

## 초6 코딩

총 40문제 · 문제와 정답·풀이 포함

### Q1 변수와 계산

변수는 값을 담은 상자야. 두 상자 a와 b의 값을 서로 바꾸는 코드를 따라가 보자.

변수 a = 5

변수 b = 3

반복 없음, 차례대로 실행:

[ a = a + b ]

[ b = a - b ]

[ a = a - b ]

실행이 끝나면 a와 b의 값은?

#### 변수 교환 (Swap)



단계	a	b
1) a=a+b	8	3
2) b=a-b	8	5
3) a=a-b	3	5

결과: 값이 서로 바뀜



- ① ① a=5, b=3
- ② ② a=3, b=5
- ③ ③ a=8, b=5
- ④ ④ a=3, b=8

🎯 정답: ② a=3, b=5

📖 핵심 아이디어: 더하기·빼기만으로 두 값 서로 바꾸기(swap).

1)  $a = a + b = 5 + 3 = 8$  (상자 a=8, b=3)

2)  $b = a - b = 8 - 3 = 5$  (상자 a=8, b=5)

3)  $a = a - b = 8 - 5 = 3$  (상자 a=3, b=5)

처음 a=5, b=3 이 a=3, b=5 로 서로 바뀌었다.

💡 임시 상자 없이도 두 값을 바꿀 수 있다니 신기하지? 실제 프로그래밍에서도 쓰는 방법이야.

Q2 조건 분기 (여러 갈래)

점수를 등급으로 바꾸는 코드야. '아니면-만약'은 위 조건이 거짓일 때만 검사해.

점수 = 85

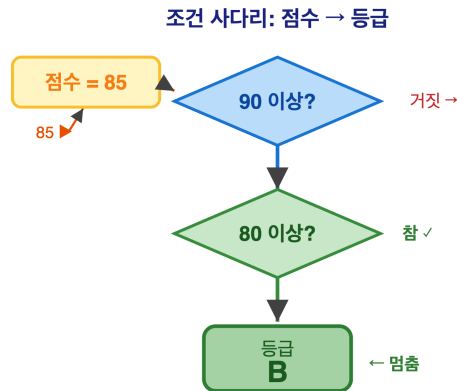
만약 점수  $\geq 90$  이면 등급 = 'A'

아니면 만약 점수  $\geq 80$  이면 등급 = 'B'

아니면 만약 점수  $\geq 70$  이면 등급 = 'C'

아니면 등급 = 'D'

등급은 무엇일까?



- ① ① A
- ② ② B
- ③ ③ C
- ④ ④ D

정답: ② B

핵심 아이디어: 조건을 위에서부터 차례로 검사, 처음 참인 곳에서 멈춤.

1)  $85 \geq 90$ ? 거짓 → 다음

2)  $85 \geq 80$ ? 참! → 등급 = 'B' 이고 아래는 검사 안 함

그래서 등급은 B.

순서가 중요해! 만약 70 검사를 맨 위에 두면 85도 C가 되어버려.

**Q3** 의사코드 읽기

누적합을 구하는 코드야. 변수 합에  $i$ 를 계속 더해 나가.

합 = 0

반복  $i = 1$  부터 4 까지 [

합 = 합 +  $i$


]

출력(합)

출력되는 값은?

- ① ① 4
- ② ② 7
- ③ ③ 10
- ④ ④ 24

 **정답: ③ 10**

 핵심 아이디어: 반복마다  $i$ 를 합에 더해 쌓기(누적).

$i=1$ : 합 =  $0+1 = 1$

$i=2$ : 합 =  $1+2 = 3$

$i=3$ : 합 =  $3+3 = 6$

$i=4$ : 합 =  $6+4 = 10$

그래서 출력은 10 ( $1+2+3+4$ ).

 1부터 100까지 더하면 5050이야. 천재 수학자 가우스가 어릴 때 빠르게 풀었대.

**Q4 중첩 반복**

반복 안에 또 반복이 들어간 '중첩 반복'이야. 별이 몇 개 찍힐까?

반복 i = 1 부터 4 까지 [

반복 j = 1 부터 i 까지 [

별 찍기 ☆

]

줄 바꿈

]

찍힌 별의 총 개수는?

**반복 안의 반복**



- ① ① 4개
- ② ② 8개
- ③ ③ 10개
- ④ ④ 16개

**정답: ③ 10개**

핵심 아이디어: 바깥 i가 정한 만큼 안쪽 j가 별을 찍어 삼각형이 된다.

i=1 → 별 1개

i=2 → 별 2개

i=3 → 별 3개

i=4 → 별 4개

총합 = 1+2+3+4 = 10개.

💡 안쪽 반복이 바깥 변수 i에 따라 길이가 달라지는 게 삼각형의 비밀이야.



Q6 알고리즘 입문

'거품정렬'은 옆 칸끼리 비교해서 왼쪽이 더 크면 자리를 바꾸는 방법이야. 가장 큰 수가 거품처럼 오른쪽 끝으로 올라가.

목록 [5, 2, 8, 1] 을 왼쪽부터 한 바퀴(옆칸 비교) 돌리면?

한 바퀴 끝난 뒤의 목록은?



- ① ① [2, 5, 8, 1]
- ② ② [2, 5, 1, 8]
- ③ ③ [1, 2, 5, 8]
- ④ ④ [5, 2, 8, 1]

🎯 정답: ② [2, 5, 1, 8]

📖 핵심 아이디어: 왼쪽이 더 크면 swap, 한 바퀴면 최댓값이 끝으로.

비교1)  $5,2 \rightarrow 5 > 2$  바꿈  $\rightarrow [2,5,8,1]$

비교2)  $5,8 \rightarrow 5 < 8$  그대로  $\rightarrow [2,5,8,1]$

비교3)  $8,1 \rightarrow 8 > 1$  바꿈  $\rightarrow [2,5,1,8]$

한 바퀴 결과 [2,5,1,8], 가장 큰 8이 맨 뒤에 자리 잡았다.

💡 이 과정을 여러 바퀴 반복하면 전체가 작은 수부터 큰 수 순서로 정렬돼.

**Q7** 순서도와 흐름

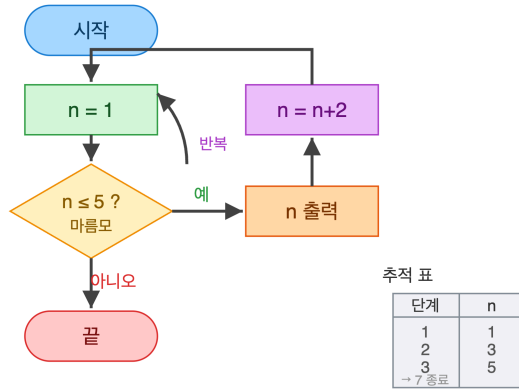
순서도(흐름도)를 따라가 보자. 마름모는 조건 검사야.

시작 →  $n = 1$  →  $[n \leq 5 ?]$

예 → 'n 출력' →  $n = n + 2$  → 다시  $[n \leq 5 ?]$

아니오 → 끝

이 순서도가 출력하는 숫자를 모두 고르고, 반복(조건이 '예')은 몇 번 일어날까?



- ① ① 1,2,3,4,5 / 5번
- ② ② 1,3,5 / 3번
- ③ ③ 1,3,5,7 / 4번
- ④ ④ 2,4 / 2번

**정답: ② 1,3,5 / 3번**

책심 아이디어: 조건이 '예'일 때만 출력하고 n을 2씩 키운다.

$n=1$ :  $1 \leq 5$  예 → 1 출력,  $n=3$

$n=3$ :  $3 \leq 5$  예 → 3 출력,  $n=5$

$n=5$ :  $5 \leq 5$  예 → 5 출력,  $n=7$

$n=7$ :  $7 \leq 5$  아니오 → 끝

출력 1,3,5 (홀수 3개), 반복은 3번.

💡  $n=n+2$  처럼 2씩 늘리면 홀수만 골라낼 수 있어. 짝수는  $n=0$ 에서 시작하면 돼.

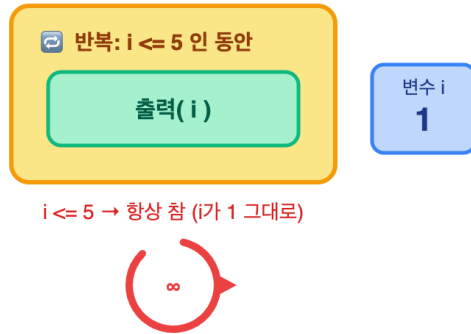
**Q8** 버그 찾기

아래 코드는 1부터 5까지 출력하려고 했지만 버그가 있어 멈추지 않아(무한 반복).

```
i = 1
반복: i <= 5 인 동안 [
출력(i)
]
```

무한 반복을 고치려면 [] 안에 무엇을 추가해야 할까?

**무한 루프 (i가 안 변함)**



- ① ①  $i = i - 1$  추가
- ② ②  $i = i + 1$  추가
- ③ ③ 출력( $i$ )를 한 번 더
- ④ ④ 조건을  $i \geq 5$  로 변경

**정답: ②  $i = i + 1$  추가**

**핵심 아이디어:** 반복을 끝내려면 조건에 쓰인 변수를 바꿔줘야 한다.

지금은  $i$ 가 계속 1이라 ' $i \leq 5$ '가 영원히 참  $\rightarrow$  무한 반복.

