leq 1000)$ points will be given. Each point is given by specifying the x and y coordinates. The x, y coordinate values of all points are integers of range [-10000, 10000].

Print the distance between the closest two points.

Input

First line contains a single integer N $(2 \leq N \leq 1000)$.

Following N lines contain two integers x, y of range [-10000, 10000].

Output

Print the distance of closest two points. (Error is okay up to 10^-6.)

Sample

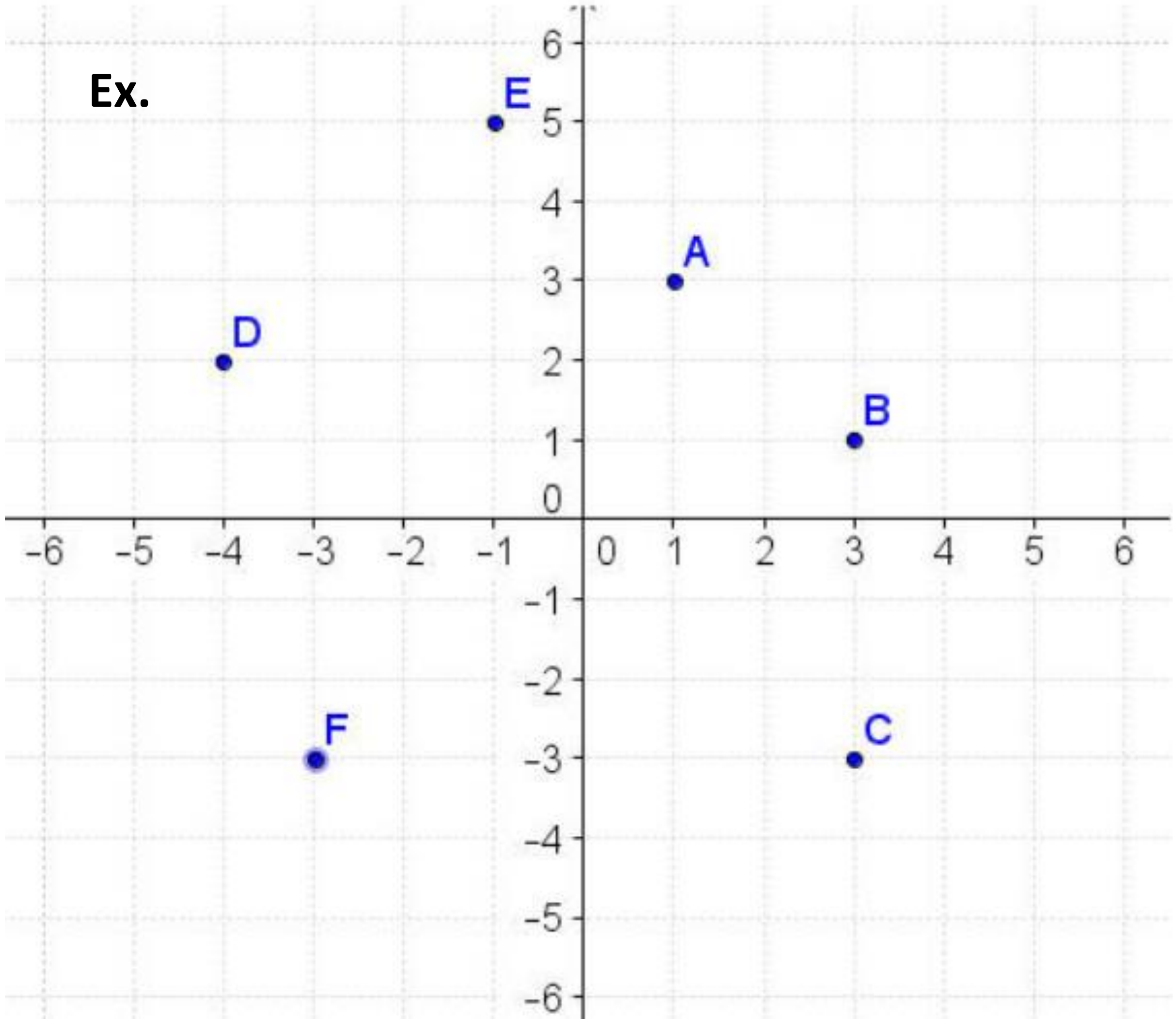
```
[input]
2
0 0
1 1

[output]
1.414214
```

```
[input]
3
0 0
0 1
1 1

[output]
1.000000
```

