

Euidong Lee
Junseok Lee
Minhyo Jeong
(snucsl.ta@gmail.com)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2022

4190.103A-001: Programming Practice Lab. 4



실습 관련 공지

실습조

- 실습조를 나누었으니 자신에게 해당하는 조교님들에게 문의 부탁드립니다.
- @shawn950303
 - 1조: 성씨가 ‘ㄱ’ 부터 ‘ㅈ’ 사이에 해당하는 학생
- @euidong
 - 2조: 성씨가 ‘ㅇ’ 부터 ‘ㅎ’ 사이에 해당하는 학생
- replit에 notification이 많이 발생해 놓칠 수도 있습니다. 필요한 경우 Zoom chat 창에 전체 채팅으로 작성해주셔도 됩니다.

설문조사

- <https://forms.gle/pW5w9G5csbWEfEep9>
- 필수는 아니지만 의견 개진은 환영합니다.

프로그래밍 연습 TA-session

프로그래밍 연습 실습 진행을 위한 설문입니다.

학번을 입력해주세요.

단답형 텍스트

이번 수업을 통해 프로그래밍을 처음 접하셨나요? *

- ☐ 네
- ☐ 아니오

실습 난이도가 어떠신가요? (1-매우 쉽다 2-쉽다 3-보통이다 4-어렵다 5-매우 어렵다) *

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

실습 진행 속도가 어떠신가요? (1-매우 느리다 2-느리다 3-보통이다 4-빠르다 5-매우 빠르다) *

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

과제 난이도가 어떠신가요? (1-매우 쉽다 2-쉽다 3-보통이다 4-어렵다 5-매우 어렵다) *

- ☐ 1
- ☐ 2
- ☐ 3

실습자료

- <http://csl.snu.ac.kr/courses/4190.103/2022-1/>
- 실습 시작 후에 홈페이지에서 확인 바랍니다.



서울대학교
SYSTEMS SOFTWARE &
ARCHITECTURE LABORATORY



HomePeopleCoursesResearchPublications

Schedule

The following schedule is tentative and subject to change without notice.
(L = Lecture @ 302-208, P = Practice session @ 302-311-1)

Day	Topic	Reading
3/2	L Course Overview (Online)	Chap. 0
	P Practice Session 1 (Online)	
3/7	L An Overview of C (Online)	Chap. 1
3/9	Presidential Election Day (no class)	
3/10	P Practice Session 2 (make-up) (Online)	
3/14	L Lexical Elements and Operators (Online)	Chap. 2
3/16	P Practice Session 3 (Online)	
3/21	L Fundamental Data Types (Online)	Chap. 3
3/23	P Practice Session 4 (Online)	

Lab. 3 실습 문제 풀이

Lab.3) 실습 4

- Up & Down 게임을 만들어 보세요
 - 컴퓨터가 $0 \leq X < 32$ 사이의 임의의 숫자를 하나 선정 (Random 함수사용)
 - 사용자에게 5번의 기회를 주고 각 기회마다 하나의 숫자를 입력
 - 입력받은 숫자와 답을 비교해서 Up, Down 중 하나를 출력
 - 답이 맞으면 Success를 출력하고 종료하고 정해진 횟수에 끝나지 않았을 경우 Fail을 출력

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

int main(void) {
    srand(time(NULL));
    int random_value = rand();
```

```
Console  Shell
> make -s
> ./main
1: 15
Down
2: 8
Down
3: 4
Down
4: 2
Down
5: 1
Success> |
```

```
Console  Shell
> make -s
> ./main
1: 1
Up
2: 2
Up
3: 3
Up
4: 4
Up
5: 5
Up
Fail. The answer is 22
> |
```

Lab.3) 실습 4 - 설명

- Up & Down 게임을 만들어 보세요

```
int main(void) {
    int input, random_value, success = 0;
    srand(time(NULL));
    random_value = rand() % 32;

    for (int i = 1; i <= 5; i++) { // ANSI C 지원 기능 아님
        printf("%d: ", i);
        scanf("%d", &input);