[ 출력( $i$ ) /  $i = i + 1$  ] 로  $i$ 를 1씩 키우면  $i$ 가 6이 되는 순간 조건이 거짓이 되어 멈춘다. (①은  $i$ 가 줄어 더 안 멈추고, ④는 처음부터 안 들게 됨)

**💡** 무한 반복은 코딩에서 흔한 실수야. '반복 변수 갱신'을 빼먹지 않았는지 꼭 확인하자.

Q9 알고리즘 입문

목록에서 가장 큰 수를 찾는 코드야. 변수 최대에 처음 값을 넣고, 더 큰 값을 만나면 바꿔.

목록 = [3, 9, 4, 9, 7]

최대 = 목록의 첫 값

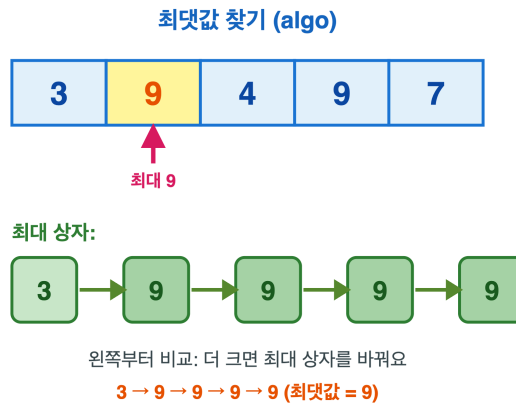
각 값 v 에 대해 [

만약  $v >$  최대 이면 최대 = v

]

출력(최대)

출력되는 값은?



- ① ① 3
- ② ② 7
- ③ ③ 9
- ④ ④ 32

🎯 정답: ③ 9

📖 핵심 아이디어: 지금까지 본 가장 큰 값을 변수 하나에 계속 기억하기.

최대=3(첫 값)

v=9:  $9 > 3$  참 → 최대=9

v=4:  $4 > 9$  거짓 → 그대로

v=9:  $9 > 9$  거짓 → 그대로

v=7:  $7 > 9$  거짓 → 그대로

출력 = 9.

💡 '>' 를 '<' 로 바꾸면 같은 방법으로 가장 작은 수를 찾을 수 있어.

**Q10** 의사코드 읽기

팩토리얼은 1부터 어떤 수까지 모두 곱한 값이야. 곱셈 누적을 추적해 보자.

곱 = 1

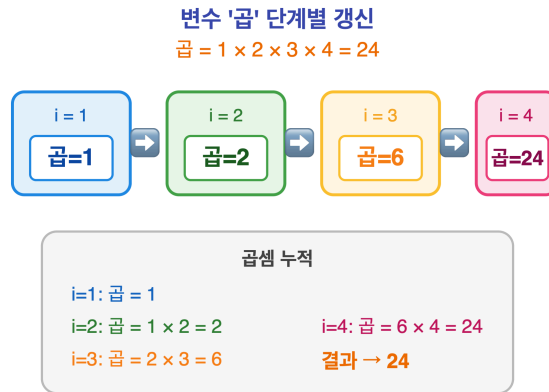
반복 i = 1 부터 4 까지 [

곱 = 곱 \* i

]

출력(곱)

출력되는 값은?



- ① ① 10
- ② ② 12
- ③ ③ 24
- ④ ④ 64

**정답: ③ 24**

**핵심 아이디어:** 더하기 대신 곱하기로 쌓으면 팩토리얼. 시작값은 반드시 1.

i=1: 곱 = 1×1 = 1

i=2: 곱 = 1×2 = 2

i=3: 곱 = 2×3 = 6

i=4: 곱 = 6×4 = 24

출력은 24 (= 1×2×3×4, '4 팩토리얼').

💡 누적합은 시작값 0, 누적곱(팩토리얼)은 시작값 1로 시작해야 해. 0으로 시작하면 곱이 영원히 0이 되거든!

**Q11** 재미 코딩

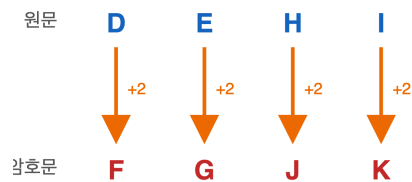
'시저 암호'는 글자를 알파벳 순서로 정해진 칸만큼 미는 방법이야.

규칙: 각 글자를 오른쪽으로 2칸 민다 (A→C, B→D, ...).

원래 글자 'H I D E' 를 2칸 밀면 어떤 글자가 될까?

(참고: ...G H I J K L M N O P...)

**암호화: 오른쪽 2칸 이동**



알파벳 띠: A B C D E F G H I J K ... Z  
D → F E → G H → J I → K

- ① ① J K F G
- ② ② F G B C
- ③ ③ I J E F
- ④ ④ K L G H

**정답: ① J K F G**

책상 핵심 아이디어: 각 글자를 알파벳에서 +2칸 이동(시저 암호).

H → I → J (2칸) = J

I → J → K = K

D → E → F = F

E → F → G = G

암호문은 'J K F G'.

💡 로마의 율리우스 카이사르(시저)가 비밀 편지에 실제로 썼대. 받는 사람은 반대로 2칸 되밀어 풀어.

Q12 재미 코딩

이진수는 0과 1만으로 수를 나타내는 방법이야. 각 자리는 오른쪽부터 1, 2, 4, 8 ... 의 값을 가져.

이진수 1 0 1 1 은 우리가 쓰는 십진수로 얼마일까?  
(자리값: 8 4 2 1)

2진수 → 10진수

8	4	2	1
1	0	1	1

$$8 + 2 + 1 = 11$$

비트가 1인 자리값만 더해요

- ① ① 7
- ② ② 11
- ③ ③ 13
- ④ ④ 1011

정답: ② 11

핵심 아이디어: 1인 자리의 자리값만 모두 더한다.

자리값: 8 4 2 1

비트: 1 0 1 1

1인 자리 = 8, 2, 1

합 = 8 + 0 + 2 + 1 = 11.

💡 컴퓨터는 전기가 켜짐(1)/꺼짐(0) 두 가지만 알아서 모든 수를 이진수로 다뤄. 손가락으로도 이진수를 세면 한 손으로 31까지 셀 수 있어!

**Q13** 변수와 계산

변수 '점수'에 다음 명령을 위에서 아래로 차례로 실행해요. (변수는 값을 담는 상자예요)

점수 = 10

점수 = 점수 + 5

점수 = 점수 × 2

마지막에 '점수' 상자에 들어 있는 값은?



- ① ① 15
- ② ② 25
- ③ ③ 30
- ④ ④ 20

**정답: ③ 30**

**핵심:** '='는 '오른쪽 값을 상자에 넣어라'라는 뜻. 한 줄씩 따라가요. 점수=10 → 10. 점수=점수+5 → 10+5=15. 점수=점수×2 → 15×2=30. 마지막 값은 30.

**💡** 프로그램에서 '='는 '같다'가 아니라 '대입(넣기)' 기호예요. '같다'는 보통 '=='로 따로 써요.

**Q14** 중첩 반복

다음 블록을 실행하면 '★'은 모두 몇 번 출력될까요?

반복 i=1..3 [

반복 j=1..4 [

★ 출력

]

]

반복 안의 반복 (i × j)

	j=1	j=2	j=3	j=4
i=1	★	★	★	★
i=2	★	★	★	★
i=3	★	★	★	★

3행 × 4열 = ★ 12개

- ① ① 7
- ② ② 12
- ③ ③ 4
- ④ ④ 3

🎯 정답: ② 12

📖 핵심: 안쪽 반복은 바깥이 한 번 돌 때마다 통째로 다시 돌아요. 바깥 i가 3번 도는데, 한 번 돌 때마다 안쪽 j가 4번. 그래서 3×4=12번.

💡 이렇게 반복 안에 반복을 넣는 걸 '중첩 반복'이라고 해요. 구구단 표를 만들 때 딱 좋아요.

**Q15** 조건 분기 (여러 갈래)

놀이공원 입장료를 나이에 따라 정해요.

만약 (나이 < 8) 이면 → 0원

아니면 만약 (나이 < 14) 이면 → 5000원

아니면 → 10000원

나이가 12살인 친구의 입장료는?

- ① ① 0원
- ② ② 5000원
- ③ ③ 10000원
- ④ ④ 15000원

🎯 정답: ② 5000원

📖 핵심: 위에서부터 검사하다가 처음으로 맞는 곳에서 멈춰요. 12<8? 아니오. 12<14? 예 → 5000원으로 결정. 아래 '아니면'은 더 보지 않아요.

💡 여러 갈래 조건은 순서가 중요해요. 큰 조건을 위에 두면 엉뚱한 갈래로 빠질 수 있어요.

