

Jaewon Choi
Keunsan Park
Haeun Lee
(snucsl.ta@gmail.com)

Systems Software &
Architecture Lab.

Seoul National University

Spring 2023

4190.103A-001: Programming Practice Lab. 4



공지

실습 외 시간 질문

- 이메일 : snuosl.ta@gmail.com
 - 이름 및 학번을 알면 튜터링 기능을 통해 여러분 코드 열람이 가능합니다.
 - 실습 및 과제 질문은 조교 이메일 통해 해주세요.
- Office hour
 - 매주 화요일 오후 4:00 ~ 5:00 zoom을 통해 운영
 - eTL 공지 확인 바랍니다.



☰ 프로그래밍연습 (001) > 공지

2023년 1학기

전체보기

검색

홈

공지

강의계획서

수강생

수강생 알림

질문 zoom 주소 공지 (매주 화요일 업데이트)



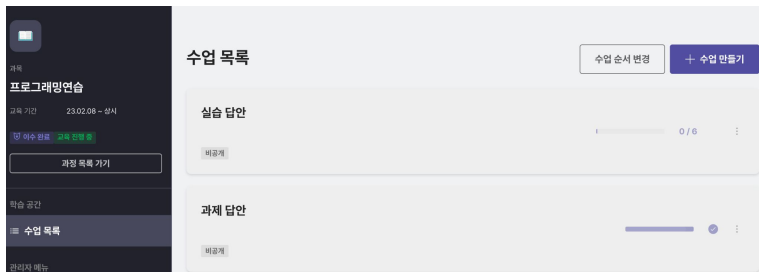
최

모든 섹션

화요일 pm 4:00 - 5:00 운영되며 이곳에 매주 줌 링크 업더

실습 & 과제 정답

- Elice에 실습 답안 & 과제 답안 목록에 제출 기한 지난 후 업데이트 예정



Lab. 3 실습 풀이

실습 1

두 양의 정수를 입력받고 큰 수를 작은 수로 나눴을 때의 몫과 나머지를 출력하세요.

그리고 두 수가 제곱의 관계인지도 (작은 수의 제곱이 큰 수와 같은지) 출력하세요.

- 제곱의 관계 : a의 제곱과 b가 같음

입출력 예시

입력 출력

예시 1	2	Quotient : 2
	4	Remainder : 0
		2 squared is 4

예시 2	8	Quotient : 2
	3	Remainder : 2
		3 squared is not 8

```
int s,b,t;
int quot,rem,flag;
scanf("%d",&s);
scanf("%d",&b);
if(s > b){
    t = s;
    s = b;
    b = t;
}
quot = b/s;
rem = b%s;

if(s*s == b)
    flag = 1;
else
    flag = 0;

printf("Quotient : %d\n",quot);
printf("Remainder : %d\n",rem);
if(flag)
    printf("%d squared is %d\n",s,b);
else
    printf("%d squared is not %d\n",s,b);
return 0;
```

실습 2

$n!$ (Factorial) 계산하는 프로그램을 만들어보세요

- 양의 정수 n ($n \leq 12$)을 입력받습니다
- $n! = (n) \times (n-1) \times (n-2) \dots \times (1)$

입력 5

출력 $5 * 4 * 3 * 2 * 1 = 120$

```
int main()
{
    // Write your code here
    int num, res=1;
    scanf("%d", &num);

    printf("%d", num);
    while(1){
        res = res * num;
        num--;
        if(num <= 0)
            break;
        printf(" * %d", num);
    }
    printf(" = %d\n", res);
    return 0;
}
```

실습 3

문자를 계속하여 입력받는 프로그램을 작성해보세요

- 'z' 가 입력이 되면, 입력받은 문자의 수를 출력한 후 프로그램을 종료합니다.

입력	a
	b
	F
	W
	r
	z
출력	6
	End

```
int main()
{
    // Write your code here
    int cnt=0;
    char c;
    while(1){
        scanf("%c",&c);
        if(c == 'z'){
            printf("%d\n",++cnt);
            break;
        }
        cnt++;
    }
    printf("End\n");
    return 0;
}
```


실습 4

양의 정수를 입력받아, 해당 양의 정수보다 작은 모든 짝수(자연수)를 출력하는 프로그램을 만들어보세요.

