

2023/03/20

# APPDONG

## C언어 멘토링

1주차  
멘토 : 김민수



# Contents

- 01. 멘토링 목표
- 02. 멘토링 진행 방향
- 03. 문제 풀이 사이트 소개
- 04. C언어 시작하기
- 05. 변수
- 06. 연산자
- 07. 입력
- 08. 문제 해결

# ■ 멘토링 목표

1. C언어 문법 공부
2. 기본적인 문제해결
3. 응용 문제 도전

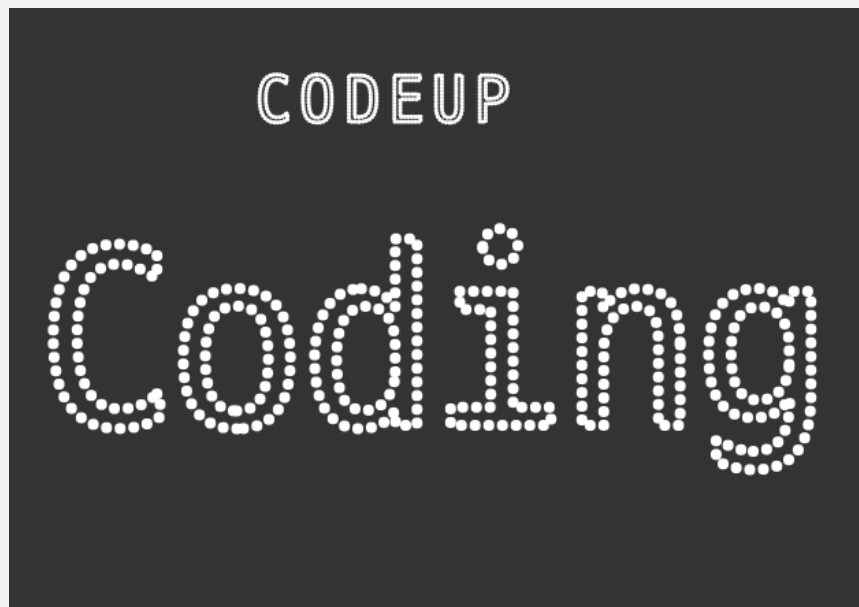
# **멘토링 진행 방향**

- 1. 멘토링은 총 5주차로 진행할 예정 (+1주차 될 수 있음)**
- 2. 월요일 오후 3시 30분에 시작 (약 90분 진행)**
- 3. 수업 형태의 이론시간과 같이 문제를 해결하는 실습 병행**

# ■ 멘토링 진행 방향

- 4. 각 주차마다 과제 있음 (필수 아님. 해오면 상품 지급)
- 5. 멘토링 자료는 카톡으로 미리 공유 (깃허브에도 따로 올릴 예정)
- 6. 멘토링 시간 외에도 공부하다가 모르는 부분은 질문

# 문제 풀이 사이트 소개



1. 코드업 (<https://codeup.kr/index.php>)
  - C언어 기초 100제

# 문제 풀이 사이트 소개

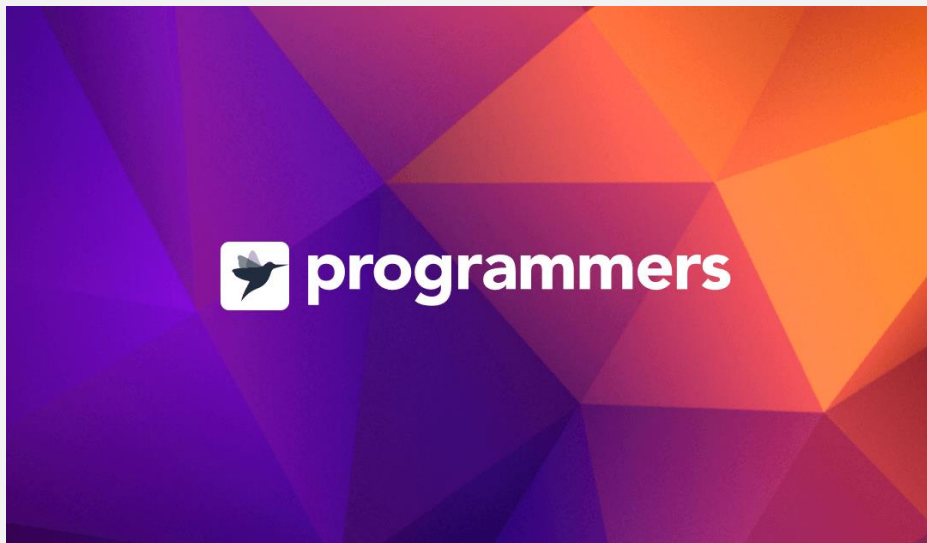
The logo for BAEKJOON ONLINE JUDGE. The text "BAEKJOON" is in a large, sans-serif font, with the "K" and "J" in blue and the "O"s in grey. Below it, "ONLINE JUDGE" is written in a smaller, all-caps, sans-serif font.