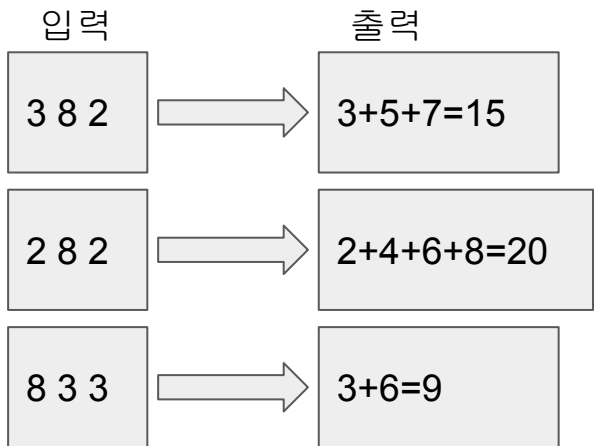
        if (input == random_value) {
            success = 1;
            break;
        }
        else if (input < random_value)
            printf("Up\n");
        else
            printf("Down\n");
    }
    if (success == 1)
        printf("Success");
    else
        printf("Fail. The answer is %d\n", random_value);

    return 0;
}
```

Lab. 3 과제 풀이

Lab.3) 과제 1

- 양의 정수를 세개(n, m, k) 입력받고 n 에서 m , 혹은 m 에서 n 으로 k 만큼 증가시켜가며 더하는 프로그램
 - 양의 정수 세개(n, m, k)를 입력
 - n, m 의 대소관계를 파악하여 작은수에서 큰수까지 K 씩 증가시켜 더함



```
❖ make -s
❖ ./main
3 8 2
3+5+7=15
❖ ./main
2 8 2
2+4+6+8=20
❖ ./main
8 3 3
3+6=9
❖
```

Lab.3) 과제 1 - 풀이

```
#include <stdio.h>
int main(void) {
    int n,m,k, temp, sum = 0;

    scanf("%d %d %d", &n, &m, &k);

    if (n > m) {
        temp = n;
        n = m;
        m = temp;
    }

    printf("%d", n);

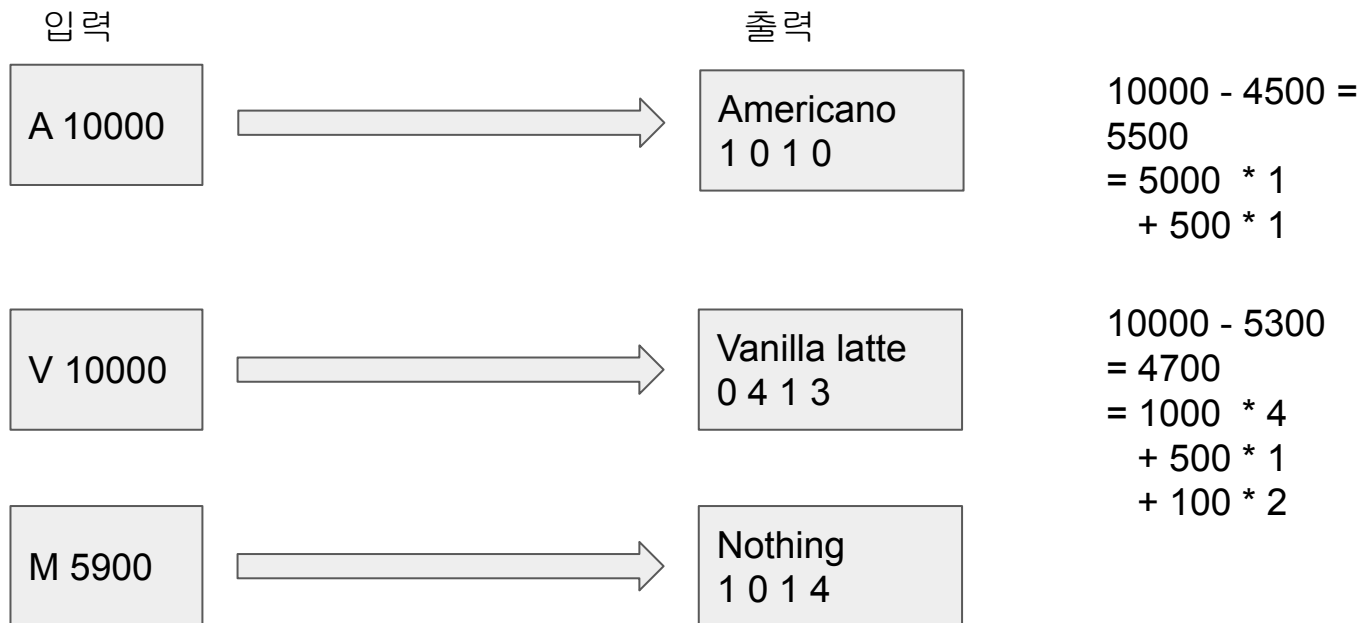
    for (sum = n, n += k; n <= m ; n = n + k) {
        sum = sum + n;
        printf("+%d", n);
    }
    printf("=%d\n", sum);
    return 0;
}
```

Lab.3) 과제 2

- 자판기에서 음료의 이름과 잔돈의 개수를 출력하는 프로그램
 - 음료를 나타내는 문자 N과 투입한 금액을 나타내는 정수 M(100의 배수)를 입력
 - 음료
 - 'A' = Americano(4500원), 'L' = Caffè latte(4800원),
 - 'M' = Caffè mocha(6000원), 'V' = Vanilla latte(5300원)
 - 잔돈
 - 5000원, 1000원, 500원, 100원
 - 단위가 높은 돈을 우선적으로 지급
 - 주문한 음료와 잔돈의 개수를 출력
 - 잔돈의 개수는 큰 단위부터 공백을 두고 출력
 - 투입한 금액이 선택한 음료의 가격보다 낮다면 'Nothing'을 출력하고 투입한 금액 전부를 반환

Lab.3) 과제 2

- 출력 예시



Lab.3) 과제 2 - 풀이