- 출력되는 각 수는 개행문자"\n"로 구분됩니다.

입력	10
출력	2
	4
	6
	8

```
int main()
{
    // Write your code here
    int num, even=2;
    scanf("%d", &num);
    while(even < num){
        printf("%d\n", even);
        even += 2;
    }
    return 0;
}
```

실습 5

양의 정수를 분석하는 프로그램을 만들어보세요

- 하나의 정수를 입력받습니다
- 짝수, 홀수인지 출력합니다
- 자리수를 출력합니다
- 소수인지 아닌지 확인 및 출력
 - 소수 : 약수가 1과 자기 자신뿐인 수
 - for문, %(modulo) 사용

예제 1

입력 7

출력 7 is a odd number
7 is a 1-digit number
7 is a prime number

예제 2

입력 15245424

출력 15245424 is a even number
15245424 is a 8-digit number
15245424 is not a prime number

```
int main()
{
    // Write your code here
    int num,t,flag=0,digit=0;
    scanf("%d",&num);
    if(num%2 == 0)
        printf("%d is a even number\n",num);
    else
        printf("%d is a odd number\n",num);

    t = num;
    while(t > 0){
        digit++;
        t /= 10;
    }
    printf("%d is a %d-digit number\n",num,digit);

    if(num == 1)
        flag = 1;
    for(int i=2;i<num;i++){
        if( num % i == 0){
            flag = 1;
            break;
        }
    }
    if(flag)
        printf("%d is not a prime number\n",num);
    else
        printf("%d is a prime number\n",num);

    return 0;
}
```

실습 6

Up & Down 게임을 만들어봅시다

- 컴퓨터가 $0 \leq X < 31$ 사이의 임의의 숫자를 하나 선정합니다.
- 사용자에게 5번의 기회를 주고 각 기회마다 하나의 숫자를 입력받습니다.
- 입력받은 숫자와 답을 비교해서 Up, Down 중 하나를 출력합니다.
- 답을 입력했으면 Success를 출력하고 멈추고, 정해진 횟수에 끝나지 않았을 경우 Fail을 출력합니다.
- 임의의 숫자는 주어진 코드의 "random_value"를 사용합니다.

```
int main()
{
    int random_value = rand() % 31;
    // Write your code here
    int num;
    for(int i=0;i<5;i++){
        scanf("%d",&num);
        if(num == random_value){
            printf("Success\n");
            return 0;
        }
        else if(num < random_value)
            printf("Up\n");
        else
            printf("Down\n");
    }
    printf("Fail\n");
    return 0;
}
```

Lab. 3 과제 풀이

과제 1

양의 정수를 입력받아 3진수로 변환하여 출력하는 프로그램을 만들어보세요.

- 입력으로 들어오는 최대 값은 "2,147,483,647"입니다.
- $3^{19} < 2,147,483,647 < 3^{20}$

입출력예시

입력	198372895
출력	111211021101110021

```
int main()
{
    // Write your code here
    int num,divd=1,quot,flag = 0;
    for(int i=0;i<19;i++)
        divd *= 3;
    scanf("%d",&num);

    for(int i = 0; i < 19; i++){
        quot = num / divd;
        if(flag == 0 && quot != 0){
            flag = 1;
        }
        if(flag){
            printf("%d",quot);
            num = num % divd;
        }
        divd /= 3;
    }
    printf("%d\n",num);
    return 0;
}
```

과제 2

환전소 프로그램을 작성해보세요.

- 환전할 금액(단위 : 원, 양의 정수)을 입력으로 받습니다.
- 환전 대상 통화(char)를 입력으로 받습니다. 이 때, 환전대상 통화인 엔화(Y)와 달러(D)를 제외한 입력은 무시합니다.
- 환전시에 대상 통화의 최소화페단위보다 작은 금액이 남는 경우 버립니다. (수수료로 떼갑니다.)
- 환전시에 단위가 높은 지폐를 우선적으로 선택합니다.
- 환전 가능한 통화는 엔화와 달러이며, 각각의 환율과 지폐단위는 아래와 같습니다.

