

C언어 강의자료

문정욱



프로그래밍 연습 4 반복문 활용, 모듈화

for문의 탈출

■ for문의 반복조건

```
for(초기화; 반복조건; 증감)
    반복문장;
```



```
for(초기화; !(탈출조건); 증감)
    반복문장;
```

i	!(탈출조건)	printf(i)
1	true	printf(1)
2	true	printf(2)
3	true	printf(3)
4	true	printf(4)
5	false	no execution

i가 5가 되면 탈출하길 원함

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for(i=1; !(   ); ++i)
        printf("%d ", i);
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

탈출 조건
== 발견 조건

입출력 결과

1 2 3 4

계속하려면 아무 키나 누르십시오 . . .

for문의 탈출

■ 연습 (약수 발견 1)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- for문을 사용하여 2부터 1씩 증가하는 수열을 입출력 결과와 같이 출력한다. 단, for문에서 증가하는 값이 49의 약수일 경우 for문을 탈출하여 그 값을 출력하지 않는다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	!(탈출조건)	printf(i)
2	true	printf(2)
3	true	printf(3)
4	true	printf(4)
5	true	printf(5)
6	true	printf(6)
7	false	no execution

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for(i=2; !( ); ++i)
```

```
        printf("%d ", i);
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

입출력 결과

2 3 4 5 6

계속하려면 아무 키나 누르십시오 . . .

for문의 탈출

■ 연습 (약수 발견 2)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- for문을 사용하여 2부터 1씩 증가하는 수열을 입출력 결과와 같이 출력한다. 단, for문에서 증가하는 값이 49의 약수일 경우 for문을 탈출하여 "divisor == %d\n" 형식으로 그 값을 출력한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	!(탈출조건)	printf(i)
2	true	printf(2)
3	true	printf(3)
4	true	printf(4)
5	true	printf(5)
6	true	printf(6)
7	false	no execution

for문 끝났을 때 i값

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for(i=2; !( ); ++i)
        printf("%d ", i);
```

```
    printf("\n");
```

```
    printf( );
```

```
    return 0;
```

```
}
```

입출력 결과

```
2 3 4 5 6
```

```
divisor == 7
```

```
계속하려면 아무 키나 누르십시오 . . .
```


for문의 탈출

■ 연습 (약수 발견 3)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- for문을 사용하여 2부터 1씩 증가하는 수열을 입출력 결과와 같이 출력하지 않는다. 단, for문에서 증가하는 값이 49의 약수일 경우 for문을 탈출하여 "divisor == %d\n" 형식으로 그 값을 출력한다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	!(탈출조건)	printf(??)
2	true	출력 안함
3	true	출력 안함
4	true	출력 안함
5	true	출력 안함
6	true	출력 안함
7	false	no execution

for문 끝났을 때 i값

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

```
int main(void)
```

```
{
```

```
    int i;
```

```
    for(i=2; !( ); ++i)
```

```
        printf( );
```

```
    printf( );
```

```
    printf( );
```

```
    return 0;
```

```
}
```

입출력 결과

```
divisor == 7
```

```
계속하려면 아무 키나 누르십시오 . . .
```

for문의 탈출

발견한 값만 출력 1

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=2; !(49%i==0); ++i)
        printf( "    ");
    printf( "    ");
    printf("divisor == %d\n", i);
    return 0;
}
```

필요 없는 문장

발견한 값만 출력 2

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=2; !(49%i==0); ++i)
        ; // null statement

    printf("divisor == %d\n", i);
    return 0;
}
```

for문의 탈출

for문 안에 문장을 빈 문장으로

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=2; !(49%i==0); ++i)
        ; // null statement

    printf("divisor == %d\n", i);
    return 0;
}
```

for문이 반복하는 문장

입출력 결과

```
divisor == 7
계속하려면 아무 키나 누르십시오 . . .
```

for문 안에 문장을 삭제

```
#include <stdio.h>

int main(void)
{
    int i;

    for(i=2; !(49%i==0); ++i)
        printf("divisor == %d\n", i);

    return 0;
}
```

for문이 반복하는 문장

입출력 결과

```
divisor == 2
divisor == 3
divisor == 4
divisor == 5
divisor == 6
계속하려면 아무 키나 누르십시오 . . .
```


for문의 탈출

■ 연습 (약수 발견 4)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- for문을 사용하여 b부터 e까지 1씩 증가하는 수열에서 가장 먼저 발견되는 49의 약수를 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	!(49%i==0)	null stat.
5	true	-
6	true	-
7	false	no execution

for문 끝났을 때 i값

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

```
int main(void)
```

```
{
```

```
    int b, e, i;
```

```
    scanf("%d%d", &b, &e);
```

```
    for(   ; !(49%i==0); ++i)
```

```
    ;
```

```
    printf("divisor == %d\n", i);
```

```
    return 0;
```

```
}
```

입출력 결과

```
5 10
```

```
divisor == 7
```

```
계속하려면 아무 키나 누르십시오 . . .
```

for문의 탈출

■ 연습 (약수 발견 4 - 문제점 1)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- for문을 사용하여 b부터 e까지 1씩 증가하는 수열에서 가장 먼저 발견되는 49의 약수를 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	!(49%i==0)	null stat.
44	true	-
45	true	-
46	true	-
47	true	-
48	true	-
49	false	no execution

i가 48일 때 for문이 끝났어야 함.