Problem. **4**

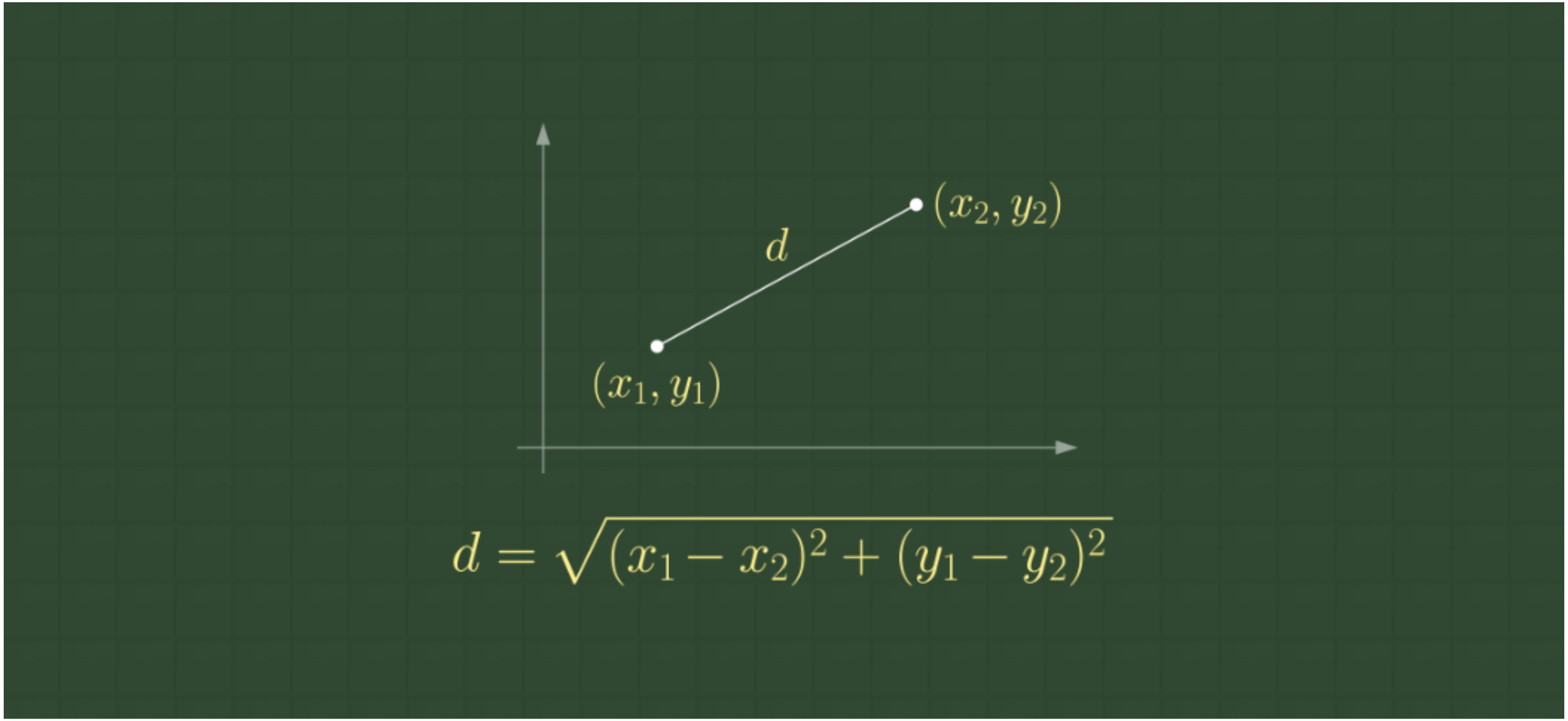
Closest Distance Points (cont.)

Ex.



Problem. **4**

Closest Distance Points (cont.)



$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$