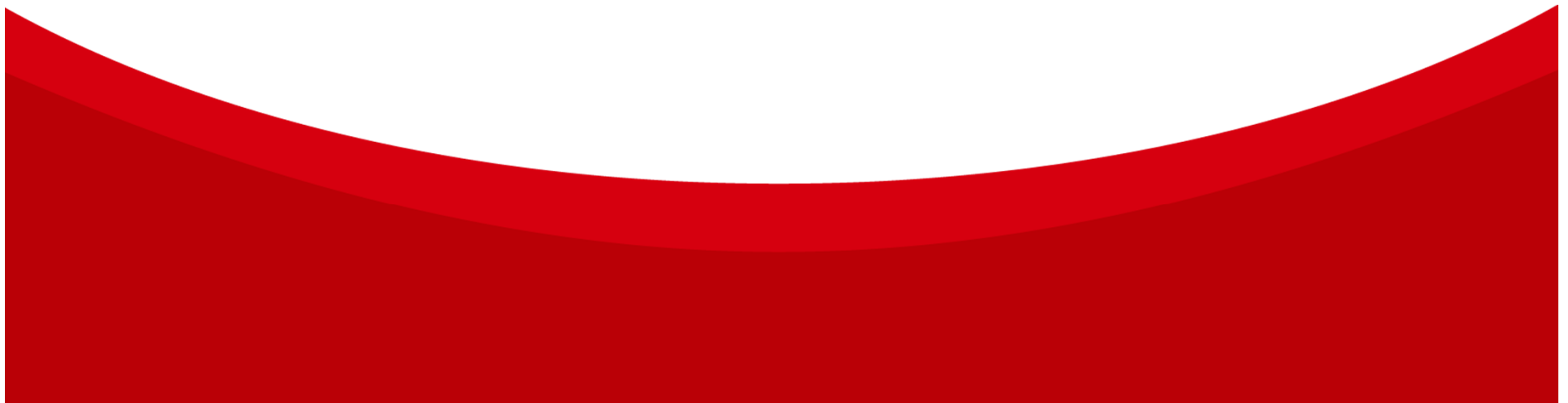


Chapter 6

반복제어문1



01

Question

알파벳 'A'부터 'Z'까지 출력하는 프로그램을 작성하시오.

출력 예

ABCDEFGHIJKLMNOPQRSTUVWXYZ

소스

```
#include <stdio.h>

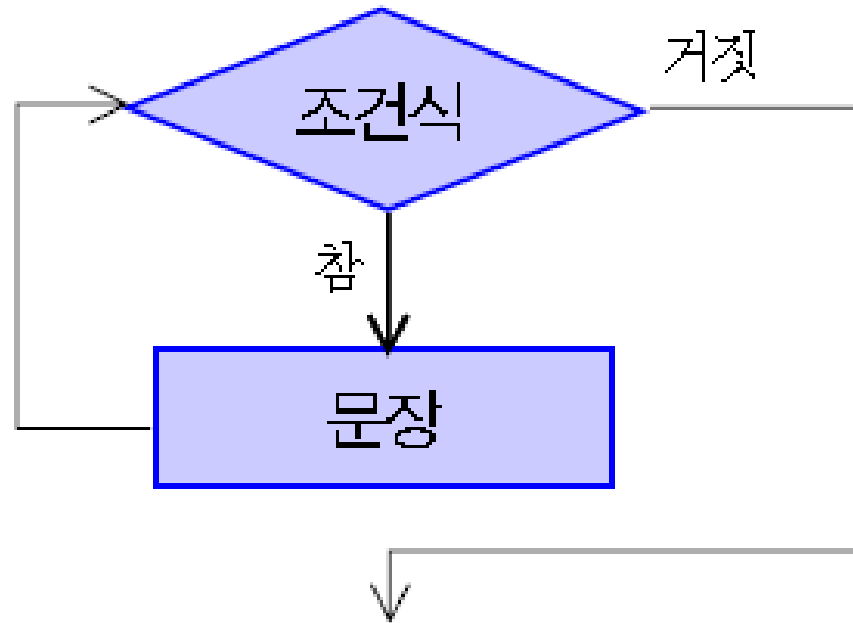
int main()
{
    char alpha = 'A';

    while (alpha <= 'Z') {
        printf("%c", alpha++);
    }
    printf("\n");

    return 0;
}
```

설명

- while문은 if문과 같이 () 안의 조건을 확인하여 조건이 참이면 실행을 하는데 if문은 단 한 번만 실행을 하고 끝나지만, while문은 한 번 실행을 하고나서 다시 조건을 확인하여 조건이 거짓이 될 때까지 계속 실행을 반복하는 것이다.