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

```
int main(void)
```

```
{
```

```
    int b, e, i;
```

```
    scanf("%d%d", &b, &e);
```

```
    for(   ; !(49%i==0); ++i)
```

```
    ;
```

```
    printf("divisor == %d\n", i);
```

```
    return 0;
```

```
}
```

Logical Error

입출력 결과

```
44 47
divisor == 49
계속하려면 아무 키나 누르십시오 . . .
```

for문의 탈출

■ 연습 (약수 발견 4 - 문제점 2)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- for문을 사용하여 b부터 e까지 1씩 증가하는 수열에서 가장 먼저 발견되는 49의 약수를 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	추가조건? && !(49%i==0)	null stat.
44	true	-
45	true	-
46	true	-
47	true	-
48	false	no execution

i가 48일 때 for문이 끝남. 하지만...

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=b;   && !(49%i==0); ++i)
        ;
    printf("divisor == %d\n", i);
    return 0;
}
```

입출력 결과

```
44 50
divisor == 49
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
44 47
divisor == 48
계속하려면 아무 키나 누르십시오 . . .
```

Logical Error

for문의 탈출

■ 연습 (약수 발견 4 - 완성)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- for문을 사용하여 b부터 e까지 1씩 증가하는 수열에서 가장 먼저 발견되는 49의 약수를 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	(b<=i && i<=e) && !(49%i==0)	null stat.
44	true	-
45	true	-
46	true	-
47	true	-
48	false	no execution

$b \leq i \leq e$ 일 때만 i를 약수로 인정.

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=b; (b<=i && i<=e) && !(49%i==0); ++i)
    {
        if( )
            printf("divisor == %d\n", i);
        else
            printf("not found.\n");
    }
    return 0;
}
```

입출력 결과

```
44 50
divisor == 49
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
44 47
not found.
계속하려면 아무 키나 누르십시오 . . .
```

Ok

for문의 탈출

순방향 검색의 범위조건의 간소화

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=b; (b<=i && i<=e) && !(49%i==0); ++i)
    {
        if( b<=i && i<=e )
            printf("divisor = %d\n", i);
        else
            printf("not found.\n");
        return 0;
    }
}
```

i는 b에서 1씩 증가하므로
항상 $b \leq i$ 는 항상 참(true)이다.

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=b; i<=e && !(49%i==0) ; ++i)
    {
        if( i<=e )
            printf("divisor = %d\n", i);
        else
            printf("not found.\n");
        return 0;
    }
}
```


for문의 탈출

역방향 검색의 범위조건의 간소화

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=e; (b<=i && i<=e) && !(49%i==0); --i)
        ;
    if( b<=i && i<=e )
        printf("divisor = %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

i는 e에서 1씩 감소하므로
항상 i<=e는 항상 참(true)이다.

```
#include <stdio.h>

int main(void)
{
    int b, e, i;

    scanf("%d%d", &b, &e);
    for(i=e; b<=i && !(49%i==0) ; --i)
        ;
    if( b<=i )
        printf("divisor = %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

for문의 탈출

■ 연습 (약수 발견 4 - 주의사항)

- 주어진 소스코드는 주어진 값의 범위에서 49의 약수를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- for문을 사용하여 b부터 e까지 1씩 증가하는 수열에서 가장 먼저 발견되는 49의 약수를 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	(b<=i && i<=e) && !(49%i==0)	null stat.
45	true && !(false)	-
46	true && !(false)	-
47	true && !(false)	-
48	true && !(false)	-
49	false && !(true)	no execution

$b \leq i \leq e$ 일 때만 i를 약수로 인정.

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

```
int main(void)
```

```
{
```

```
    int b, e, i;
```

```
    scanf("%d%d", &b, &e);
```

```
    for(i=b; (b<=i && i<=e) && !(49%i==0); ++i)
```

```
    ;
```

```
    if( 49%i==0 )
```

```
        printf("divisor == %d\n", i);
```

```
    else
```

```
        printf("not found.\n");
```

```
    return 0;
```

```
}
```

입출력 결과

```
45 50
```

```
divisor == 49
```

```
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
45 48
```

```
divisor == 49
```

```
계속하려면 아무 키나 누르십시오 . . .
```

Logical Error

for문의 탈출

■ 연습 (최대값 찾기)

- 주어진 소스코드는 주어진 범위에서 주어진 조건을 만족하는 최대값을 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 b, e에 저장한다.
- b이상 e이하 범위에서 210의 약수이고 7의 배수인 정수 중 최대값을 찾아 그 값을 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