BAEKJOON  
ONLINE JUDGE

2. 백준 (<https://www.acmicpc.net/>)

- 단계별로 풀어보기

# 문제 풀이 사이트 소개



3. 프로그래머스 (<https://programmers.co.kr/>)

- 코딩테스트 입문



# c언어 시작하기

## 1. Hello, World! 프로그램 분석

```
1  #include <stdio.h>
2
3  int main() {
4      printf("Hello, World!\n");
5
6      return 0;
7  }
```

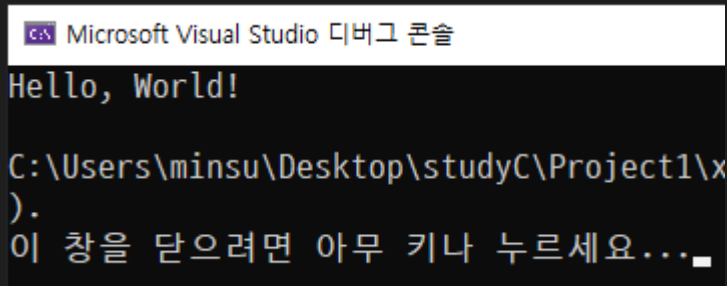
`stdio.h` -> Standard Input Output Header

`int main()` -> 정수형을 반환하는 메인 함수

`printf();` -> 괄호 안의 내용을 출력

`\n` -> 개행 문자

`return 0;` -> 0을 반환 (프로그램 종료)



Microsoft Visual Studio 디버그 콘솔

```
Hello, World!

C:\Users\minsu\Desktop\studyC\Project1\...
).
```

이 창을 닫으려면 아무 키나 누르세요...

# c언어 시작하기

## 2. 주석(Comment)

```
1  #include <stdio.h>
2
3  int main() {
4      // 한 줄에 쓰는 주석
5
6      /*
7       * 여러 줄에
8       * 걸쳐서 쓰는
9       * 주석
10     */
11
12     return 0;
13 }
```