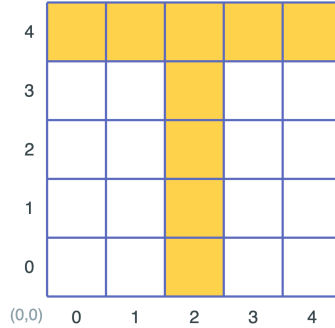
**Q16** 좌표·픽셀아트

5×5 격자(왼쪽 아래가 (0,0))에서 다음 좌표 칸을 칠해요.

(0,4),(1,4),(2,4),(3,4),(4,4),(2,3),(2,2),(2,1),(2,0)

어떤 알파벳 모양이 될까요?

알파벳 T 만들기



- ① ① T
- ② ② L
- ③ ③ H
- ④ ④ +

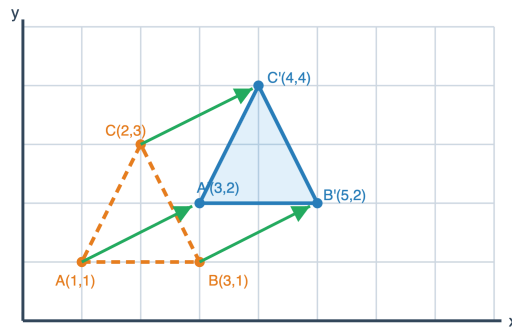
**정답: ① T**

**핵심:** 좌표를 하나씩 칠해 모양을 봐요. 맨 윗줄(y=4)이 가로로 꽉 차고, 가운데(x=2)가 아래로 쭉 내려와요. 알파벳 T 모양.

**💡** 디지털 글자(폰트)도 이렇게 작은 점(픽셀)을 켜고 꺼서 만들어요.

**Q17** 좌표·픽셀아트

삼각형의 세 꼭짓점이 (1,1), (3,1), (1,4) 에 있어요. 모든 점을 '오른쪽으로 2칸, 위로 1칸' 옮기면 새 꼭짓점 좌표는?



오른쪽 2칸, 위 1칸 이동

- ① ① (3,2),(5,2),(3,5)
- ② ② (1,3),(3,3),(1,6)
- ③ ③ (3,0),(5,0),(3,3)
- ④ ④ (2,1),(4,1),(2,4)

**정답: ① (3,2),(5,2),(3,5)**

**핵심:** 평행이동은 모든 점의 x에 +2, y에 +1을 똑같이 더해요. (1,1)→(3,2), (3,1)→(5,2), (1,4)→(3,5). 모양과 크기는 그대로, 위치만 바뀌어요.

**💡** 게임 캐릭터가 한 걸음 움직이는 것도 모든 점에 같은 수를 더하는 '평행이동'이에요.

**Q18** 알고리즘 입문

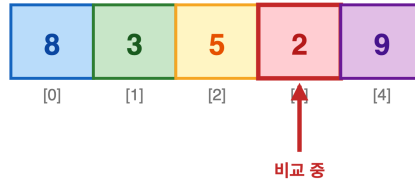
목록 [8, 3, 5, 2, 9] 에서 가장 작은 수를 찾는 방법이에요.

작은수 = 8 (첫 번째 값)

나머지를 왼쪽부터 차례로 보며, 더 작은 수를 만나면 '작은수'를 그 값으로 바꿔요.

끝까지 했을 때 '작은수'의 값은?

**최솟값 찾기: 배열 훑기**



**작은수(min) 변화 추적**



- ① ① 2
- ② ② 3
- ③ ③ 8
- ④ ④ 9

**정답: ① 2**

**핵심:** 지금까지 본 것 중 가장 작은 값을 기억해 뒀요. 시작 8.  $3 < 8 \rightarrow$  작은수=3. 5는 안 작음.  $2 < 3 \rightarrow$  작은수=2. 9는 안 작음. 최종 2.

**💡** 최댓값 찾기는 부등호만 반대로 하면 돼요. '더 큰 수를 만나면 바꿔라'.

Q19 나만의 블록 / 함수 입문

넓이를 구하는 나만의 블록(함수)을 정의했어요. (함수는 입력을 받아 결과를 돌려주는 도구예요)

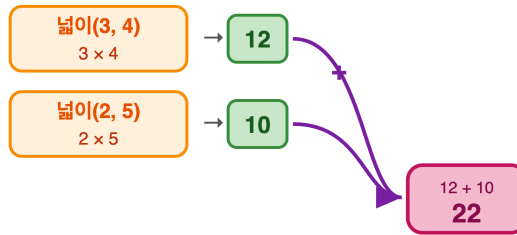
함수 넓이(가로, 세로):

돌려주기 가로 × 세로

넓이(3, 4) + 넓이(2, 5) 의 값은?

함수 정의와 호출

함수    넓이(가로, 세로):  
          돌려주기 가로 × 세로



- ① ① 22
- ② ② 14
- ③ ③ 24
- ④ ④ 17

🎯 정답: ① 22

📖 핵심: 함수에 값을 넣으면 계산 결과를 돌려줘요. 넓이(3,4)=3×4=12. 넓이(2,5)=2×5=10. 12+10=22.

💡 한 번 만든 함수는 입력만 바뀌 몇 번이고 다시 쓸 수 있어요. 같은 계산을 또 적을 필요가 없어요.

**Q20** 의사코드 읽기

다음 의사코드를 따라가면 '개수'의 최종 값은? ('나머지'는 나눈 뒤 남는 수)

개수 = 0

반복 n=1..6 [

만약 (n 을 2로 나눈 나머지 = 0) 이면 [

개수 = 개수 + 1

]

]

**2로 나눈 나머지**

n						
1	2	3	4	5	6	n
1	0	1	0	1	0	나머지
	✓		✓		✓	짝수?

**개수 세기**



- ① ① 2
- ② ② 3
- ③ ③ 6
- ④ ④ 4

**정답: ③ 3**

**핵심:** 조건에 맞을 때만 개수를 1 늘려요. 1~6 중 2로 나눈 나머지가 0인 짝수는 2,4,6 세 개. 그때마다 개수가 늘어 최종 3.

**💡** '2로 나눈 나머지'로 짝수·홀수를 가려내는 건 코딩에서 아주 자주 쓰는 꿀팁이에요.

**Q21** 재미 코딩

어떤 수가 홀수인지 짝수인지 정하는 규칙이에요.

만약 (수 ÷ 2 의 나머지 = 0) 이면 '짝수', 아니면 '홀수'.

다음 중 이 규칙으로 '홀수'가 나오는 수는?

- ① ① 14
- ② ② 20
- ③ ③ 27
- ④ ④ 8

**정답: ③ 27**

**핵심:** 2로 나눈 나머지가 0이면 짝수, 1이면 홀수.  $27 \div 2 = 13 \dots$  나머지 1 → 홀수.  $14 \cdot 20 \cdot 8$ 은 나머지가 0이라 모두 짝수.

**💡** 나머지를 구하는 'mod' 연산은 시계(12로 나눈 나머지)나 요일 계산에도 쓰여요.

**Q22** 알고리즘 입문

목록 [12, 7, 20, 5, 15, 8] 에서 '10보다 큰 수'가 몇 개인지 세요.

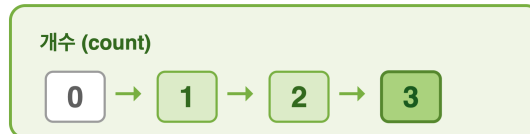
개수 = 0

각 수에 대해 → 만약 (수 > 10) 이면 개수 = 개수 + 1

10보다 큰 칸 세기



✓ = 10 초과



- ① ① 2
- ② ② 3
- ③ ③ 4
- ④ ④ 5

**정답: ② 3**

**핵심:** 모든 수를 한 번씩 검사하며 조건에 맞으면 센다. 12>10✓, 7✗, 20✓, 5✗, 15✓, 8✗. 맞는 수는 12,20,15 세 개.

**💡** 이렇게 '조건에 맞는 것만 세기'는 검색 결과 개수, 합격자 수 세기 같은 데서 똑같이 쓰여요.

**Q23** 재미 코딩

한글 자음을 숫자로 바꾸는 암호표가 있어요. ㄱ=1, ㄴ=2, ㄷ=3, ㄹ=4, ㅁ=5 처럼 순서대로 번호를 매겨요.

암호 '5-1-2'를 글자로 풀면?

- ① ① ㄹㄱㄴ
- ② ② ㄱㄴㄷ
- ③ ③ ㄷㄹㅁ
- ④ ④ ㄴㄱㅁ

**정답: ① ㄹㄱㄴ**

**핵심:** 암호표를 거꾸로 읽어 숫자를 글자로 되돌려요. 5→ㅁ, 1→ㄱ, 2→ㄴ. 그래서 'ㄹㄱㄴ'!

**💡** 이렇게 글자와 숫자를 짝지어 저장하는 방식이 실제 컴퓨터가 글자를 다루는 원리(문자 코드)예요.

**Q24** 순서도와 흐름

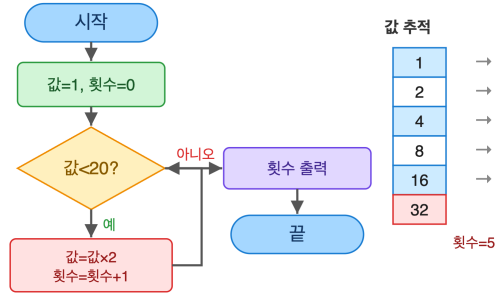
다음 순서도를 따라가요.