```
for(초기화; 범위조건 && !(발견조건); 증감)
;
if(범위조건) {
    발견시 처리 문장;
}
else {
    미발견시 처리 문장;
}
```

발견조건 == 탈출조건

최소값: 순방향 검색
최대값: 역방향 검색

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

```
int main(void)
```

```
{
```

```
    int b, e, i;
```

```
    scanf("%d%d", &b, &e);
```

```
    for(i=e; b<=i && !(210%i==0 && i%7==0) ;--i)
```

```
    ;
```

```
    if( b<=i )
```

```
        printf("max == %d\n", i);
```

```
    else
```

```
        printf("not found.\n");
```

```
    return 0;
```

```
}
```

입출력 결과

```
30 80
```

```
max == 70
```

```
계속하려면 아무 키나 누르십시오 . . .
```

입출력 결과

```
30 34
```

```
not found.
```

```
계속하려면 아무 키나 누르십시오 . . .
```

for문의 활용 형태 3가지

■ 연습 (반복 회수의 활용)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 a와 b에 순서대로 저장한다.
- a^b를 계산하여 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
2 8
2 to the 8th power == 256
```

계속

입출력 결과

```
5 4
5 to the 4th power == 625
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
{
```

```
    int a, b;
    int i, m=1;
```

```
    scanf("%d%d", &a, &b);
```

```
    for(  )
        m*=a;
```

```
    printf("%d to the %dth power == %d\n",
           a, b, m);
    return 0;
```

```
}
```


for문의 활용 형태 3가지

■ 연습 (반복 범위의 활용)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 1개를 입력 받아 int형 변수 n에 저장한다.
- n!를 계산하여 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
5
5! == 120
```

계속

입출력 결과

```
10
10! == 3628800
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int n;
    int i, m=1;

    scanf("%d", &n);

    for(  )
        m*=i;

    printf("%d! == %d\n", n, m);
    return 0;
}
```


for문의 활용 형태 3가지

■ 연습 (반복 후 값 활용)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 2개를 입력 받아 int형 변수 a와 b에 순서대로 저장한다.
- a와 b의 최소공배수를 계산하여 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
12 18
LCM == 36
```

계속

입출력 결과

```
1 1
LCM == 1
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
{
```

```
    int a, b;
    int i;
```

```
    scanf("%d%d", &a, &b);
```

```
    for( )
    ;
    printf("LCM == %d\n", i);
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i,j;
```

```
    for(j=0;j<10;j=j+1) {
```

```
    }
```

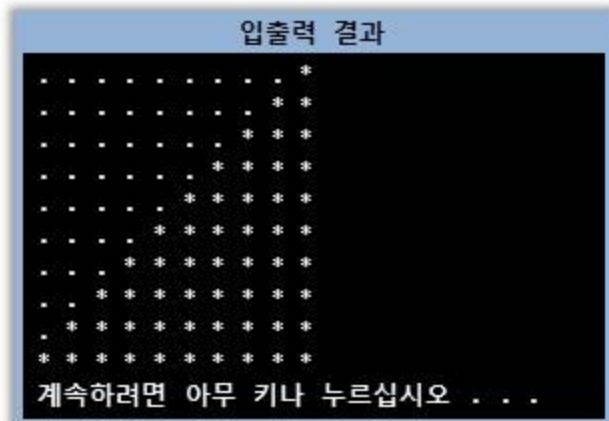
```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i,j;
```

```
    for(j=0;j<10;j=j+1) {
```

```
    }
```

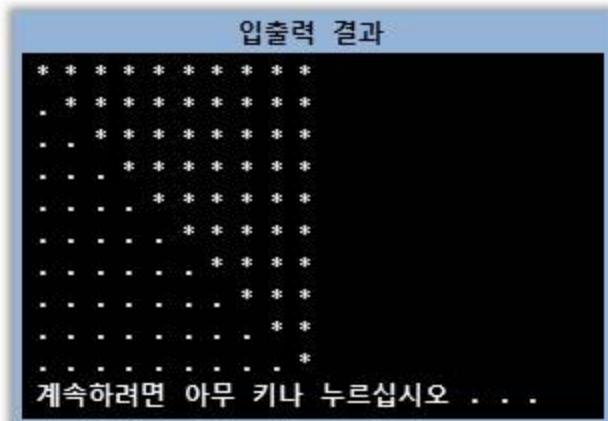
```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i,j;
```

```
    for(j=0;j<10;j=j+1) {
```

```
        
```

```
        
```

```
        
```

```
        
```

```
        
```

```
    }
```