- `char alpha = 'A';`
문자변수를 선언하고 초기값으로 'A'를 대입한다.
- `while (alpha <= 'Z') { .. }`
조건이 참인 동안 블록내의 문장을 실행한다.
위 문장은 변수의 값이 'Z' 이하인 동안 계속 반복 실행하게 된다.
- `printf("%c", alpha++);`
변수의 값을 출력한 후 증가시킨다. (처음에는 alpha에 초기값으로 저장된 'A'를 출력하고 alpha의 값을 'B'로 변경한다.)

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

자가진단 1

1부터 15까지 차례로 출력하는 프로그램을 작성하시오

출력 예

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

- ~~hint~~

int 변수를 선언하고 초기값으로 1을 대입한다.

02

Question

정수 변수 num을 선언하여 1을 대입한 후 1씩 증가시키면서 10까지의 누적 합을 구하는 프로그램을 while문을 이용하여 작성하고, 1부터 10까지의 합과 while문이 끝난 후의 num의 값을 출력하는 프로그램을 작성하시오

출력 예

1부터 10까지의 합 = 55

while문이 끝난 후의 num의 값 = 11

소스

```
#include <stdio.h>

int main()
{
    int sum = 0, num = 1;

    while (num <= 10) {
        sum += num;
        num++;
    }

    printf("1부터 10까지의 합 = %d \n", sum);
    printf("while문이 끝난 후의 num의 값 = %d \n", num);

    return 0;
}
```

설명

- `int sum=0, num=1;`

합계를 저장할 변수 `sum`을 선언하고 초기값 0을 대입한다.

합계에 더해질 값을 저장할 변수 `num`을 선언하고 초기값 1을 대입한다.

- `while (num <= 10) {
 sum += num;
 num++;
}`

`num`을 합계(`sum`)에 더한 후에 값이 증가된다.

위 작업을 반복하다가 조건을 벗어나면(`num`이 10보다 커지면)
`while` 문을 벗어난다.

`num`과 `sum` 값의 변화를 표로 살펴보면 다음과 같다.

while문 조건확인시 num의 값	1	2	3	4	5	6	7	8	9	10	11
sum += num 실행후 sum값	1	3	6	10	15	21	28	36	45	55	while문 탈출
num++ 실행후 num의 값	2	3	4	5	6	7	8	9	10	11	

자가진단 2

100 이하의 정수를 입력받아 while 문을 이용하여 1부터 입력받은 정수까지의 합을 구하여 출력하는 프로그램을 작성하시오.

입력 예 10

출력 예 55

03

Question

점수를 입력받아 80점 이상이면 합격메시지를 그렇지 않으면 불합격 메시지를 출력하는 작업을 반복하다가 0~100점 이외의 점수가 입력되면 종료하는 프로그램을 작성하시오.

입·출력 예

점수를 입력하세요. 50
죄송합니다. 불합격입니다.
점수를 입력하세요. 95
축하합니다. 합격입니다.
점수를 입력하세요. 101

Chapter 6

반복제어문1

소스1

```
#include <stdio.h>

int main()
{
    int score;

    printf("점수를 입력하세요. ");
    scanf("%d", &score);

    while (score >= 0 && score <=100) {
        if (score >= 80) {
            printf("축하합니다. 합격입니다. \n");
        }
        else {
            printf("죄송합니다. 불합격입니다. \n");
        }

        printf("\n점수를 입력하세요. ");
        scanf("%d", &score);
    }

    return 0;
}
```

설명

- 위 프로그램은 일단 점수를 입력받고 조건이 맞으면 while문을 실행하여 출력한 후 while문 내에서 다시 입력을 받아 조건이 맞으면 계속 실행하고 그렇지 않으면 프로그램을 중단하는 형식으로 작성된 소스이다.
- 이렇게 똑같은 입력을 여러 곳에서 받도록 하는 것은 좋은 방법이 아니다. 위 프로그램을 다음과 같이 바꾸어서 다시 실행해 보자.

소스2

```
#include <stdio.h>

int main()
{
    int score;

    while (1) {
        printf("점수를 입력하세요. ");
        scanf("%d", &score);

        if (score < 0 || score > 100) break;
        if (score >= 80) {
            printf("축하합니다. 합격입니다. \n\n");
        }
        else {
            printf("죄송합니다. 불합격입니다. \n\n");
        }
    }

    return 0;
}
```


설명

- `while (1) { }`
조건이 항상 참이므로 이러한 while문을 무한 반복문이라고 한다.
무한 반복문의 경우에는 블록 내에서 while문을 탈출할 수 있는 명령이 없으면 프로그램이 종료되지 않는다.
- `if (score < 0 || score > 100) break;`
만약 점수가 음수 또는 100보다 큰 값이 입력되면
while문을 강제로 종료하도록 한다.
`break;` 는 반복문을 탈출(종료)하는 명령이다.