```
int main(void) {
    int price, input, change;
    int coin_5000, coin_1000, coin_500, coin_100;
    int success;
    char beverage;

    scanf("%c %d", &beverage, &input);

    if (beverage == 'A') {
        price = 4500;
    } else if (beverage == 'L') {
        price = 4800;
    } else if (beverage == 'M') {
        price = 6000;
    } else if (beverage == 'V') {
        price = 5300;
    } else {
        printf("Invalid beverage\n");
        return 0;
    }

    if (input < price) {
        change = input;
        success = 0;
    } else {
        change = input - price;
        success = 1;
    }

    coin_5000 = change / 5000;
    change %= 5000;
    coin_1000 = change / 1000;
    change %= 1000;
    coin_500 = change / 500;
    change %= 500;
    coin_100 = change / 100;

    if (success == 0) {
        printf("Nothing\n");
    } else {
        if (beverage == 'A') {
            printf("Americano\n");
        } else if (beverage == 'L') {
            printf("Cafe Latte\n");
        } else if (beverage == 'M') {
            printf("Cafe mocha\n");
        } else if (beverage == 'V') {
            printf("Vanilla latte\n");
        }
    }

    printf("%d %d %d %d\n", coin_5000, coin_1000, coin_500, coin_100);
    return 0;
}
```

```
if (input < price) {
    change = input;
    success = 0;
} else {
    change = input - price;
    success = 1;
}

coin_5000 = change / 5000;
change %= 5000;
coin_1000 = change / 1000;
change %= 1000;
coin_500 = change / 500;
change %= 500;
coin_100 = change / 100;

if (success == 0) {
    printf("Nothing\n");
} else {
    if (beverage == 'A') {
        printf("Americano\n");
    } else if (beverage == 'L') {
        printf("Cafe Latte\n");
    } else if (beverage == 'M') {
        printf("Cafe mocha\n");
    } else if (beverage == 'V') {
        printf("Vanilla latte\n");
    }
}

printf("%d %d %d %d\n", coin_5000, coin_1000, coin_500, coin_100);
return 0;
}
```

Tips

Compile Error / Warning Message

- replit에서 발생하는 에러와 경고 메시지를 만났을 때 당황하지 말고
- 메시지를 찬찬히 잘 읽어보면 해결의 실마리가 있을 수 있습니다.
- 주로 문법 오류 때문에 발생하는데,
 - 키워드의 맞춤법이 틀리거나
 - 문장부호를 빠뜨리는 경우
 - 선언되지 않은 변수 사용 등

C Compiler

- **Syntax of the language**
 - Rules for putting words and punctuation to make correct statements
- **Compiler**
 - A program that checks on the legality of C code
 - If errors, compilers prints error messages and stops
 - If NO errors, compiler translates the C code into object code

Compile Error / Warning Message

- 예시
 - “file name”:line:column: message