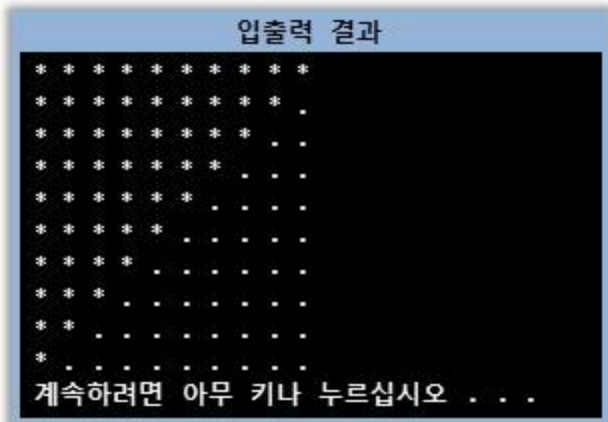
```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

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

```
int main(void)
```

{

```
int n1,n2,i,j;
```

```
for(j=0;j<10;j=j+1) {
```


}

```
return 0;
```

}

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2	n3

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

```
int main(void)
```

```
{
```

```
    int n1,n2,n3,i,j;
```

```
    for(j=0;j<10;j=j+1) {
```

```
    }
```

```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2	n3

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

```
int main(void)
```

{

```
int n1,n2,n3,i,j;
```

```
for(j=0;j<10;j=j+1) {
```

[illegible]

}

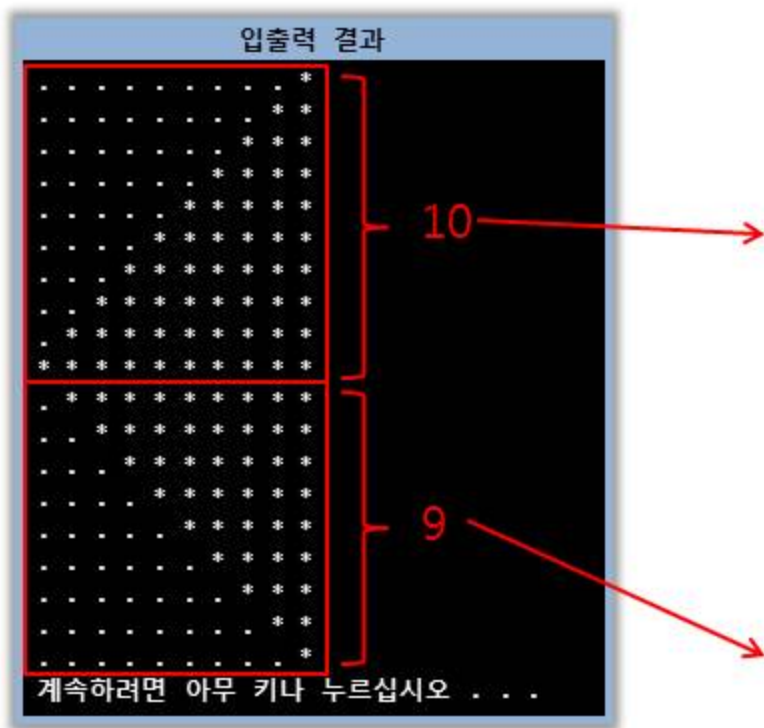
```
return 0;
```

}

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

j	n1	n2

반복문의 활용: 도형 출력

j	n1	n2

j	n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i,j;
```

```
    for( ) {
```

```
        
```

```
        
```

```
        
```

```
        
```

```
        
```

```
    }
```

```
    for( ) {
```

```
        
```

```
        
```

```
        
```

```
        
```

```
        
```

```
    }
```

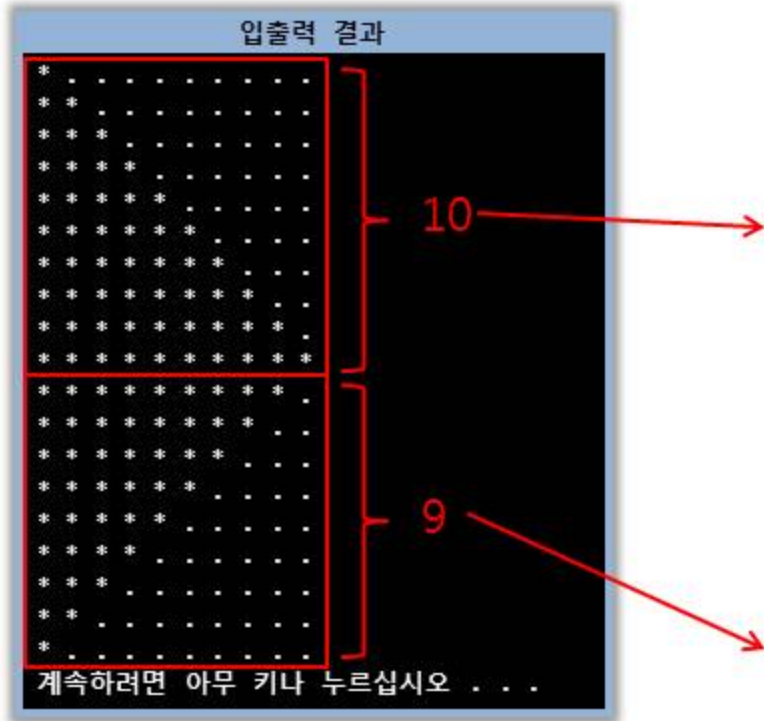
```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2