시작 → 값=1, 횟수=0 → [값 < 20 ?]

예 → 값=값×2, 횟수=횟수+1 (다시 판단으로 되돌아감)

아니오 → '횟수' 출력 → 끝

출력되는 '횟수'는?



- ①) ① 4
- ②) ② 5
- ③) ③ 6
- ④) ④ 3

**정답: ② 5**

**핵심:** 조건이 참인 동안만 반복하고, 둘 때마다 횟수를 센다. 값: 1→2(횟수1)→4(2)→8(3)→16(4)→32(5). 32는 20보다 커서 반복을 멈추고 횟수=5를 출력.

**마름모(조건)에서 '아니오'로 빠져나가는 순간 반복이 끝나요. 빠져나갈 길이 없으면 무한 반복이 돼요.**

Q25 변수와 계산

변수(값을 담는 상자)  $a=5$ ,  $b=3$  으로 시작해. 아래를 위에서 아래로 차례대로 실행하면 마지막에  $a$ 와  $b$ 는 각각 얼마일까?

- ①  $a = a + b$
- ②  $b = a - b$
- ③  $a = a * b$



- ① ①  $a=40$ ,  $b=5$
- ② ②  $a=15$ ,  $b=3$
- ③ ③  $a=40$ ,  $b=3$
- ④ ④  $a=24$ ,  $b=5$

🎯 정답: ①  $a=40$ ,  $b=5$

📖 핵심: 한 줄씩 차례로 상자 값을 갱신하며 따라간다. 시작  $a=5$ ,  $b=3$ . ①  $a=5+3=8$  ( $a=8$ ,  $b=3$ ). ②  $b=a-b=8-3=5$  ( $a=8$ ,  $b=5$ ). ③  $a=a*b=8*5=40$  ( $a=40$ ,  $b=5$ ). 그래서  $a=40$ ,  $b=5$ .

💡 같은 줄도 순서를 바꾸면 결과가 완전히 달라져. 그래서 프로그램에서는 '순서'가 정말 중요해!

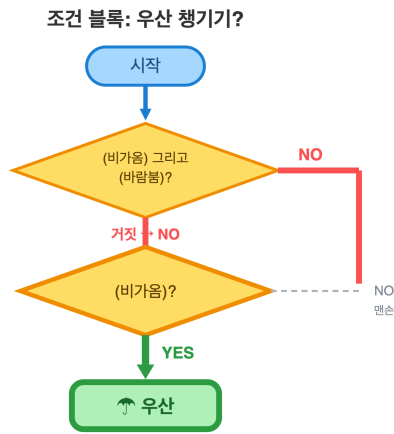
**Q26** 조건 분기 (여러 갈래)

오늘은 비가 오고(참), 바람은 불지 않아(거짓). 아래 블록을 실행하면 무엇이 출력될까?

만약 (비가옴) 그리고 (바람붐) [ 출력 '장화' ]

아니면 만약 (비가옴) [ 출력 '우산' ]

아니면 [ 출력 '맨발' ]



- ① ① 장화
- ② ② 우산
- ③ ③ 맨발
- ④ ④ 아무것도 안 함

**정답: ② 우산**

**핵심:** '그리고'는 두 조건이 모두 참일 때만 참이다. 첫 줄은 (참) 그리고 (거짓) = 거짓이라 통과 못 함. 다음 '아니면 만약 (비가옴)'은 참이라 여기서 멈추고 '우산'을 출력한다. 한 갈래가 실행되면 나머지 아니면은 건너뛴다.

**💡** '그리고(AND)'는 둘 다 참, '또는(OR)'은 하나만 참이어도 참이야. 두 글자 차이가 결과를 바꿔!

**Q27** 의사코드 읽기

아래 의사코드를 실행하면 마지막에 '합'은 얼마일까?

합 = 0

반복 n=1..5 [

만약 n이 홀수이면 [ 합 = 합 + n ]

]

- ① ① 9
- ② ② 15
- ③ ③ 6
- ④ ④ 25

**정답: ① 9**

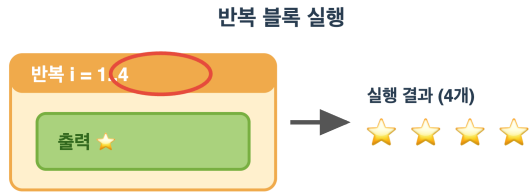
**핵심:** 1부터 5까지 가면서 홀수일 때만 더한다. 홀수는 1, 3, 5. 합 = 1+3+5 = 9. 짝수 2, 4는 조건을 통과하지 못해 더해지지 않는다.

**💡** 홀수만 차례로 더하면 1, 1+3=4, 1+3+5=9 처럼 항상 제곱수(1, 4, 9, 16...)가 돼. 신기하지?

Q28 버그 찾기

별 '★'을 5개 출력하려고 했는데 막상 돌려 보니 4개만 나와. 어디를 고쳐야 5개가 될까?

반복 `i=1..4` [ 출력 '★' ]



-----  
목표와 비교

결과 (4)  
★ ★ ★ ★

목표 (5)  
★ ★ ★ ★ ★  
+1 부족

- ① ① 반복 `i=1..4` 를 `i=1..5` 로 바꾼다
- ② ② 출력 '★' 를 두 번 쓴다
- ③ ③ 반복 블록을 통째로 지운다
- ④ ④ `i=1..4` 를 `i=2..4` 로 바꾼다

🎯 정답: ① 반복 `i=1..4` 를 `i=1..5` 로 바꾼다

📖 핵심: 반복 횟수를 세어 본다. `i=1..4`는 1, 2, 3, 4 로 딱 4번 돈다(한 끝 차이, off-by-one 버그). 5번 돌게 하려면 끝 값을 5로 바꿔 `i=1..5(1,2,3,4,5)`로 만들면 된다.

💡 '하나 모자라거나 하나 더 도는' 한 끝 차이 실수는 프로그래머가 가장 자주 하는 버그 중 하나야!

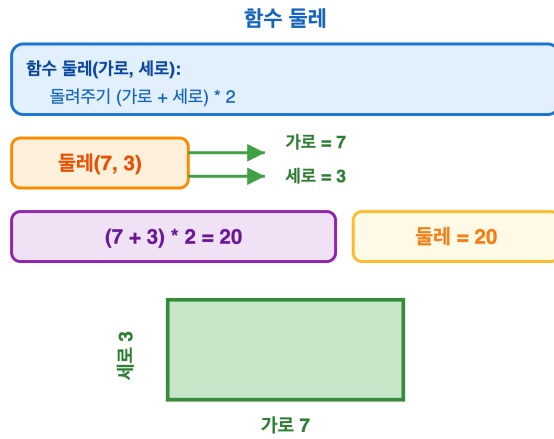
Q29 나만의 블록 / 함수 입문

입력 두 개를 받아 값을 돌려주는(반환하는) 함수를 정의했어.

함수 돌레(가로, 세로):

돌려주기 (가로 + 세로) \* 2

그러면 돌레(7, 3) 을 부르면 어떤 값이 나올까?



- ① ① 20
- ② ② 21
- ③ ③ 10
- ④ ④ 13

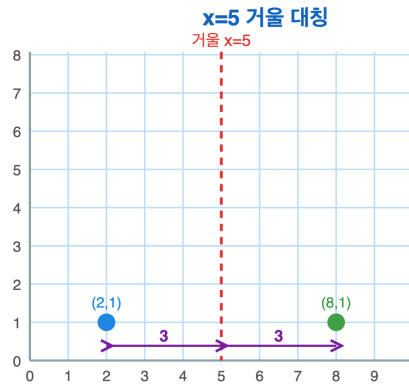
**정답: ① 20**

**핵심:** 함수를 부를 때 넣은 값이 입력 자리에 들어간다. 가로=7, 세로=3 이므로 (가로+세로)\*2 = (7+3)\*2 = 10\*2 = 20. '돌려주기'는 이 값을 결과로 내보낸다는 뜻이다.

**💡** 한 번 만든 돌레 함수는 돌레(5,2), 돌레(8,8)처럼 숫자만 바꿔 몇 번이고 다시 쓸 수 있어. 그게 함수의 힘!

**Q30** 좌표·픽셀아트

점 (2, 1) 을 세로선  $x=5$  를 거울처럼 두고 좌우로 반사(대칭)시키려 해. 규칙은 '새  $x = 10 -$  기존  $x, y$  는 그대로'. 반사된 새 좌표는?