```
Console Shell
❯ make -s
main.c:12:26: error: expected expression
    printf("a=<b ; %d\n",a=<b);
                          ^
1 error generated.
make: *** [<built-in>: main.o] Error 1
exit status 2
❯
```

```
Console Shell
❯ make -s
main.c:8:11: error: expression is not assignable
    if (b*b = a)
        ~~~~ ^
main.c:15:13: error: expression is not assignable
    if (a*a = b)
        ~~~~ ^
2 errors generated.
make: *** [<built-in>: main.o] Error 1
exit status 2
❯
```

Compile Error / Warning Message

- 예시

```
Console Shell
❖ make -s
main.c:10:1: error: expected expression
@shawn950303
^
1 error generated.
make: *** [<builtin>: main.o] Error 1
exit status 2
❖
```

```
Console Shell
❖ make -s
❖ ./main
3
signal: segmentation fault (core dumped)
❖
```

Compile Error / Warning Message

- 예시

```
10 | if (90 <= score <= 100)
11 |     printf("Grade: ");
12 | else if (80 <= score)
13 |     printf();
14 |
15 | return 0;
16 | }
17 |
```

result of comparison of constant 100 with boolean expression is always true

논리연산자 (logical operator)

논리연산자 &&, ||, !

- 논리연산자는 연산식의 조합의 참, 거짓 판단에 사용됩니다.
- 논리 연산자는 세 가지 종류가 있습니다.
 - 모두 참일 때만 참인 **and** 연산자: **&&**
 - 하나라도 참이면 참인 **or** 연산자: **||**
 - 참은 거짓으로, 거짓은 참으로 **not** 연산자: **!**



논리연산자 &&, ||, !

- and 연산자(&&)를 기준으로 왼쪽 연산식과 오른쪽 연산식으로 표현할 수 있습니다.
- $(6 > 3) \ \&\& \ (5 < 3)$ 의 연산식을 생각해봅시다.
- $(6 > 3)$ 은 참이고, $(5 < 3)$ 은 거짓 입니다. 즉, (참) && (거짓) 의 연산식으로,
- and 연산자의 피연산식이 모두 참이 아니므로 결과는 거짓, 0이 됩니다.

논리연산자 &&, ||, !

- or 연산자(||)의 경우에도 왼쪽/오른쪽 연산식으로 표현할 수 있습니다.
- $(6 > 3) || (5 < 3)$ 의 연산식을 생각해봅시다.
- $(6 > 3)$ 은 참이고, $(5 < 3)$ 은 거짓 입니다. 즉, (참) || (거짓) 의 연산식으로,
- or 연산자의 피연산식이 중 참이 있으므로 결과는 참, 1이 됩니다.

논리연산자 &&, ||, !

- not 연산자(!)의 경우, 연산 수식의 반대로 논리를 평가합니다.
- 연산 수식이 거짓이면 참, 참이면 거짓으로 판별합니다.
- $(5 < 3)$ 은 거짓이지만 not 연산자를 취하는 경우,
- $!(5 < 3)$ 으로 !(거짓)은 참이 됩니다.

실습 1

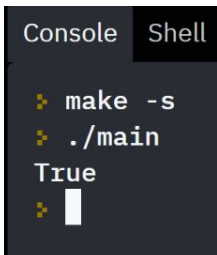
실습 1 - 설명

- 주어진 소스코드의 [빈 칸]에 논리연산자를 사용하여 true를 출력하시오 (1)

```
#include <stdio.h>

int main(void)
{
    if ( (6 > 3) [빈 칸] (5 < 3) )
        printf("True\n");

    return 0;
}
```



```
Console Shell
❯ make -s
❯ ./main
True
❯
```

실습 1 - 설명

- 주어진 소스코드의 [빈 칸]에 논리연산자를 사용하여 True를 출력하시오 (2)
- C 언어에서 논리값 거짓은 0, 참은 0이 아닌 값이며 보통 1을 사용합니다.

Console Shell