j	n1	n2

반복문의 활용: 도형 출력

j	n1	n2

j	n1	n2

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

```
int main(void)
```

```
{
```

```
    int n1,n2,i,j;
```

```
    for( ) {
```

```
        
```

```
        
```

```
        
```

```
        
```

```
        
```

```
    }
```

```
    for( ) {
```

```
        
```

```
        
```

```
        
```

```
        
```

```
        
```

```
    }
```

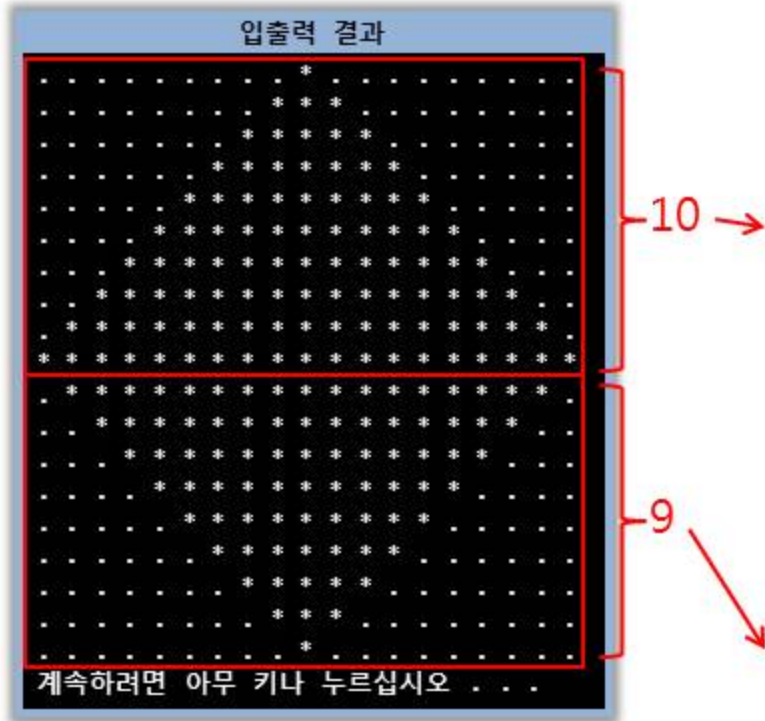
```
    return 0;
```

```
}
```

반복문의 활용: 도형 출력

■ 연습

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



j	n1	n2	n3

j	n1	n2	n3

반복문의 활용: 도형 출력

j	n1	n2	n3

j	n1	n2	n3

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

```
int main(void)
```

```
{
```

```
    int n1,n2,n3,i,j;
```

```
    for( ) {
```

```
        for( ) {
```

```
            for( ) {
```

```
            }
```

```
        } for( ) {
```

```
            for( ) {
```

```
                for( ) {
```

```
                }
```

```
            } return 0;
```

```
}
```

반복문의 활용: 배열의 조작

■ 연습

- 주어진 소스코드는 배열의 값을 순서대로 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

11 22 33 44 55

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5]={11,22,33,44,55};
    int i;

    for( )
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 주어진 소스코드는 배열의 값을 역순으로 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
55 44 33 22 11
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a[5]={11,22,33,44,55};
    int i;

    for( )
        printf("%d ", a[i]);
    printf("\n");
    return 0;
}
```


반복문의 활용: 배열의 조작

■ 연습

- 주어진 소스코드는 배열의 값의 합을 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

sum == 165

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int i, s=0;

    for( )
    {
        printf("sum == %d\n", s);
        return 0;
    }
```

반복문의 활용: 배열의 조작

■ 연습

- 주어진 소스코드는 배열 a, b의 원소간의 합을 계산하여 배열 c에 저장하고 그 값들을 순서대로 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
12 23 34 45 56
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int b[5] = { 1, 1, 1, 1, 1};
    int c[5];
    int i;

    for(  )
        

    for( i=0; i<5; ++i )
        printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 주어진 소스코드는 배열 a, b의 원소간의 차를 계산하여 배열 c에 저장하고 그 값들을 순서대로 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