- ① ① (8, 1)
- ② ② (3, 1)
- ③ ③ (2, 9)
- ④ ④ (8, 2)

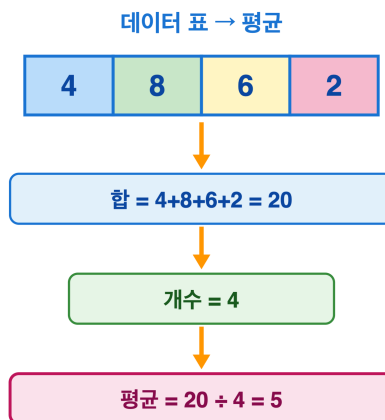
**정답: ① (8, 1)**

**핵심:** 좌우 반사는  $x$ 만 바뀌고  $y$ 는 그대로다. 규칙대로 새  $x = 10 - 2 = 8, y = 1$  유지. 따라서 (8, 1). 거울선  $x=5$  에서 원래 점은 왼쪽으로 3칸, 새 점은 오른쪽으로 3칸 떨어져 대칭을 이룬다.

**💡** 좌우 반사는  $x$ 만, 위아래 반사는  $y$ 만 바뀌. 둘을 모두 하면 점이 180도 돌아간 것처럼 보여!

**Q31** 알고리즘 입문

목록 [4, 8, 6, 2] 의 평균을 구하려 해. 모두 더한 합을 개수로 나누면 돼. 평균은 얼마일까?



- ① ① 5
- ② ② 6
- ③ ③ 20
- ④ ④ 4

**정답: ① 5**

**핵심:** 평균 = 합 ÷ 개수. 합 =  $4+8+6+2 = 20$ , 개수 = 4. 평균 =  $20 \div 4 = 5$ . 합을 변수에 차곡차곡 더한 뒤 마지막에 개수로 나누는 순서가 중요하다.

**💡** 평균은 모든 값을 '똑같이 나눠 가지면 얼마일까'를 알려줘. 그래서 가장 큰 값보다 작고 가장 작은 값보다 커.

Q32 재미 코딩

이진수(0과 1만 쓰는 수)로 셀 때 0, 1 다음은 10, 11, 100... 처럼 이어져. 그러면 이진수 101 바로 다음 수(1을 더한 수)는 무엇일까?

이진수 세기 (Binary)

십진수	이진수
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111



- ① ① 110
- ② ② 102
- ③ ③ 100
- ④ ④ 111

정답: ① 110

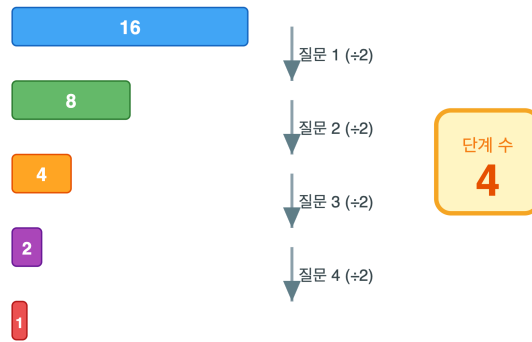
핵심: 이진수도 자리가 짝 차면 뒷자리로 올림(받아올림)이 생긴다. 101은 십진수로 5다.  $5+1=6$  이고, 6을 이진수로 쓰면 110. 맨 끝 1에 1을 더하면 10이 되어 올림이 생겨  $101 \rightarrow 110$  이 된다.

이진수에는 2라는 숫자가 아예 없어.  $1+1$ 을 하면 곧바로 10(올림)이 돼. 컴퓨터가 딱 이렇게 계산해!

Q33 알고리즘 입문

친구가 1부터 16까지 중 수 하나를 마음속에 골랐어. 너는 '그 수가  $\square$ 보다 큰가요?'만 물어볼 수 있어. 매번 '남은 범위의 한가운데'를 물어 반씩 줄여 가면, 최악의 경우라도 몇 번이면 반드시 맞힐 수 있을까?

절반씩 자르기: 16 → 1



- ① ① 16
- ② ② 8
- ③ ③ 4
- ④ ④ 2

정답: ③ 4

핵심: 한가운데를 물으면 후보가 매번 절반으로 줄어든다(이진탐색). 16 → 8 → 4 → 2 → 1 로 절반씩 4번 나누면 후보가 1개로 좁혀진다. 그래서 4번이면 충분하다.  $16=2 \times 2 \times 2 \times 2$  이므로 답이 4가 된다.

이렇게 반씩 줄이면 1024개도 단 10번이면 찾아! 하나씩 묻는 것보다 어마어마하게 빨라.

Q34 의사코드 읽기

아래 의사코드를 실행하면 반복이 끝난 뒤 b의 값은 얼마일까?

a = 1, b = 1

반복 3번 [

c = a + b

a = b

b = c

]

변수 상자와 반복



단계	a	b	변화
시작	1	1	-
1회	1	2	b+1
2회	2	3	a+b
3회	3	5	a+b

마지막 b = 5 ☆

각 단계마다 a, b 값이 바뀜

- ① ① 5
- ② ② 3
- ③ ③ 8
- ④ ④ 4

정답: ① 5

핵심: c에 두 수의 합을 잠시 담고 나서 a, b를 차례로 옮긴다(앞 두 수를 더해 다음 수를 만드는 규칙). 시작 a=1, b=1. 1회: c=2, a=1, b=2. 2회: c=3, a=2, b=3. 3회: c=5, a=3, b=5. 끝난 뒤 b=5.

이 규칙으로 나오는 수 1, 1, 2, 3, 5, 8, 13... 을 피보나치 수라고 해. 꽃잎이나 솔방울에서도 발견돼!

Q35 순서도와 흐름

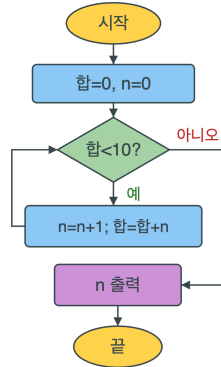
아래 순서도를 따라가면 끝에서 출력되는 n은 얼마일까?

시작 → 합=0, n=0

마름모 '합 < 10 ?'

예 → n=n+1, 합=합+n → 마름모로 되돌아감

아니오 → n 출력 → 끝



추적 표

n	합
1	1
2	3
3	6
4	10

합=10 → 멈춤 (n=4 출력)

- ①) ① 3
- ②) ② 4
- ③) ③ 5
- ④) ④ 10

정답: ② 4

핵심: 합이 10보다 작은 동안만 n을 1씩 키우며 더한다. 추적하면 n=1→합=1, n=2→합=3, n=3→합=6, n=4→합=10. 이제 '합 < 10?'은 10 < 10 이라 거짓이 되어 반복을 멈추고 n=4 를 출력한다. 조건을 '딱 넘는 순간' 멈추는 것에 주의!

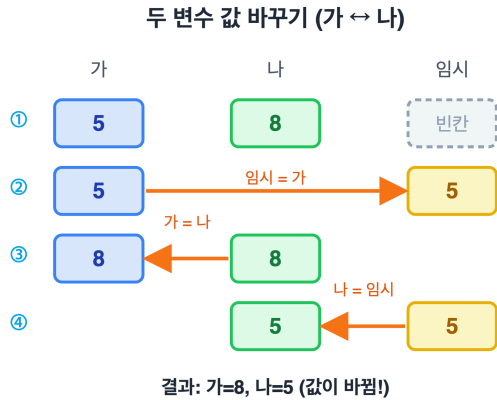
💡 1+2+3+4=10 처럼 1부터 차례로 더한 수를 '삼각수'라고 해. 점을 삼각형으로 쌓은 개수와 같아!

Q36 변수와 계산

변수(값을 담는 상자)를 서로 바꾸는 코드야.

가 = 5  
나 = 8  
임시 = 가  
가 = 나  
나 = 임시

모두 실행한 뒤 '가'와 '나'의 값은?



- ① ① 가=5, 나=8
- ② ② 가=8, 나=5
- ③ ③ 가=8, 나=8
- ④ ④ 가=5, 나=5

🎯 정답: ② 가=8, 나=5

📖 핵심: 두 상자를 바로 바꾸면 값이 덮어써져 사라지므로 '임시' 상자에 잠깐 보관한다. 임시=가(=5) → 가=나(=8) → 나=임시(=5). 그래서 가=8, 나=5로 깔끔하게 교환된다.

💡 이렇게 두 값을 바꾸는 걸 프로그래밍에서 'swap(스왑)'이라고 불러. 게임에서 아이템 두 칸을 맞바꿀 때도 똑같은 방법을 써!

**Q37** 조건 분기 (여러 갈래)

주차 요금 규칙이야.

만약 시간  $\leq 30$ 분 이면 요금 = 0원


아니면 만약 시간  $\leq 60$ 분 이면 요금 = 1000원

아니면 요금 = 2000원

45분 주차했을 때 요금은?

- ① ① 0원
- ② ② 1000원
- ③ ③ 2000원
- ④ ④ 3000원

 **정답: ② 1000원**

 **핵심:** 조건은 위에서부터 차례로 확인한다. 45분은 '30분 이하'가 아니므로 다음으로 간다. '60분 이하'에 해당( $45 \leq 60$ )하므로 1000원에서 멈춘다.

 이렇게 여러 갈래로 나뉘는 조건을 '아니면-만약(else if)'이라고 해. 위 조건이 먼저 걸리면 아래는 아예 확인하지 않아서 빠르지!

Q38 중첩 반복

별로 계단을 그리는 코드야.