자가진단 3

한 개의 정수를 입력받아 양수(positive integer)인지 음수(negative number)인지 출력하는 작업을 반복하다가 0이 입력되면 종료하는 프로그램을 작성하시오.

입·출력 예

```
number? 10  
positive integer  
number? -10  
negative number  
number? 0
```

04

Question

정수를 계속 입력을 받다가 0이 입력되면 0을 제외하고 이전에 입력된 자료의 수와 합계, 평균을 출력하는 프로그램을 작성하시오. (평균은 반올림하여 소수 둘째자리까지 출력한다.)

입력 예 15 88 97 0

출력 예 입력된 자료의 개수 = 3
 입력된 자료의 합계 = 200
 입력된 자료의 평균 = 66.67

소스

```
#include <stdio.h>

int main()
{
    int su;
    int sum = 0, cnt = 0;
    double avg;

    while (1) {
        scanf("%d", &su);
        if (su == 0) break;
        sum += su;
        cnt++;
    }
    avg = (double)sum / cnt;

    printf("입력된 자료의 개수 = %d \n", cnt);
    printf("입력된 자료의 합계 = %d \n", sum);
    printf("입력된 자료의 평균 = %.2lf \n", avg);

    return 0;
}
```

설명

- 입력받을 값(su), 합계(sum), 자료의 수(cnt), 평균(avg)을 저장할 변수를 선언하고 합계와 자료의 수를 0으로 초기화한다.
- while (1) { ... }
블록 내의 문장을 무한 반복하다가 break;을 만나면 벗어난다.
- scanf("%d", &su);
if (su == 0) break;
su를 입력받아서 값이 0이면 while문을 벗어난다.
- sum += su;
cnt++;

입력받은 값을 합계에 누적시키고 개수를 증가시킨다.
만약 입력받은 값이 0이라면 앞에서 while문을 벗어날 것이므로 0이 아닌 경우에
만 실행하게 된다.

- `avg = (double)sum / cnt;`
평균을 구한다.

`sum`과 `cnt`가 정수이므로 소수점 이하까지 값을 구하기 위해서는 두 값중 하나 이상을 실수로 형 변환을 해주어야 한다.

위에서는 `sum`을 `double`형으로 변환을 하였는데 이를 생략하면 계산결과의 정수부분만 `avg`에 저장이 된다.

- `printf ("입력된 자료의 평균 = %.2lf \n", avg);`
`%.2lf`는 평균을 소수 세째 자리에서 반올림하여 소수 둘째자리까지 출력한다.

자가진단 4

정수를 계속 입력받다가 100 이상의 수가 입력이 되면 마지막 입력된 수를 포함하여 합계와 평균을 출력하는 프로그램을 작성하시오. (평균은 반올림하여 소수 첫째자리까지 출력한다.)

입력 예

1 2 3 4 5 6 7 8 9 10 100

출력 예

155

14.1

- hint

입력을 받으면 먼저 누적을 시킨 후에 조건을 확인한다.

05

Question

정수를 계속 입력 받다가 0이 입력되면 입력된 수중 홀수의 합과 평균을 출력하는 프로그램을 작성하십시오 (정수 미만은 버림)

입력 예 5 8 17 6 31 0

출력 예 홀수의 합 = 53
 홀수의 평균 = 17

소스1

```
#include <stdio.h>

int main()
{
    int su;
    int sum = 0, cnt = 0;
    int avg;

    while (1) {
        scanf("%d", &su);
        if (su == 0) break;
        if (su % 2 == 0) continue;
        sum += su;
        cnt++;
    }
    avg = sum / cnt;
    printf("홀수의 합 = %d \n", sum);
    printf("홀수의 평균 = %d \n", avg);
    return 0;
}
```

설명

- if (su == 0) **break**;
입력된 수가 0이면 반복문(while)을 종료한다.
- if (su%2 == 0) **continue**;
continue;는 그 이후의 문장을 실행하지 않고 반복문(while)의 처음으로 돌아가서 다시 실행하는 명령이다. 위의 문장은 입력된 수(su)가 짝수이면 sum += su; cnt++;를 실행하지 않고 처음으로 돌아가서 다시 입력을 받게 된다.