10 21 32 43 54

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int b[5] = { 1, 1, 1, 1, 1};
    int c[5];
    int i;

    for( )

    for( i=0; i<5; ++i )
        printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습 (순방향 배열 검색)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 1개를 입력 받아 int형 변수 value에 저장한다.
- 배열 a에서 앞에서 부터 검색하여 value와 같은 값을 가진 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
33
index == 2
계속
```

입출력 결과

```
44
index == 1
계속
```

입출력 결과

```
99
not found.
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a[5]={11,44,33,44,55};
    int i,value;

    scanf("%d", &value);
    for( )
    ;
    if( )
        printf("index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습 (순방향 배열 검색)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 1개를 입력 받아 int형 변수 value에 저장한다.
- 배열 a에서 앞에서 부터 검색하여 value와 같은 값을 가진 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	범위조건? && !(발견조건?)	null stat.

```
#include <stdio.h>

int main(void)
{
    int a[5]={11,44,33,44,55};
    int i,value;

    scanf("%d", &value);
    for( )
        ;
    if( )
        printf("index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```


반복문의 활용: 배열의 조작

■ 연습 (역방향 배열 검색)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 1개를 입력 받아 int형 변수 value에 저장한다.
- 배열 a에서 뒤에서 부터 검색하여 value와 같은 값을 가진 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
33
index == 2
계속
```

입출력 결과

```
44
index == 3
계속
```

입출력 결과

```
99
not found.
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int main(void)
{
```

```
    int a[5]={11,44,33,44,55};
    int i,value;
```

```
    scanf("%d", &value);
```

```
    for( )
    ;
```

```
    if( )
        printf("index == %d\n", i);
```

```
    else
        printf("not found.\n");
    return 0;
```

```
}
```

반복문의 활용: 배열의 조작

■ 연습 (역방향 배열 검색)

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 1개를 입력 받아 int형 변수 value에 저장한다.
- 배열 a에서 뒤에서부터 검색하여 value와 같은 값을 가진 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

i	범위조건? && !(발견조건?)	null stat.

```
#include <stdio.h>

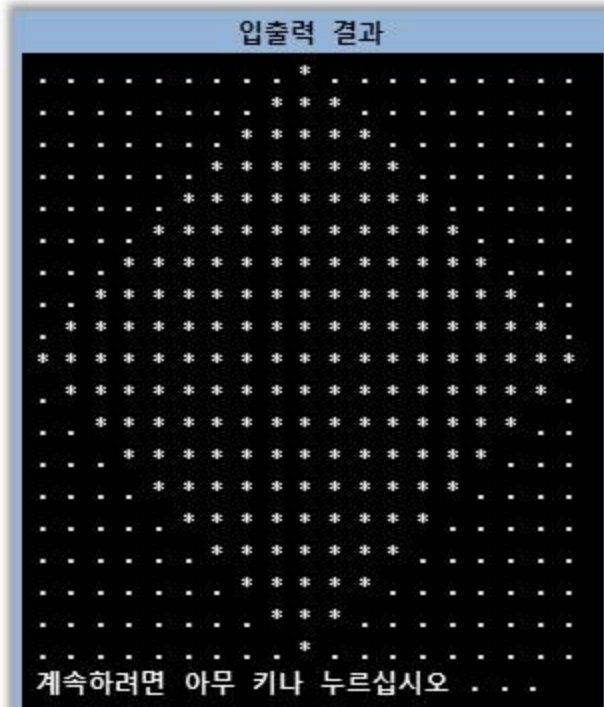
int main(void)
{
    int a[5]={11,44,33,44,55};
    int i,value;

    scanf("%d", &value);
    for( )
        ;
    if( )
        printf("index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

모듈화

■ 연습 - 단일 변수 전달

- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.



```
#include <stdio.h>

int main(void)
{
    int n1,n2,n3,i,j;

    for(j=0;j<10;j=j+1) {
        n1=9-j;
        n2=1+2*j;
        n3=9-j;
        for(i=0;i<n1;i=i+1) printf(". ");
        for(i=0;i<n2;i=i+1) printf("* ");
        for(i=0;i<n3;i=i+1) printf(". ");
        printf("\n");
    }
    for(j=0;j<9;j=j+1) {
        n1=1+j;
        n2=17-2*j;
        n3=1+j;
        for(i=0;i<n1;i=i+1) printf(". ");
        for(i=0;i<n2;i=i+1) printf("* ");
        for(i=0;i<n3;i=i+1) printf(". ");
        printf("\n");
    }
    return 0;
}
```

모듈화