```
❯ make -s
❯ ./main
True
True
❯
```

```
#include <stdio.h>

int main(void)
{
    if ((0) [빈 칸1] (1))
        printf("False\n");
    else
        printf("True\n");

    if ([빈 칸2] (0))
        printf("True\n");
    else
        printf("False\n");

    return 0;
}
```

실습 1 - 설명

- 주어진 소스코드의 [빈 칸]에 논리연산자를 사용하여 True를 출력하시오 (3)

Console Shell

```
❯ make -s
❯ ./main
True
❯
```

```
#include <stdio.h>

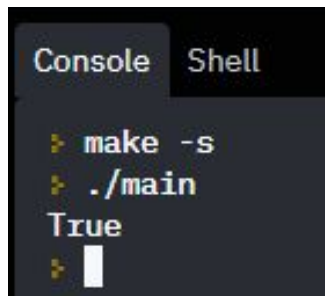
int main(void)
{
    int a, b;
    a = 10; b = 20;

    if ([빈 칸1](a > b) && [빈 칸2](b > a))
        printf("True\n");

    return 0;
}
```

실습 1 - 설명

- 주어진 소스코드의 [빈 칸]에 논리연산자를 사용하여 True를 출력하시오 (4)



```
root@kali:~/C/4190.103A-001# make -s
root@kali:~/C/4190.103A-001# ./main
True
root@kali:~/C/4190.103A-001#
```

```
#include <stdio.h>

int main(void)
{
    int a, b, c, d;
    a = 10; b = 20; c = 30; d = 40;

    if ([빈 칸1] (a > b) && [빈 칸2] (b > a) && \
        [빈 칸3] (c > a) && [빈 칸4] (a > d))
    {
        printf("True\n");
    }

    return 0;
}
```

실습 1 - 설명

- 20대를 가려내는 프로그램
 - 하나의 양의 정수를 입력 받음
 - 출력
 - 입력이 20 보다 크거나 같고, 30보다 작은 경우
 - In one's twenties
 - 이외의 경우
 - Not in one's twenties
 - 논리 연산자를 사용해 구현할 것

```
Console Shell
❖ make -s
❖ ./main
10
Not in one's twenties
❖ ./main
20
In one's twenties
❖ ./main
30
Not in one's twenties
❖ ./main
25
In one's twenties
❖
```

실습 2

실습 2 - 설명

- 연도를 나타내는 양의 정수 N 을 랜덤으로 생성해, 윤년인지 평년인지 출력하는 프로그램을 작성하시오. (생성 범위: $1 \leq N \leq 2050$)
 - 윤년에 대한 규칙은 다음과 같습니다.
 - ① 주어진 연도가 400으로 나누어 떨어지면 윤년이다.
 - ② 주어진 연도가 100으로 나누어 떨어지면 평년이다.
 - ③ ②의 조건을 만족하지 않는 경우, 4로 나누어 떨어지면 윤년이다.
 - ④ 이외에는 모두 평년이다.

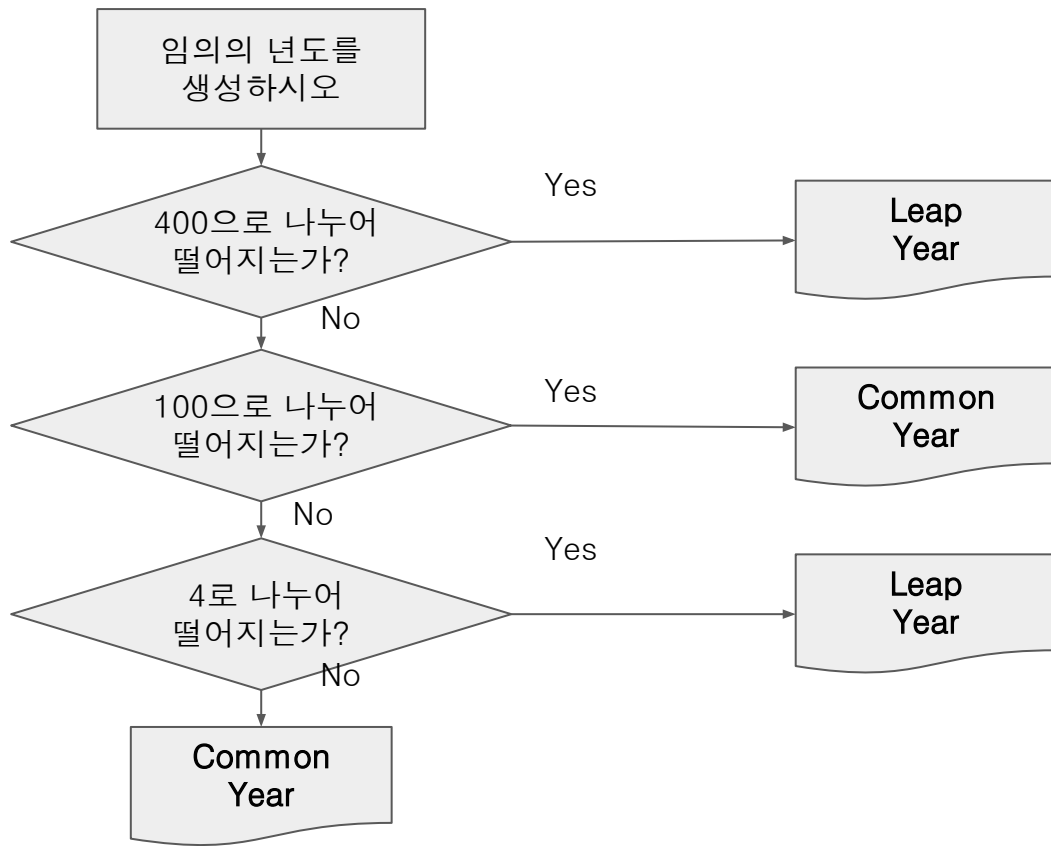
실습 2 - 설명

- 출력 예시 (랜덤 생성된 입력 => 출력)
 - 4 => [년도] Leap Year
 - 2016 => [년도] Leap Year
 - 1900 => [년도] Common Year
 - 여러분의 생년은 평년인가요 윤년인가요?