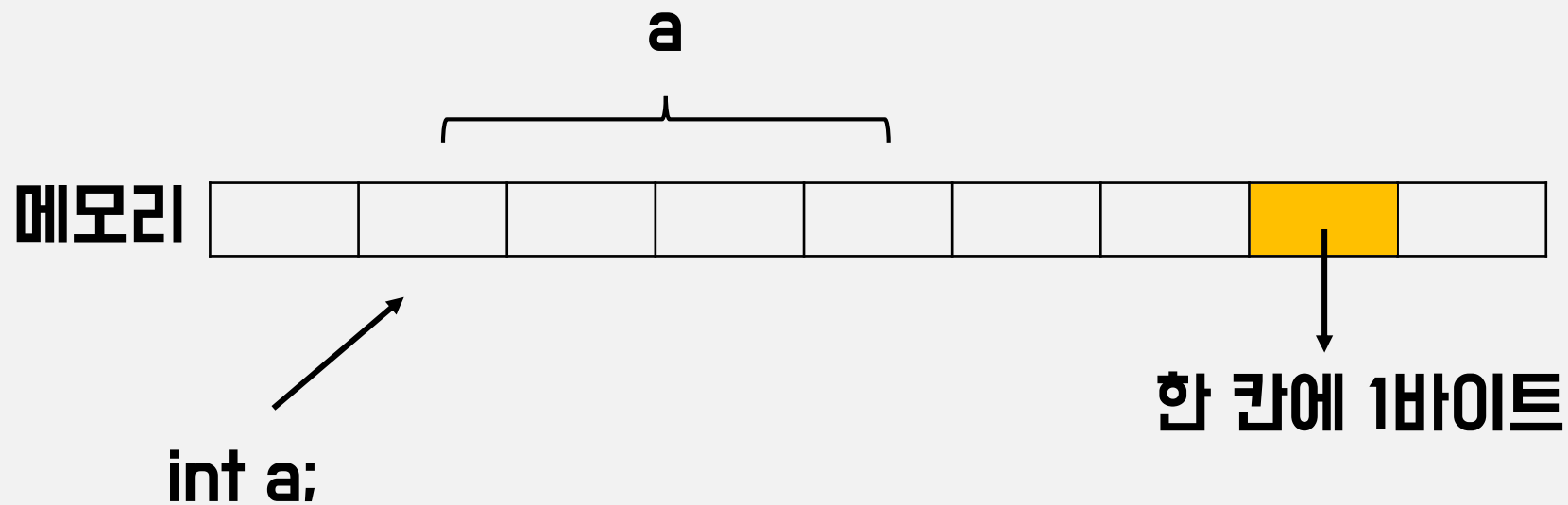
주석으로 표시된 부분은 컴파일러가 무시함

코드를 부가 설명하는 코멘트로 많이 사용함

주석을 사용하는 습관이 중요함

# 변수

C언어에서 말하는 변수(Variable) = "바뀔 수 있는 어떤 값을 보관하는 곳"



# 변수

## 1. 변수 선언

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5      a = 10;
6
7      printf("a의 값은 : %d\n", a);
8
9      return 0;
10 }
```

`int a;` -> 정수 형 변수 `a`를 선언

`a = 10;` -> `a`에 값 10을 저장

`%d` -> 다음에 오는 첫번째 변수를 10진수로 출력

Microsoft Visual Studio 디버그 콘솔

a의 값은 : 10

C:\Users\minsu\Desktop\studyC\Project1\x  
개).  
이 창을 닫으려면 아무 키나 누르세요...\_

# 변수

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5      a = 155;
6
7      printf("a의 값은 %d진수로 %d이다.\n", 10, a);
8      printf("a의 값은 %d진수로 %o이다.\n", 8, a);
9      printf("a의 값은 %d진수로 %x이다.\n", 16, a);
10
11     return 0;
12 }
```

10진수 : %d

8진수 : %o

16진수 : %x

Microsoft Visual Studio 디버그 콘솔

```
a의 값은 10진수로 155이다.
a의 값은 8진수로 233이다.
a의 값은 16진수로 9b이다.
```

```
C:\Users\minsu\Desktop\studyC\Project1\
개).
이 창을 닫으려면 아무 키나 누르세요..._
```

# 변수

## 2. 실수형 변수

실수형 변수에는 **float**과 **double** 두 종류가 있음

**float** : 4바이트 / 소수 6자리까지 정확하게 표현  
/ 값을 저장할 때 맨 끝에 f를 붙여야 함. ex) 3.14f

**double** : 8바이트 크기 / 소수 15자리까지 정확하게 표현

# 변수

## 2. 실수형 변수

```
1  #include <stdio.h>
2
3  int main() {
4      float a = 3.141592f;
5      double b = 3.141592;
6
7      printf("a : %f\n", a);
8      printf("b : %f\n", b);
9
10     return 0;
11 }
```

float 타입은 값을 저장할 때 마지막에 f를 작성  
(작성하지 않으면 float으로 인식하지 않아서  
컴파일 에러가 발생)

%f -> 실수형 변수를 출력

C:\> Microsoft Visual Studio 디버그 콘솔

```
a : 3.141592
b : 3.141592
```

```
C:\Users\minsu\Desktop\studyC\Project1\
).
```

# 변수

## 3. printf 형식

```
1 #include <stdio.h>
2
3 int main() {
4     float a = 1.234567f;
5     double b = 1.234567;
6     int c = 123;
7
8     printf("a : %.2f\n", a);
9     printf("b : %6.3f\n", b);
10    printf("c : %5d\n", c);
11
12    return 0;
13 }
```