```
#include <stdio.h>

int main(void)
{
    int n1,n2,n3,i,j;

    for(j=0;j<10;j=j+1) {
        n1=9-j;
        n2=1+2*j;
        n3=9-j;
        for(i=0;i<n1;i=i+1) printf(". ");
        for(i=0;i<n2;i=i+1) printf("* ");
        for(i=0;i<n3;i=i+1) printf(". ");
        printf("\n");
    }
    for(j=0;j<9;j=j+1) {
        n1=1+j;
        n2=17-2*j;
        n3=1+j;
        for(i=0;i<n1;i=i+1) printf(". ");
        for(i=0;i<n2;i=i+1) printf("* ");
        for(i=0;i<n3;i=i+1) printf(". ");
        printf("\n");
    }
    return 0;
}
```

입력: n1, n2, n3

반환: 없음

```
#include <stdio.h>

 line()
{
    int i;

    for(i=0;i<n1;i=i+1) printf(". ");
    for(i=0;i<n2;i=i+1) printf("* ");
    for(i=0;i<n3;i=i+1) printf(". ");
    printf("\n");
}

int main(void)
{
    int n1,n2,n3,j;

    for(j=0;j<10;j=j+1) {
        n1=9-j;
        n2=1+2*j;
        n3=9-j;
        
    }
    for(j=0;j<9;j=j+1) {
        n1=1+j;
        n2=17-2*j;
        n3=1+j;
        
    }
    return 0;
}
```

모듈화

■ 연습 - 배열전달

- 주어진 소스코드는 배열의 값의 합을 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

sum == 165

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int i, s=0;

    for( i=0; i<5; ++i )
        s += a[i];
    printf("sum == %d\n", s);
    return 0;
}
```


모듈화

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int i, s=0;

    for( i=0; i<5; ++i )
        s += a[i];
    printf("sum == %d\n", s);
    return 0;
}
```

입력: a(배열)

반환: s

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

```
sum( )
{
    int i, s=0;

    for( i=0; i<5; ++i )
        s += a[i];
}
```

```
int main(void)
{
    int a[5] = {11,22,33,44,55};

    printf("sum == %d\n", );
    return 0;
}
```

모듈화

■ 연습 - 배열전달 및 반환

- 주어진 소스코드는 배열 a, b의 원소간의 합을 계산하여 배열 c에 저장하고 그 값들을 순서대로 출력하는 프로그램이다.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

12 23 34 45 56

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int b[5] = { 1, 1, 1, 1, 1};
    int c[5];
    int i;

    for( i=0; i<5; ++i )
        c[i] = a[i] + b[i];

    for( i=0; i<5; ++i )
        printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

모듈화

```
#include <stdio.h>

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int b[5] = { 1, 1, 1, 1, 1};
    int c[5];
    int i;

    for( i=0; i<5; ++i )
        c[i] = a[i] + b[i];

    for( i=0; i<5; ++i )
        printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

입력: a(배열), b(배열), c(배열)

반환: 없음

배열은 반환이 불가능하다.
대신 배열은 메모리가 공유되므로
그 안에 값을 변경하기만 하면 반환의 효과가 있다.

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

```
sum( )
{
    int i;

    for( i=0; i<5; ++i )
        c[i] = a[i] + b[i];
}

int main(void)
{
    int a[5] = {11,22,33,44,55};
    int b[5] = { 1, 1, 1, 1, 1};
    int c[5];
    int i;

    for( i=0; i<5; ++i )
        printf("%d ", c[i]);
    printf("\n");
    return 0;
}
```

모듈화

■ 연습 - 반환 값의 의미

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 3개를 입력 받아 int형 변수 beg, end, value에 순서대로 저장한다.
- 배열 a의 원소들을 색인 범위 [beg, end)에서 순방향 검색할 때 value와 같은 첫 번째 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

$beg \leq i < end$

입출력 결과

```

2 6
7
found: index == 5
계속
2 5
7
not found.
계속하려면 아무 키나 누르십시오 . . .
  
```

입출력 결과

```

0 1 2 3 4 5 6
{5,7,3,9,2,7,2}
0 1 2 3 4 5 6
{5,7,3,9,2,7,2}
  
```

```

#include <stdio.h>

int search()
{
    int i;

    for()
        ;
    return i;
}

int main(void)
{
    // 0 1 2 3 4 5 6
    int a[7] = {5,7,3,9,2,7,2};
    int i, beg, end, value;

    scanf("%d%d%d", &beg, &end, &value);
    i=search(a, beg, end, value);
    if( i<end )
        printf("found: index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
  
```

모듈화

찾고 난 후 인덱스 반환

```
#include <stdio.h>

int search( int a[], int b, int e, int v )
{
    int i;

    for( i=b; i<e && !(a[i]==v); ++i )
        ;
    return i;
}

int main(void)
{
    // 0 1 2 3 4 5 6
    int a[7] = {5,7,3,9,2,7,2};
    int i, beg, end, value;

    scanf("%d%d%d", &beg, &end, &value);
    i=search(a, beg, end, value);
    if( i<end )
        printf("found: index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

찾으면 바로 인덱스 반환