반복 i = 1부터 4까지 [

반복 j = 1부터 i까지 [

별 출력

]

줄바꿈

]

출력되는 별(★)의 총 개수는?

별 계단 만들기 (반복)

★ 1줄: 1개

★★ 2줄: 2개

★★★ 3줄: 3개

★★★★ 4줄: 4개

$$\begin{array}{r}
 1+2+3+4 \\
 \hline
 = 10
 \end{array}$$

별 모두 10개 ★

- ① ① 4
- ② ② 6
- ③ ③ 10
- ④ ④ 16

🎯 정답: ③ 10

📖 핵심: 안쪽 반복은 바깥 i에 따라 횟수가 바뀐다. i=1일 때 별 1개, i=2일 때 2개, i=3일 때 3개, i=4일 때 4개. 합치면 1+2+3+4=10개.

💡 안쪽 반복 횟수가 바깥 변수에 맞춰 자라면 삼각형·계단 모양이 만들어져. 모니터의 픽셀 그림도 이런 중첩 반복으로 그린단다!

**Q39** 버그 찾기

1부터 5까지 모두 더해 15를 만들려는 코드인데, 결과가 10만 나와.

```
합 = 0
```


```
반복 i = 1부터 4까지 [
```


```
합 = 합 + i
```


```
]
```

버그를 고치려면 어떻게 해야 할까?

- ① ① 반복을  $i = 1$ 부터 5까지 로 바꾼다
- ② ② 합 = 1 로 시작한다
- ③ ③ 합 = 합 -  $i$  로 바꾼다
- ④ ④ 반복을 아예 없앤다

 **정답: ① 반복을  $i = 1$ 부터 5까지 로 바꾼다**

 **핵심:** 반복 끝값이 4라서 5를 한 번도 더하지 않았다(이를 '한끝 차이, off-by-one'이라 한다).  $1+2+3+4=10$ . 끝값을 5로 고치면 마지막에 +5가 되어 15가 된다.

 '한끝 차이' 버그는 프로그래머가 가장 자주 만드는 실수 중 하나야. 반복의 시작과 끝을 항상 두 번 확인하는 습관이 중요해!

**Q40** 변수와 계산

온도 변수를 차례로 바꾸는 코드야.

온도 = 20

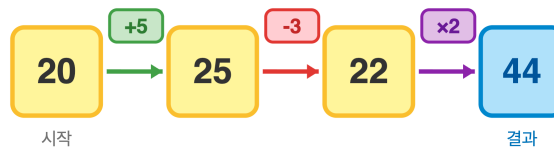
온도 = 온도 + 5

온도 = 온도 - 3

온도 = 온도 \* 2

마지막 '온도' 값은?

**변수 상자: 온도 변화**



온도 = 20 → 25 → 22 → 44

- ① ① 22
- ② ② 44
- ③ ③ 24
- ④ ④ 40

**정답: ② 44**

**핵심:** '온도 = 온도 ...'는 지금 값으로 계산해 다시 같은 상자에 넣는다는 뜻(갱신). 20 → +5는 25 → -3은 22 → x2는 44. 위에서 아래로 순서대로 따라가면 된다.

**💡** '온도 = 온도 + 5'는 수학의 등식이 아니라 '새 값을 다시 담아라'는 명령이야. 그래서 같은 상자 이름이 양쪽에 나와도 괜찮아!

## 초6 코딩

총 40문제 · 문제와 정답·풀이 포함

### Q41 알고리즘 입문

목록 [4, 7, 1, 9, 3]에서 앞에서부터 한 칸씩 보며 9를 찾는다(선형 탐색). 9를 찾을 때까지 숫자를 몇 번 비교할까?

#### 선형 탐색: 9 찾기



4번째 비교에서 9 찾음!

- ① ① 1
- ② ② 3
- ③ ③ 4
- ④ ④ 5

정답: ③ 4

핵심: 선형 탐색은 맨 앞부터 하나씩 확인한다. 4(아님)→7(아님)→1(아님)→9(찾음!). 네 번째에서 찾았으니 비교는 4번.

선형 탐색은 가장 단순하지만 목록이 길면 느려. 그래서 미리 순서대로 정렬해 두면 반씩 줄여 찾는 더 빠른 방법을 쓸 수 있어!

### Q42 의사코드 읽기

반복 i = 1부터 2까지 [

반복 j = 1부터 2까지 [

출력 (i, j)

]

]

출력되는 순서에서 '세 번째'로 나오는 것은?

- ① ① (1,2)
- ② ② (2,1)
- ③ ③ (1,1)
- ④ ④ (2,2)

정답: ② (2,1)

핵심: 안쪽 반복(j)이 먼저 다 돈 뒤 바깥(i)이 한 칸 넘어간다. i=1일 때 (1,1),(1,2) → i=2일 때 (2,1),(2,2). 세 번째는 (2,1).

중첩 반복은 자동차 계기판 숫자처럼 안쪽 자리가 먼저 끝까지 돌고 나서 바깥 자리가 하나 올라가는 것과 똑같아!

Q43 나만의 블록 / 함수 입문

값을 돌려주는(반환) 나만의 블록을 만들었어.

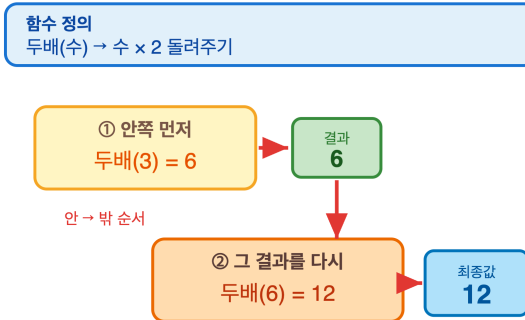
정의 함수 두배(수):

돌려주기 수 \* 2

출력 두배( 두배(3) )

무엇이 출력될까?

함수 호출: 두배(두배(3))



- ① ① 6
- ② ② 9
- ③ ③ 12
- ④ ④ 3

🎯 정답: ③ 12

📖 핵심: 안쪽 괄호부터 계산한다. 두배(3)은  $3 \times 2 = 6$ 을 돌려준다. 그 6이 다시 두배에 들어가  $6 \times 2 = 12$ . 함수가 돌려준 값을 또 다른 함수에 바로 넣을 수 있다.

💡 이렇게 함수 결과를 다시 함수에 넣는 걸 '함성'이라고 해. 한 번 잘 만든 블록은 몇 번이든 다시 써먹을 수 있어서 편리하지!

**Q44** 중첩 반복

반복이 세 겹으로 쌓인 코드야.

반복 i = 1부터 3까지 [

반복 j = 1부터 4까지 [

반복 k = 1부터 2까지 [

동작 실행

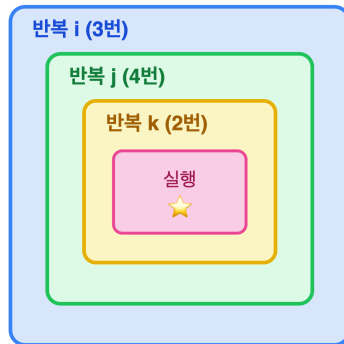
]

]

]

'동작 실행'은 모두 몇 번 일어날까?

세 겹 반복



$$\begin{array}{r} 3 \\ \times 4 \\ \times 2 \\ \hline 24 \end{array}$$

- ① ① 18
- ② ② 24
- ③ ③ 9
- ④ ④ 14

**정답: ② 24**

**핵심:** 중첩 반복의 총 실행 횟수는 각 반복 횟수의 곱이다. 바깥 3번마다 가운데 4번, 그 안마다 2번이 들므로  $3 \times 4 \times 2 = 24$ 번.

**💡** 반복을 한 겹 더 넣을 때마다 횟수가 곱해져서 금방 커져. 그래서 안쪽 반복에 무거운 동작을 넣으면 프로그램이 느려질 수 있어!

**Q45** 조건 분기 (여러 갈래)

가위바위보 승리 판정 규칙이야('또는'은 셋 중 하나만 맞아도 참).

만약 (나=가위 그리고 상대=보)

또는 (나=바위 그리고 상대=가위)

또는 (나=보 그리고 상대=바위)

이면 "승"

나=바위, 상대=가위 일 때 결과는?

- ① ① 승
- ② ② 패
- ③ ③ 비김
- ④ ④ 알 수 없음

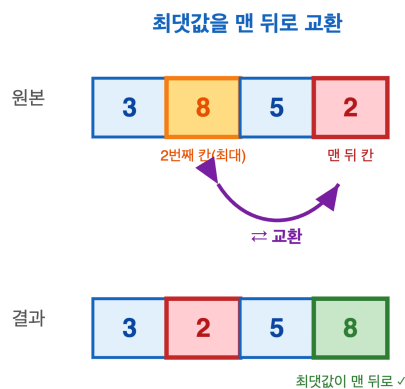
**정답: ① 승**

**핵심:** '그리고'는 둘 다 맞아야 참, '또는'은 하나만 맞아도 참이다. (나=바위 그리고 상대=가위)가 정확히 맞으므로 그 묶음이 참 → 전체가 참 → "승".

**💡** 바위는 가위를 이긴다는 규칙을 컴퓨터는 '그리고/또는' 논리로 표현해. 게임 속 승패 판정도 이런 논리식 한 줄로 깔끔하게 정해진다!

**Q46** 알고리즘 입문

목록 [3, 8, 5, 2]에서 '가장 큰 수를 찾아 맨 뒤 칸과 자리를 바꾸기'를 한 번 실행했어. 실행 후 목록은?