Microsoft Visual Studio 디버그 콘솔

```
a : 1.23
b : 1.235
c : 123
```

```
C:\Users\minsu\Desktop\studyC\Project1\
개).
이 창을 닫으려면 아무 키나 누르세요...
```

**%.2f -> 소수점 둘째 자리까지만 표시**

**%6.3f -> 소수점 셋째 자리까지만 표시 & 6자리**

**%5d -> 5자리**



# 변수

## 3. printf 형식

**%.2f : 소수점 아래 둘째 자리까지만 표시 (셋째 자리에서 반올림)**

a 

1	.	2	3
---	---	---	---

**%.6.3f : 소수점 아래 셋째 자리까지만 표시 (넷째 자리에서 반올림) & 6자리**

b 

	1	.	2	3	5
--	---	---	---	---	---

**%5d : 5자리**

c 

		1	2	3
--	--	---	---	---

# 변수

## 4. 여러가지 자료형

자료형 / 명칭		크기	값의 표현범위
정수형	char	1바이트	-128이상 +127이하
	short	2바이트	-32,768이상 +32,767이하
	int	4바이트	-2,147,483,648이상 +2,147,483,647이하
	long	4바이트	-2,147,483,648이상 +2,147,483,647이하
	long long	8바이트	-9,223,372,036,854,775,808이상 +9,223,372,036,854,775,807이하
실수형	float	4바이트	$\pm 3.4 \times 10^{-37}$ 이상 $\pm 3.4 \times 10^{38}$ 이하
	double	8바이트	$\pm 1.7 \times 10^{-307}$ 이상 $\pm 3.4 \times 10^{308}$ 이하
	long double	8바이트 이상	double 이상의 표현범위

구분	명칭	설명
부호가 있는 변수	signed	기본(default) 형식
부호가 없는 변수	unsigned	음수를 표현할 수 없고, 양수 값의 표현범위가 두배 정도 늘어남

# ■ 연산자

## 1. 산술 연산자(Arithmetic Operator)

더하기(+) 빼기(-) 곱하기(\*) 나누기(/) 나머지(%) → + - \* / %

## 2. 대입 연산자(Assignment Operator)

대입(=) → a = 5; 우측의 값을 좌측에 대입

+= -= \*= ... 등등 복합 대입 연산자도 있음

# 연산자

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b;
5     a = 10;
6     b = 3;
7
8     printf("a + b = %d\n", a + b);
9     printf("a - b = %d\n", a - b);
10    printf("a * b = %d\n", a * b);
11    printf("a / b = %d\n", a / b);
12    printf("a %% b = %d\n", a % b);
13
14    return 0;
15 }
```

Microsoft Visual Studio 디버그 콘솔