달러 (환율: 1300원)

환전가능 지폐	100달러
	50달러
	10달러
	5달러

엔화 (환율: 10원)

환전가능 지폐	5000엔
	2000엔
	1000엔
	500엔

```
int main()
{
    // Write your code here
    int money, cnt;
    char currency, run;

    printf("환전소에 오신것을 환영합니다.\n");

    printf("환전할 금액을 입력하세요.\n");
    scanf("%d",&money);

    printf("환전할 통화를 입력하세요. (Y : 엔화, D : 달러)\n");
    while(1)
    {
        scanf(" %c", &currency);

        if(currency == 'Y')
            break;
        else if (currency == 'D')
            break;
    }
}
```

```
if (currency == 'Y')
{
    money = money / 10;

    cnt = money / 5000;
    money = money - cnt * 5000;
    printf("5000엔 : %d\n", cnt);

    cnt = money / 2000;
    money = money - cnt * 2000;
    printf("2000엔 : %d\n", cnt);

    cnt = money / 1000;
    money = money - cnt * 1000;
    printf("1000엔 : %d\n", cnt);

    cnt = money / 500;
    money = money - cnt * 500;
    printf("500엔 : %d\n", cnt);
}
else if (currency == 'D')
{
    money = money / 1300;

    cnt = money / 100;
    money = money - cnt * 100;
    printf("100달러 : %d\n", cnt);

    cnt = money / 50;
    money = money - cnt * 50;
    printf("50달러 : %d\n", cnt);

    cnt = money / 10;
    money = money - cnt * 10;
    printf("10달러 : %d\n", cnt);

    cnt = money / 5;
    money = money - cnt * 5;
    printf("5달러 : %d\n", cnt);
}
```

Tips

Compile Error / Warning Message

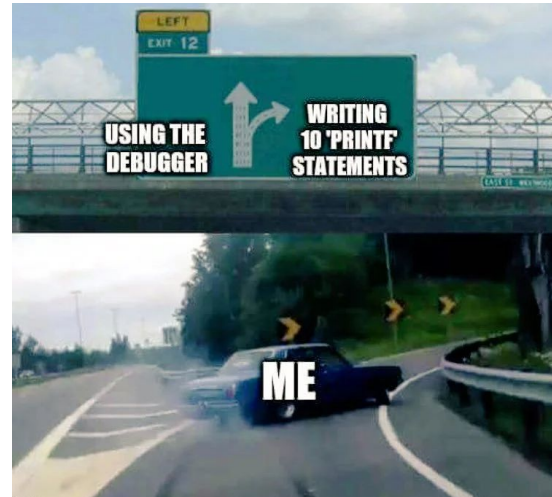
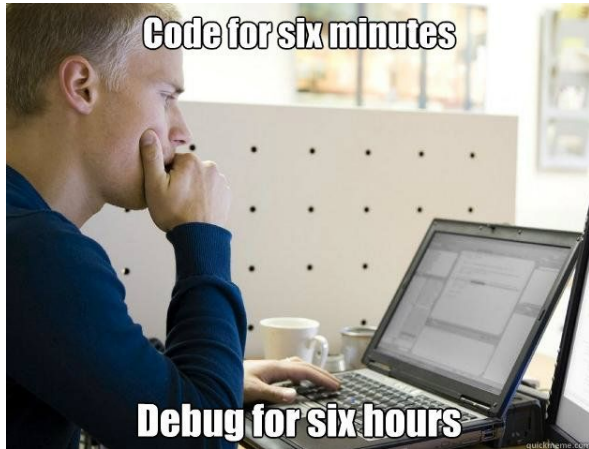
- Elice에서 발생하는 에러와 경고 메시지를 만났을 때 당황하지 말고
- 메시지를 찬찬히 잘 읽어보면 해결의 실마리가 있을 수 있습니다.
- 주로 문법 오류 때문에 발생하는데,
 - 키워드의 맞춤법이 틀리거나
 - 문장부호를 빠뜨리는 경우
 - 선언되지 않은 변수 사용 등

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

Debugging

- 소스 코드에서 버그를 찾고 수정하는 과정
- 코드를 잘짜는 것 만큼 중요한 능력
- printf()를 활용하여 문제가 되는 구간을 찾아내 봅시다.