- ① ① [3, 2, 5, 8]
- ② ② [8, 3, 5, 2]
- ③ ③ [2, 3, 5, 8]
- ④ ④ [3, 8, 5, 2]

**정답: ① [3, 2, 5, 8]**

**핵심:** 먼저 가장 큰 수를 찾는다(8). 그 8의 자리와 맨 뒤 칸(2)을 맞바꾼다. 8과 2만 자리를 바꾸므로 [3, 2, 5, 8]. 나머지는 그대로다.

**💡** 가장 큰 수를 뒤로 보내는 걸 반복하면 목록 전체가 정렬돼. 이것이 '선택 정렬'의 기본 아이디어야!

**Q47** 재미 코딩

글자를 숫자로 바꾸는 암호 규칙이야(가=1, 나=2, 다=3, 라=4).

암호 만들기: 글자번호  $\times 2 + 1$

어떤 글자를 암호로 바꿨더니 7이 나왔어. 원래 글자는?

- ① ① 가
- ② ② 나
- ③ ③ 다
- ④ ④ 라

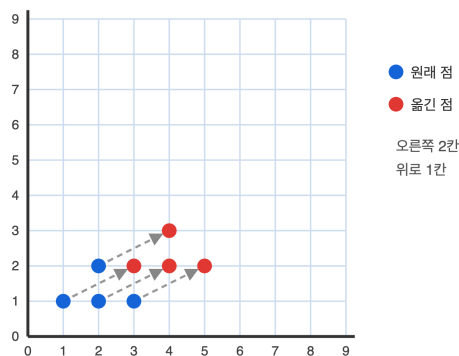
**정답: ③ 다**

**핵심:** 암호를 풀려면 만든 순서를 거꾸로 되돌린다. 만들 때  $\times 2$  후  $+1$  했으니, 풀 때는  $-1$  먼저 그 다음  $\div 2$ .  $7-1=6$ ,  $6\div 2=3$  → 번호 3 → '다'.

**💡** 암호를 거꾸로 풀 때는 연산 순서도 반대로 뒤집어야 해. 진짜 암호 기술도 '만드는 규칙'과 '되돌리는 규칙'이 짝을 이룬단다!

**Q48** 좌표·픽셀아트

격자(좌하 원점에 그린 모양의 네 점 (1,1), (2,1), (3,1), (2,2))를 모두 "오른쪽으로 2칸, 위로 1칸" 평행이동했어. 이동한 뒤 네 점의 새 좌표는?



- ① ① (3,2), (4,2), (5,2), (4,3)
- ② ② (3,2), (4,2), (5,2), (4,2)
- ③ ③ (2,2), (3,2), (4,2), (3,3)
- ④ ④ (3,1), (4,1), (5,1), (4,2)

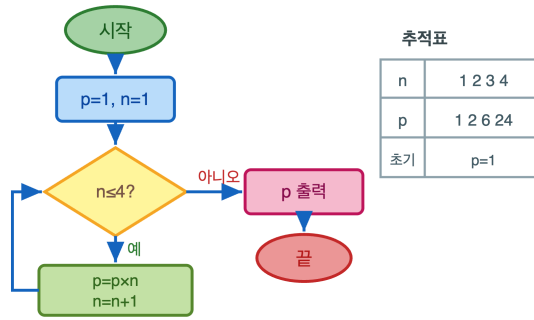
**정답: ① (3,2), (4,2), (5,2), (4,3)**

**핵심:** 평행이동은 모든 점의 x에 2, y에 1을 더하기. (1,1)→(3,2), (2,1)→(4,2), (3,1)→(5,2), (2,2)→(4,3). 한 점도 빠짐없이 같은 값을 더하는 게 핵심.

**💡** 게임에서 캐릭터가 움직일 때도 위치 좌표에 똑같은 값을 더하는 평행이동을 써.

**Q49** 순서도와 흐름

아래 순서도를 따라가면 출력되는 p 값은? 시작 → p=1, n=1 → (마름모) n<=4? → 예: p=p×n 그리고 n=n+1 하고 마름모로 되 돌아감 / 아니오: p 출력하고 끝.



- ① ① 12
- ② ② 24
- ③ ③ 10
- ④ ④ 120

**정답: ② 24**

한 칸씩 추적: n=1 p=1×1=1 → n=2 p=1×2=2 → n=3 p=2×3=6 → n=4 p=6×4=24 → n=5. n=5는 4보다 커서 멈추고 24 출력. (1×2×3×4)

이렇게 1부터 차례로 곱하는 값을 수학에서는 '팩토리얼'이라고 불러.

**Q50** 버그 찾기

1부터 5까지 더해서 합을 출력하려는 의사코드인데 오류가 나. 무엇을 고쳐야 할까?

반복 i=1..5 [ 합 = 합 + i ]

합 출력

- ① ① 반복 앞에 '합 = 0' 을 추가한다
- ② ② i를 2부터 시작하게 바꾼다
- ③ ③ 반복을 6까지로 늘린다
- ④ ④ '합 출력'을 반복 안으로 넣는다

**정답: ① 반복 앞에 '합 = 0' 을 추가한다**

'합 = 합 + i'를 처음 실행하려면 합에 시작값이 있어야 해. 시작값(초기값)을 정하지 않아서 오류가 나. 반복 앞에 '합 = 0'을 넣으면 0+1+2+3+4+5=15가 나와.

값을 차곡차곡 모으는 변수는 시작값을 0으로, 곱셈으로 모으는 변수는 1로 두는 게 보통이야.

**Q51** 중첩 반복

아래 중첩 반복을 실행하면 별 ★ 은 모두 몇 개 찍힐까?

```
반복 i=1..4 [  
반복 j=1..i [ ★ 찍기 ]  
]
```

이중 반복: 별 삼각형

```
for i = 1..4:  
  for j = 1..i:  
    print ★  
  새 줄
```

바깥: i = 줄 번호  
안쪽: 별 개수 = i

i	별 개수
i=1	1
i=2	2
i=3	3
i=4	4

1, 2, 3, 4 = 합 10개

- ① ① 4
- ② ② 10
- ③ ③ 16
- ④ ④ 8

**정답: ② 10**

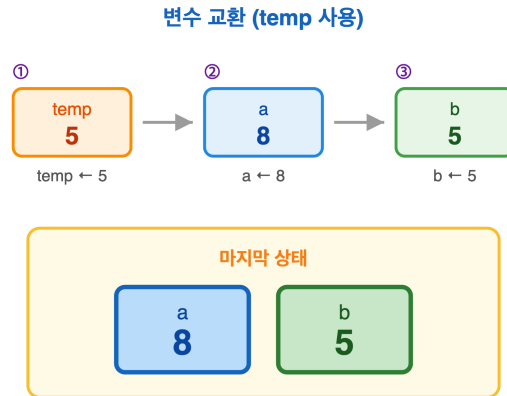
안쪽 반복은 i와 같은 횟수만큼 돌아. i=1일 때 1개, i=2일 때 2개, i=3일 때 3개, i=4일 때 4개. 모두 더하면 1+2+3+4=10개.

이렇게 한 칸씩 늘어나며 쌓이는 수를 '삼각수'라고 해. 1, 3, 6, 10, 15...

**Q52** 변수와 계산

상자 a에는 5, 상자 b에는 8이 들어 있어. 아래 순서대로 실행하면 끝났을 때 a와 b의 값은?

```
temp = a
a = b
b = temp
```



- ① ① a=8, b=5
- ② ② a=5, b=8
- ③ ③ a=8, b=8
- ④ ④ a=5, b=5

**정답: ① a=8, b=5**

temp에 a(5)를 잠시 보관 → a에 b(8)를 넣음 → b에 temp(5)를 넣음. 결과는 두 값이 서로 바뀐 a=8, b=5. 임시 상자 temp가 없으면 한쪽 값을 덮어써서 잃어버려.

두 값을 맞바꾸는 걸 '스왑(swap)'이라 하고, 컵 두 개의 물을 바꿀 때 빈 컵 하나가 필요한 것과 같아.

**Q53** 조건 분기 (여러 갈래)

점수로 등급을 매기는 의사코드야. 점수가 80일 때 출력되는 등급은?

```
만약 점수 >= 90 이면 "A"
아니면 만약 점수 >= 80 이면 "B"
아니면 만약 점수 >= 70 이면 "C"
아니면 "D"
```

- ① ① A
- ② ② B
- ③ ③ C
- ④ ④ D

**정답: ② B**

위에서부터 검사: 80 >= 90? 아니오 → 80 >= 80? 예! 여기서 "B"를 정하고 멈춰. 아래 조건은 더 보지 않아. '아니면 만약'은 앞이 거짓일 때만 검사한다는 게 핵심.