```
a + b = 13
a - b = 7
a * b = 30
a / b = 3
a % b = 1
```

```
C:\Users\minsu\Desktop\studyC\Project1\x
개).
이 창을 닫으려면 아무 키나 누르세요...
```

**a / b의 결과가 3.3333이 아니라 3인 이유  
: 변수 a, b 모두 int형으로 선언했기 때문**

**%는 출력할 때 사용하려면 %%로 작성  
-> %d %f 등으로 이용되기 때문에  
%하나로는 표기되지 않음**

# 연산자

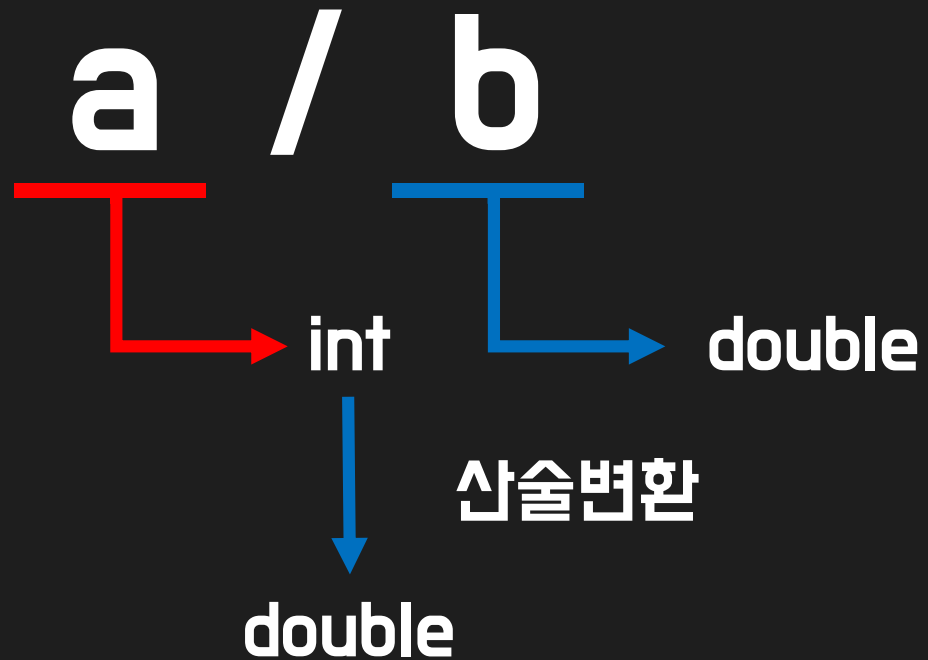
```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5      double b;
6      a = 10, b = 3;
7
8      printf("a / b = %f\n", a / b);
9
10     return 0;
11 }
```

선택 Microsoft Visual Studio 디버그 콘솔

a / b = 3.333333

C:\Users\minsu\Desktop\studyC\Project1\src\main.c (개).

이 창을 닫으려면 아무 키나 누르세요...\_



# 연산자

## 3. 증감 연산자

연산자가 피연산자 앞에 오는 경우 전위형(prefix)

ex) ++a --a

연산자가 피연산자 뒤에 오는 경우 후위형(postfix)

ex) a++ a--

# 연산자

```
1  #include <stdio.h>
2
3  int main() {
4      int a;
5
6      a = 1;
7      printf("a += 1 결과 : %d\n", a += 1);
8
9      a = 1;
10     printf("++a 결과 : %d\n", ++a);
11
12     a = 1;
13     printf("a++ 결과 : %d\n", a++);
14
15     return 0;
16 }
```

Microsoft Visual Studio 디버그 콘솔

```
a += 1 결과 : 2
++a 결과 : 2
a++ 결과 : 1

C:\Users\minsu\Desktop\studyC\Project1\
개).
이 창을 닫으려면 아무 키나 누르세요...
```

$a += 1 \rightarrow a = a + 1$

전위형의 경우 연산 후의 결과값을 반환  
→ 증감 연산이 바로 수행된다.

후위형의 경우 값을 반환 후 연산을 수행

# 연산자

## 4. 비트 연산자

비트 하나 하나에 대한 연산

- AND(&) 연산
- OR(!) 연산
- XOR(^) 연산
- 쉬프트(<< 또는 >>) 연산
- 반전(~) 연산

등등..



# 연산자

## 4. 비트 연산자

int a = 10; 비트열 00000000 00000000 00000000 00001010  
int b = 12; 비트열 00000000 00000000 00000000 00001100

00001010  
00001100



00001000

&(논리곱)

00001110

!(논리합)

00000110

^(배타적 논리합)

# 연산자

## 4. 비트 연산자

int a = 10; 비트열 00000000 00000000 00000000 00001010

~a; 비트열 11111111 11111111 11111111 11110101



반전(~) : 피연산자의 비트를 반전

# 연산자

## 4. 비트 연산자

00001010  
00010100

사라짐

int a = 10; 비트열 00000000 00000000 00000000 00001010

a << 1; 비트열 00000000 00000000 00000000 00010100

0으로 채움

int a = 10; 비트열 00000000 00000000 00000000 00001010

a >> 2; 비트열 00000000 00000000 00000000 00000010

0으로 채움

사라짐

# 연산자

```
1  #include <stdio.h>
2
3  int main() {
4      int a = 0xAF;    // 175 (10101111)
5      int b = 0xB5;    // 181 (10110101)
6
7      printf("a & b = %x\n", a & b);
8      printf("a | b = %x\n", a | b);
9      printf("a ^ b = %x\n", a ^ b);
10     printf("~a = %x\n", ~a);
11     printf("a << 2 = %x\n", a << 2);
12     printf("b >> 3 = %x\n", b >> 3);
13
14     return 0;
15 }
```

Microsoft Visual Studio 디버그 콘솔