논리 연산자(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,

```
1 #include<stdio.h>
2 #include<stdbool.h>
3
4 int main(){
5
6     bool flag = (6 > 3) && (5 < 3);
7     printf("%d\n", flag);
8     printf(flag ? "true\n" : "false\n");
9     printf("%s\n", flag ? "true" : "false");
10
11     return 0;
12 }
```

결과

```
/* 코드가 실행되는 중입니다... */
0
false
false

/* 코드 실행이 완료되었습니다! */
```

=

```
if(flag)
    printf("true\n");
else
    printf("false\n");
```

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.

```
bool flag = (6 > 3) || (5 < 3);  
printf("%d\n", flag);  
printf(flag ? "true\n" : "false\n");  
printf("%s\n", flag ? "true" : "false");
```

```
/* 코드가 실행되는 중입니다... */  
1  
true  
true  
  
/* 코드 실행이 완료되었습니다! */  
□
```

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.

```
bool flag = (6 > 3) || (5 < 3);  
printf("%d\n", !flag);  
printf(!flag ? "true\n" : "false\n");  
printf("%s\n", !flag ? "true" : "false");
```

```
/* 코드가 실행되는 중입니다... */  
0  
false  
false  
/* 코드 실행이 완료되었습니다! */
```

문자(character)

ASCII

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

```
int main(){
    char c1,c2;
    c1 = 'a';
    c2 = 'F';
    printf("%d\n",c1+c2);

    return 0;
}
```

```
/* 코드가 실행되는 중입니다... */
167
/* 코드 실행이 완료되었습니다! */
```


getchar() / putchar()

```
int main(){  
  
    int c1,c2;  
    printf("Input1 : ");  
    c1 = getchar();  
    putchar(c1);  
    putchar('\n');  
  
    printf("Input2 : ");  
    c2 = getchar();  
    putchar(c2);  
  
    return 0;  
}
```

```
Input1 : a  
a  
Input2 :
```

```
int main(){  
  
    int c1,c2;  
    printf("Input1 : ");  
    c1 = getchar();  
    putchar(c1);  
    putchar('\n');  
  
    printf("Input2 : ");  
    c2 = getchar();  
    putchar(c2);  
  
    return 0;  
}
```

```
Input1 : a  
a  
Input2 : b  
b
```

```
int main(){  
  
    char c1,c2;  
    printf("Input1 : ");  
    scanf("%c",&c1);  
    printf("%c\n",c1);  
  
    printf("Input2 : ");  
    scanf(" %c",&c2);  
    printf("%c\n",c2);  
  
    return 0;  
}
```

* 엔터(\n)까지 버퍼에 존재

getchar() / putchar()

- 한 줄(line)에 찍 쓴 후 엔터를 친 경우

```
int c;  
while((c = getchar()) != EOF)  
    putchar(c);  
return 0;
```

```
/* 코드가 실행되는 중입니다... */  
abcdefg  
abcdefg  
█
```

switch문

Switch문

- 변수와 동일한 값을 가지는 case :로 이동하여 실행문을 순차적으로 실행
- 변수와 동일한 값을 가지는 case :가 없다면 default : 로 이동하여 순차적으로 실행
- break 문을 만나면 switch를 빠져나감, break가 없다면 아래 구문들을 순차적으로 실행

```
switch(변수) {
    case 값1 :                // if (변수 == 값1) { }
        실행문1;
        break; //switch 문을 빠져나감
    case 값2 :                // else if (변수 == 값2) { }
        실행문2;
        break;
    default :                 // else { }
        실행문3;    }
```

```
char c;
switch( c = getchar()){
    case 'a' :
        printf("a\n");
        break;
    case 'a' + 1 :
        printf("b\n");
        break;
    case 99 :
        printf("c\n");
        break;
    default :
        printf("not a or b or c\n");
        break;
}
```

실습 & 과제

과제 / 실습

- 과제 내용
 - 당일 진행한 실습과 관련된 문제
- 제출 기한
 - 실습 : 당일 자정까지, 다음날 자정까지 지각제출 허용 (단, 점수 -30%)
 - 과제 : 다음 실습 수업 전날 23:59 까지
- 제출 방법
 - Elice의 Submit 기능 활용