소스2

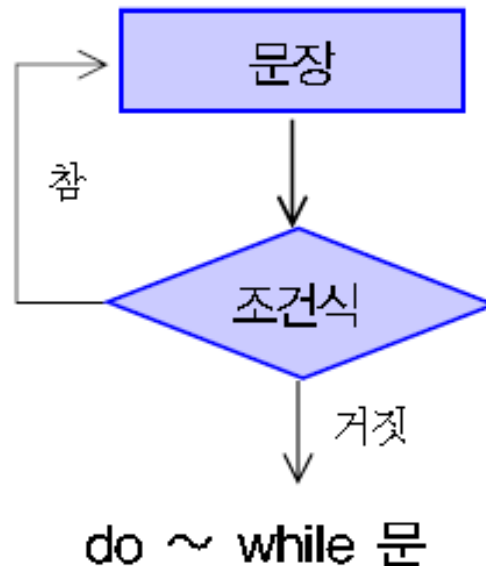
```
#include <stdio.h>

int main()
{
    int su;
    int sum = 0, cnt = 0;
    int avg;

    do {
        scanf("%d", &su);
        if (su % 2 == 1) {
            sum += su;
            cnt++;
        }
    } while (su != 0);
    avg = sum / cnt;
    printf("홀수의 합 = %d \n", sum);
    printf("홀수의 평균 = %d \n", avg);
    return 0;
}
```

설명

- while문의 경우는 처음부터 조건을 확인하여 만약 조건에 맞지 않으면 한 번도 실행하지 않는 경우도 있다.
- do ~ while문은 while문과 같지만 **조건에 상관없이 반드시 한 번은 실행을 한다**는 점이 다르다. do 다음의 내용을 실행하고 while()의 조건을 확인한다. 조건이 참이면 do부터 다시 실행하고 거짓이면 다음 줄로 이동한다.



자가진단 5

정수를 입력받아서 3의 배수가 아닌 경우에는 아무 작업도 하지 않고 3의 배수인 경우에는 3으로 나눈 몫을 출력하는 작업을 반복하다가 -1이 입력되면 종료하는 프로그램을 작성하시오 (아래 입출력 예에서 이태릭체는 입력 받은 것이고 진한 것이 출력된 것이다.)

입·출력 예

5
12
4
21
7
100
-1

• *hint*

3의 배수는 3으로 나눈 나머지가 0이다.

06

Question

아래 예와 같이 메시지를 출력하고 숫자를 입력받아 선택한 번호에 해당하는 메시지를 출력하는 작업을 반복하다가 4가 입력되면 메시지 출력 후 종료하는 프로그램을 작성하시오 (1~4 이외의 수가 입력되면 “잘못 선택하였습니다.” 라고 출력한다.)

입·출력 예

1. 입력하기
2. 출력하기
3. 삭제하기
4. 끝내기
작업할 번호를 선택하세요. 2

출력하기를 선택하였습니다.

1. 입력하기
2. 출력하기
3. 삭제하기
4. 끝내기
작업할 번호를 선택하세요. 5

잘못 선택하였습니다.

1. 입력하기
 2. 출력하기
 3. 삭제하기
 4. 끝내기
- 작업할 번호를 선택하세요. 4

끝내기를 선택하였습니다.

Chapter 6

반복제어문1

소스

```
#include <stdio.h>

int main()
{
    int num;
```



```
do {  
    printf("1. 입력하기 \n");  
    printf("2. 출력하기 \n");  
    printf("3. 삭제하기 \n");  
    printf("4. 끝내기 \n");  
  
    printf("작업할 번호를 선택하세요. ");  
    scanf("%d", &num);
```

```
switch (num) {  
case 1:  
    printf("\n입력하기를 선택 하였습니다. \n\n");  
    break;  
case 2:  
    printf("\n출력하기를 선택 하였습니다. \n\n");  
    break;  
case 3:  
    printf("\n삭제하기를 선택 하였습니다. \n\n");  
    break;  
case 4:  
    printf("\n끝내기를 선택 하였습니다. \n\n");  
    break;  
default:  
    printf("\n잘못 입력 하였습니다. \n\n");  
}  
} while (num != 4);  
  
return 0;  
}
```

설명

- `do { ... } while (num != 4);`
블록 안의 내용을 실행한 후 `num`이 4가 아니면 계속 반복 실행한다. 처음에 한 번은 무조건 실행을 한다.

Chapter 6

반복제어문1

자가진단 6

아래와 같이 나라 이름을 출력하고 숫자를 입력받아 해당하는 나라의 수도를 출력하는 작업을 반복하다가 해당하는 번호 이외의 숫자가 입력되면 “none” 이라고 출력한 후 종료하는 프로그램을 작성하시오.

입·출력 예

```
1. Korea  
2. USA  
3. Japan  
4. China  
number? 1
```

Seoul

```
1. Korea  
2. USA  
3. Japan  
4. China  
number? 5
```

none

• hint

각 나라의 수도 : 대한민국(Korea)-서울(Seoul), 미국(USA)-워싱턴(Washington), 일본(Japan)-동경(Tokyo), 중국(China)-북경(Beijing)

Thank You!!!