```
a & b = a5
a | b = bf
a ^ b = 1a
~a = fffffff50
a << 2 = 2bc
b >> 3 = 16
```

C:\Users\minsu\Desktop\studyC\Project1\x  
개).  
이 창을 닫으려면 아무 키나 누르세요...\_

왜 이런 결과가 나오는 지  
직접 비트 연산 해보기

# 입력

**scanf():** 를 사용해서 정수, 실수, 문자 등등 입력 가능

**int 형 :** scanf(“%d”, &a);

**float 형 :** scanf(“%f”, &a);

**double 형 :** scanf(“%lf”, &a);

**char 형 :** scanf(“%c”, &a);

**주의 해야 할 점**

**printf():** 에서는 float double 모두 %f로 받았지만

**scanf():** 에서는 %f %lf로 구분해야 한다

# 입력

```
1 #include <stdio.h>
2
3 int main() {
4     double celsius; // 섭씨 온도
5
6     printf("섭씨 온도를 화씨 온도로 변환\n");
7     printf("섭씨 온도를 입력 >> ");
8
9     scanf_s("%lf", &celsius);
10
11     printf("섭씨 %f도는 화씨로 %f도이다.\n", celsius, 9 * celsius / 5 + 32);
12
13     return 0;
14 }
```

Microsoft Visual Studio 디버그 콘솔

```
섭씨 온도를 화씨 온도로 변환
섭씨 온도를 입력 >> 36.5
섭씨 36.500000도는 화씨로 97.700000도이다.
```

```
C:\Users\minsu\Desktop\studyC\Project1\x64\
개).
이 창을 닫으려면 아무 키나 누르세요...
```

scanf\_s() ?

-> scanf()는 입력받는 데이터의 크기를 확인하지 않음  
그냥 사용하면 컴파일 에러가 발생함

# 입력

```
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  int main() {
6      double celsius; // 섭씨 온도
7
8      printf("섭씨 온도를 화씨 온도로 변환\n");
9      printf("섭씨 온도를 입력 >> ");
10
11     scanf("%lf", &celsius);
12
13     printf("섭씨 %f도는 화씨로 %f도이다.\n", celsius, 9 * celsius / 5 + 32);
14
15     return 0;
16 }
```

`_CRT_SECURE_NO_WARNINGS` 를 추가하면  
에러를 무시할 수 있음

# 입력

## 아스키코드(ASCII code)

문자를 숫자에 대응시켜 놓은 표준

숫자를 이용하여 문자를 처리하거나  
문자를 이용하여 숫자를 처리할 수 있음

ex) `printf("%c", 65);` -> A  
`printf("%d", 'A');` -> 65

제어 문자		공백 문자		구두점		숫자		알파벳			
10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60	
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0x08	BS	40	0x28	(	72	0x48	H	104	0x68	h
9	0x09	HT	41	0x29	)	73	0x49	I	105	0x69	i
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z
27	0x1B	ESC	59	0x3B	;	91	0x5B	[	123	0x7B	{
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C	
29	0x1D	GS	61	0x3D	=	93	0x5D	]	125	0x7D	}
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL



# 문제 해결

## 백준 단계별로 풀어보기 1단계

### "입출력과 사칙연산" 1번~10번

전체	출처	단계	분류	추가된 문제	더 보기 ▾	검색	Q
단계별은 @jh05013님이 관리하고 계십니다.							
단계	제목	설명	정보	총 문제	내가 맞은 문제		
1	입출력과 사칙연산	입력, 출력과 사칙연산을 연습해 봅시다. Hello World!		13			
2	조건문	if 등의 조건문을 사용해 봅시다.		7			
3	반복문	for, while 등의 반복문을 사용해 봅시다.		12			
4	1차원 배열	배열을 사용해 봅시다.		10			
5	문자열	문자열을 다루는 문제들을 해결해 봅시다.		11			
6	심화 1	지금까지의 프로그래밍 문법으로 더 어려운 문제들을 풀어봅시다.		10			
7	2차원 배열	배열 안에 배열이 있다면 어떨까요? 2차원 배열을 만들어 봅시다.		4			