```
Console Shell
❖ make -s
❖ ./main
916 Leap Year.
❖ ./main
471 Common Year.
❖ ./main
1145 Common Year.
❖ ./main
91 Common Year.
❖ ./main
2007 Common Year.
❖ ./main
1258 Common Year.
❖ ./main
1664 Leap Year.
❖ ./main
1543 Common Year.
❖
```

실습 2 - 설명

- 코딩 전에 순서도를 그리면 도움이 됩니다.



실습 2 - 힌트

- if, else, else if / modular operation / rand() 함수 등을 사용해 보세요.
- rand() % N의 범위는 0~N-1 입니다.
 - 1~N의 범위로 결과를 얻고 싶으면 어떻게 하면 될까요?

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

int main() {
    int year;
    srand(time(NULL));
    year = rand() % ARBITRARY_NUMBER;
    // WHAT ELSE?
    ...
}
```

실습 2 - 제출

- 앞서 작성한 프로그램을 이용하여 1000년 부터 2000년까지, 100년 간격으로 윤년 여부를 판별하는 프로그램을 작성하시오.

```
Console Shell
❖ make -s
❖ ./main
1000 Common Year.
1100 Common Year.
1200 Leap Year.
1300 Common Year.
1400 Common Year.
1500 Common Year.
1600 Leap Year.
1700 Common Year.
1800 Common Year.
1900 Common Year.
2000 Leap Year.
❖
```

실습 3

실습 3 - 설명

- 문자로 연산하기
 - 문자는 ASCII 코드 규칙에 의해 정수로 저장되어 덧셈, 뺄셈 등을 할 수 있습니다.
 - 다음 코드를 편집기에 입력한 뒤 실행해보세요.

```
#include <stdio.h>

int main()
{
    printf("%c %d\n", 'a' + 1, 'a' + 1);
    printf("%c %d\n", 97 + 1, 97 + 1);

    return 0;
}
```

Console Shell

```
❏ make -s
❏ ./main
b 98
b 98
❏
```

실습 3 - 설명

- 문자로 연산하기
 - 'a' + 1과 같이 작은따옴표로 묶은 문자는 정수를 더하거나 뺄 수 있습니다.
 - 'a'의 ASCII 코드값은 97이고, 97에 1을 더하면 98입니다.
 - 따라서 98에 해당하는 문자 b가 출력됩니다.

dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char	dec	hex	oct	char
0	0	000	NULL	32	20	040	space	64	40	100	@	96	60	140	`
1	1	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	2	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	3	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	4	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	5	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	6	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	7	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	8	010	BS	40	28	050	(	72	48	110	H	104	68	150	h
9	9	011	TAB	41	29	051	)	73	49	111	I	105	69	151	i
10	a	012	LF	42	2a	052	*	74	4a	112	J	106	6a	152	j
11	b	013	VT	43	2b	053	+	75	4b	113	K	107	6b	153	k
12	c	014	FF	44	2c	054	,	76	4c	114	L	108	6c	154	l
13	d	015	CR	45	2d	055	-	77	4d	115	M	109	6d	155	m
14	e	016	SO	46	2e	056	.	78	4e	116	N	110	6e	156	n
15	f	017	SI	47	2f	057	/	79	4f	117	O	111	6f	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1a	032	SUB	58	3a	072	:	90	5a	132	Z	122	7a	172	z
27	1b	033	ESC	59	3b	073	;	91	5b	133	[	123	7b	173	{
28	1c	034	FS	60	3c	074	<	92	5c	134	\	124	7c	174	
29	1d	035	GS	61	3d	075	=	93	5d	135	]	125	7d	175	}
30	1e	036	RS	62	3e	076	>	94	5e	136	^	126	7e	176	~
31	1f	037	US	63	3f	077	?	95	5f	137	_	127	7f	177	DEL

실습 3 - 설명

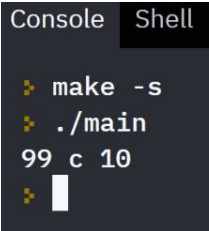
- 문자로 연산하기
 - 다음 코드를 편집기에 입력한 뒤 실행해보세요.
 - 'Wn' 같은 제어문자도 저장해서 사용할 수 있습니다.

```
#include <stdio.h>

int main()
{
    char c1 = 'a';
    char newLine = '\n';

    printf("%d %c %d %c", c1 + 2, c1 + 2, newLine, newLine);

    return 0;
}
```



```
Console Shell
❯ make -s
❯ ./main
99 c 10
❯
```

실습 3 - 설명

- 문자로 연산하기
 - 한 개의 문자를 입력 받아 대문자는 소문자로, 소문자는 대문자로 변환하는 프로그램을 작성하시오.
 - 입력 범위: a~z, A~Z
 - 입출력 예시 (입력->출력): a->A, Z->z, 1->wrong input.
 - 입력범위 외 입력은 wrong input.으로 출력

```
Console Shell
❯ make -s
❯ ./main
a
A
❯ ./main
Z
z
❯ ./main
0
wrong input.
❯ ./main
@
wrong input.
❯ ./main
d
D
❯ ./main
E
e
❯
```

실습 3 - 설명

- 문자로 연산하기

```
#include <stdio.h>

int main()
{
    char c;
    scanf("%c", &c);

    if (c >= 'a' && c <= 'z')
        c = c - 'a' + 'A';
    else if (c >= 'A' && c <= 'Z')
        c = c - 'A' + 'a';
    else {
        printf("wrong input.\n");
        return 0;
    }
    printf("%c\n", c);

    return 0;
}
```

실습 3 - 제출

- 문자로 연산하기
 - 한개의 알파벳 소문자를 입력받아, 그 문자부터 z 까지 출력하는 프로그램을 만드시오
 - 입력 예1 : a
 - 출력 예1 : abcdefghijklmnopqrstuvwxyz
 - 입력 예1 : p
 - 출력 예1 : pqrstuvwxyz
 - 입력 범위 (a~z) 외에는 wrong input 출력

```
Console  Shell
❯ make -s
❯ ./main
a
abcdefghijklmnopqrstuvwxyz
❯ ./main
z
z
❯ ./main
m
mnopqrstuvwxyz
❯ ./main
0
wrong input
❯ ./main
!
wrong input
❯
```

실습 4

실습 4 - 설명

- getchar(), putchar() 실습
 - 한 문자를 입력하고 출력하는 예제를 실행해보세요.

```
#include <stdio.h>

int main(void) {
    printf("input: ");
    char ch = getchar();
    printf("output: ");
    putchar(ch);
    ch = getchar();
    putchar(ch);
    return 0;
}
```