```
#include <stdio.h>

int search(int a[], int b, int e, int v)
{
    int i;

    for( i=b; i<e; ++i )
        
    return i;
}

int main(void)
{
    // 0 1 2 3 4 5 6
    int a[7] = {5,7,3,9,2,7,2};
    int i, beg, end, value;

    scanf("%d%d%d", &beg, &end, &value);
    i=search(a, beg, end, value);
    if( i<end )
        printf("found: index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```


모듈화

■ 연습 - 재귀호출

- 주어진 소스코드는 배열에서 주어진 값이 저장된 배열의 인덱스를 찾아 출력하는 프로그램이다.
- 키보드로부터 정수 3개를 입력 받아 int형 변수 beg, end, value에 순서대로 저장한다.
- 배열 a의 원소들을 색인 범위 [beg, end)에서 순방향 검색할 때 value와 같은 첫 번째 원소의 인덱스를 출력하라.
- 주어진 입출력 결과를 참고하여 소스코드를 완성하라.

입출력 결과

```
2 6
7
found: index == 5
계속
```

입출력 결과

```
2 5
7
not found.
계속하려면 아무 키나 누르십시오 . . .
```

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

```
int search( int a[], int b, int e, int v )
{
    
}
```

```
int main(void)
{
    // 0 1 2 3 4 5 6
    int a[7] = {5,7,3,9,2,7,2};
    int i, beg, end, value;

    scanf("%d%d%d", &beg, &end, &value);
    i=search(a, beg, end, value);
    if( i<end )
        printf("found: index == %d\n", i);
    else
        printf("not found.\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 배열의 특정 값 바꾸기
 - 배열에 저장된 값 중에서 3을 0으로 바꾸어라.

입출력 결과

0 0 2 0 0

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5]={3,3,2,3,3};
    int sz=5, i;

    for( )

    for( i=0; i<sz; ++i )
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 앞에서부터 배열의 특정 값 바꾸기
 - 배열에 저장된 값 중에서 3을 0으로 바꾸어라.
 - 배열의 앞에서 부터 검색하여 배열 원소의 값이 3이면 0으로 바꾸고, 만일 원소의 값이 3이 아니면 바꾸기 작업을 종료한다.

입출력 결과

```
0 0 2 3 3
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a[5]={3,3,2,3,3};
    int sz=5, i;

    for( )

    for( i=0; i<5; ++i )
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 뒤에서부터 배열의 특정 값 바꾸기
 - 배열에 저장된 값 중에서 3을 0으로 바꾸어라.
 - 이때, 배열의 뒤쪽부터 검색하여 배열 원소의 값이 3이면 0으로 바꾸고, 만일 원소의 값이 3이 아니면 바꾸기 작업을 종료한다.

입출력 결과

3 3 2 0 0

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>

int main(void)
{
    int a[5]={3,3,2,3,3};
    int sz=5, i;

    for( ; )

    for( i=0; i<5; ++i )
        printf("%d ",a[i]);
    printf("\n");
    return 0;
}
```

반복문의 활용: 배열의 조작

■ 연습

- 뒤에서부터 배열의 특정 값 바꾸기
 - 배열에 저장된 값 중에서 3을 0으로 바꾸어라.
 - 이때, 배열의 뒤쪽부터 검색하여 배열 원소의 값이 3이면 0으로 바꾸고, 만일 원소의 값이 3이 아니면 바꾸기 작업을 종료한다.
 - 그리고 앞에서 언급한 3이 아닌 원소의 값을 1증가 시킨다.

입출력 결과

3 3 3 0 0

계속하려면 아무 키나 누르십시오 . . .

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

```
int main(void)
```

```
{
```

```
    int a[5]={3,3,2,3,3};
```

```
    int sz=5, i;
```

```
    for(
```

```
        
```

```
    )
```

```
    for( i=0; i<5; ++i )
```

```
        printf("%d ",a[i]);
```

```
    printf("\n");
```

```
    return 0;
```

```
}
```

반복문의 활용: 배열의 조작

■ 특별문제

• 이진수 출력

- 입력한 정수 자리수로 만들 수 있는 모든 이진수를 오름차순으로 출력하라.

입출력 결과

```
3
0 0 0
0 0 1
0 1 0
0 1 1
1 0 0
1 0 1
1 1 0
1 1 1
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

int main(void)
{
    int a[1024]={0,};
    int sz, n=1;
    int i, j;

    scanf("%d", &sz);
    for(  ) n*=2;
    for( j=0; j<n; ++j ) {
        for( i=0; i<sz; ++i )
            printf("%d ", a[i]);
        printf("\n");
        
        
        
    }
    return 0;
}
```