Console Shell

```
❯ make -s
❯ ./main
input: a
output: a
❯ ./main
input: c
output: c
❯ ./main
input: @
output: @
❯
```

실습 4 - 설명

- 입력받은 문자의 ASCII CODE를 출력해 봅시다.

```
#include <stdio.h>

int main(void) {

    char ch;
    printf("input: ");
    ch = getchar();
    printf("ASCII code of %c: %d\n", ch, ch);

    return 0;
}
```

Console Shell

```
❯ make -s
❯ ./main
input: a
ASCII code of a: 97
❯ ./main
input: @
ASCII code of @: 64
❯ ./main
input: A
ASCII code of A: 65
❯ ./main
input: *
ASCII code of *: 42
❯
```

실습 4 - 제출

- EOF로 종료될 때까지 문자를 입력 받아 대문자는 소문자로,
- 소문자는 대문자로 출력해봅시다.
 - 입력 : 알파벳 문자, 개행문자 (['a','z'], ['A','Z'], ['\n'])
 - 예외처리: 범위 이외의 입력은 다음 문자 출력: '#'
 - 출력: 알파벳문자, 개행문자, '#' (['a','z'], ['A','Z'], ['\n'], ['#'])
 - 가능하면 getchar(), putchar()를 이용해서 구현해볼 것

```
Console  Shell
❯ make -s
❯ ./main
What Does The Fox Say?
wHAT#dOES#tHE#fOX#sAY#
█
```

실습 4 – TIPS

- 수업시간에 본 코드를 이용해서 구현해보세요.

```
#include <stdio.h>
int main(void)
{
    int c;
    while ((c = getchar()) != EOF)
        putchar(c);
    return 0;
}
```

과제 1

과제 1 - 문제

- getchar()를 이용해 알파벳 문자를 입력 받아 첫 문자는 대문자로, 나머지는 소문자로 변환하여 출력하는 프로그램을 작성하시오.
 - 입력 : 알파벳 문자 (['a','z'],['A','Z']), 개행 문자 (['\n']), 공백 문자 ([' '])
 - 예외처리 : 위 입력 이외의 character인 경우 무시한다.
 - 출력 : 문제 조건에 따라 case 변경한 문자들
 - 종료조건 : '\n' 입력되었을 때

```
Console  Shell
> make -s
> ./main
what does the fox say?
What does the fox say
> ./main
What Does The Fox Say?
What does the fox say
> ./main
wHAT DOES THE FOX SAY?
What does the fox say
> 
```

과제 1 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 문자를 입력 받는가? - 2점
- 기준 3
 - 정해진 형식대로 결과를 출력하고 계산이 정확한가? - 2점

과제 2

과제 2 - 문제

- 아주 간단한 암호 해독 프로그램
- 조건 1
 - 사용자는 다음과 같이 입력한다고 가정 : [자연수] ($1 \leq n \leq 26$)
 - 입력 예시) 10
 - 이외의 입력은 예외처리한다.
- 조건 2
 - 이를 해독하기 위해, 입력된 숫자를 순서에 해당하는 대문자 알파벳으로 변환 후,
 - 오른쪽으로 다섯 칸 SHIFT 한다. (* A의 옆은 B, B의 옆은 C, ..., Z의 오른쪽은 A입니다.)
- 조건 3
 - 입력된 위치에서 다섯칸 칸 오른쪽으로 SHIFT 한 위치의 해독된 알파벳을 출력
 - 입출력 예시1) 입력 - 10, 출력 - O
 - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 - 입출력 예시2) 입력 - 25, 출력 - D
 - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

```
Console Shell
❯ make -s
❯ ./main
10
0
❯ ./main
25
D
❯
```

과제 2 - 평가 기준

- 기준 1
 - 프로그램이 동작하는가? - 1점
- 기준 2
 - 정상적으로 1개의 숫자를 입력 받는가? - 2점
- 기준 3
 - 정해진 형식 대로 결과를 출력하고 문자가 정확한가? - 2점

실습과 과제의 범위 및 기한

- 실습 3/25 23:59 까지
 - 실습2
 - 실습3
 - 실습4
- 과제 3/29 23:59 까지
 - 과제1
 - 과제2