이런 갈래 검사는 위에서부터 차례로 보고 처음 맞는 곳에서 멈춰. 그래서 조건 순서가 중요해.

Q54 나만의 블록 / 함수 입문

입력 두 개를 받아 값을 돌려주는 함수를 정의했어.

함수 넓이(가로, 세로):  
돌려주기 가로 × 세로

그러면 '넓이(3, 4) + 넓이(2, 5)' 의 값은?

함수: 넓이(가로, 세로)

함수 넓이(가로, 세로):  
돌려주기 가로 × 세로

넓이(3, 4)  
→ 3 × 4 = 12

넓이(2, 5)  
→ 2 × 5 = 10

$$12 + 10 = 22$$

$$12 + 10 = 22$$

- ① ① 22
- ② ② 20
- ③ ③ 24
- ④ ④ 17

정답: ① 22

함수가 값을 '돌려주기' 때문에 호출한 자리에 결과 숫자가 들어가. 넓이(3,4)=12, 넓이(2,5)=10. 그래서 12+10=22.

값을 돌려주는 함수는 계산기 버튼 같아서, 부른 자리에 답이 쏙 들어가 다른 식에 바로 쓸 수 있어.

Q55 의사코드 읽기

목록 [3, 8, 5, 2, 7, 4] 에서 홀수의 개수를 세는 의사코드야. 출력값은?

개수 = 0

목록의 각 수 x 에 대해 반복 [

만약 (x 를 2로 나눈 나머지)=1 이면 개수 = 개수 + 1

]

개수 출력

- ① ① 2
- ② ② 3
- ③ ③ 4
- ④ ④ 6

정답: ② 3

각 수를 2로 나눈 나머지가 1이면 홀수. 3(홀)→1, 8(짝), 5(홀)→2, 2(짝), 7(홀)→3, 4(짝). 홀수는 3, 5, 7로 모두 3개.

나머지를 구하는 연산(mod)은 짝·홀 판정, 시계(12시간), 요일 계산처럼 '돌고 도는' 규칙에 정말 많이 쓰여.

**Q56** 알고리즘 입문

정렬된 목록 [2, 5, 8, 11, 15, 19, 23] 에서 '이진탐색'으로 15를 찾으려고 해. 이진탐색은 가운데 수와 비교해 절반씩 줄여가는 방법이야. 15를 찾기까지 비교는 모두 몇 번 할까?

**이진 탐색: 15 찾기**



- ① ① 1번
- ② ② 2번
- ③ ③ 3번
- ④ ④ 4번

**정답: ③ 3번**

① 가운데 11과 비교: 15는 더 크니 오른쪽 [15,19,23]만 본다. ② 가운데 19와 비교: 15는 더 작으니 왼쪽 [15]만 본다. ③ 가운데 15와 비교: 같다, 찾음! 모두 3번 비교.

이진탐색은 한 번 비교할 때마다 범위를 반으로 줄여서, 1000개 중에서도 10번이면 찾아내. 사전을 펼쳐 찾는 방법과 똑같아.

**Q57** 재미 코딩

이진수는 0과 1만으로 수를 나타내고, 자리값은 오른쪽부터 1, 2, 4, 8 이야. 이진수 1011 을 십진수로 바꾸면?

- ① ① 11
- ② ② 13
- ③ ③ 9
- ④ ④ 8

**정답: ① 11**

각 자리값 × 그 자리 숫자를 더해. 자리값 8,4,2,1에 1,0,1,1을 맞추면  $1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 8 + 0 + 2 + 1 = 11$ .

컴퓨터는 전기가 켜짐(1)/꺼짐(0)만 알아서 모든 수와 글자, 그림을 이진수로 저장해.

**Q58** 재미 코딩

어떤 수를 2로 나눈 나머지가 0이면 짝수, 1이면 홀수야. 17을 2로 나눈 나머지는 얼마이고, 17은 짝수일까 홀수일까?

- ① ① 나머지 0, 짝수
- ② ② 나머지 1, 홀수
- ③ ③ 나머지 2, 짝수
- ④ ④ 나머지 1, 짝수

**정답: ② 나머지 1, 홀수**

$17 \div 2 = 8 \dots$  나머지 1. 나머지가 1이므로 17은 홀수. 나머지로 짝·홀을 가르는 건 컴퓨터가 가장 자주 쓰는 판정 중 하나야.

줄을 둘씩 짝지을 때 한 명이 남으면 홀수, 딱 맞으면 짝수인 것과 같은 원리야.

Q59 알고리즘 입문

목록 [4, 9, 2, 7, 5] 에서 왼쪽부터 차례로 하나씩 살펴보며 7을 찾는 '선형 탐색'을 해. 7은 몇 번째 칸에 있을까? (맨 앞을 1번째로 센다)

목록에서 7 찾기



- ① ① 2번째
- ② ② 3번째
- ③ ③ 4번째
- ④ ④ 5번째

🎯 정답: ③ 4번째

📖 맨 앞부터 비교: 1번째 4(아님), 2번째 9(아님), 3번째 2(아님), 4번째 7(같다!). 그래서 4번째 칸에서 찾아. 정렬이 안 된 목록은 이렇게 하나씩 보는 선형 탐색을 써.

💡 선형 탐색은 정렬이 안 되어 있어도 쓸 수 있지만, 운이 나쁘면 끝까지 다 봐야 해서 이진탐색보다 느릴 수 있어.

**Q60** 변수와 계산

세 변수에 시험 점수가 들어 있어요. 국어=80, 수학=90, 영어=70.

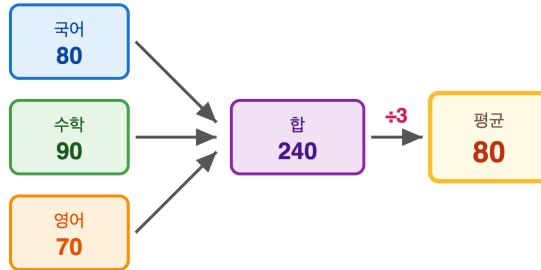
의사코드:

합 ← 국어 + 수학 + 영어

평균 ← 합 ÷ 3

반복이 끝난 뒤 '평균'에 들어가는 값은?

변수 → 평균 계산



$$\begin{aligned} \text{합} &= 80 + 90 + 70 = 240 \\ \text{평균} &= 240 \div 3 = 80 \end{aligned}$$

- ① ① 70
- ② ② 75
- ③ ③ 80
- ④ ④ 85

**정답: ③ 80**

합 = 80 + 90 + 70 = 240. 평균 = 240 ÷ 3 = 80. 변수에 값을 차례로 넣어 계산해요.

평균은 점수들을 고르게 뺐을 때의 값이라, 한 과목이 오르면 평균도 같이 올라가요.

**Q61** 조건 분기 (여러 갈래)

가위바위보 결과를 정하는 의사코드예요. 나='가위', 친구='보'.

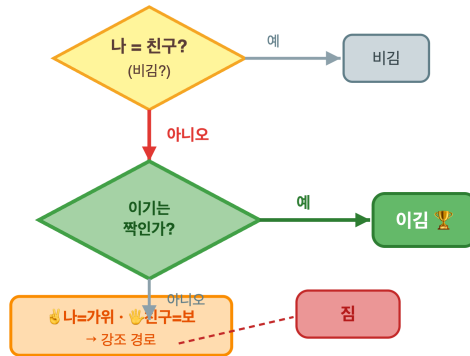
만약 (나 = 친구) 이면 → '비김'

아니면 만약 (나='가위' 그리고 친구='보') 또는 (나='바위' 그리고 친구='가위') 또는 (나='보' 그리고 친구='바위') 이면 → '이김'

아니면 → '짐'

출력되는 말은?

가위바위보 조건 분기



- ① ① 이김
- ② ② 짐
- ③ ③ 비김
- ④ ④ 알 수 없음

**정답: ① 이김**

먼저 나=친구? 가위≠보라 아니예요. 다음 조건에서 (나='가위' 그리고 친구='보')가 참이므로 '이김'. '또는'은 하나만 참이어도 전체가 참이에요.

가위바위보는 서로 물고 물리는 구조라, 항상 이기는 손이 하나도 없어요.

**Q62** 알고리즘 입문

목록 [3, 8, 5, 2, 9]에서 값 5를 앞에서부터 차례로 찾아요(선형 탐색).  
맨 앞을 1번째로 셀 때, 5는 '몇 번째 자리'에서 처음 만날까요?

선형 탐색: 값 '5' 찾기



- ① ① 1번째
- ② ② 2번째
- ③ ③ 3번째
- ④ ④ 4번째

**정답: ③ 3번째**

앞에서부터 비교해요. 1번째  $3 \neq 5$ , 2번째  $8 \neq 5$ , 3번째  $5 = 5$  → 찾음! 선형 탐색은 한 칸씩 차례로 확인해요.

목록이 정렬되어 있지 않을 때는 이렇게 하나씩 보는 선형 탐색이 가장 기본 방법이에요.

**Q63** 재미 코딩

사물함 번호가 홀수면 '왼쪽 줄', 짝수면 '오른쪽 줄'에 있어요.

판정 규칙: 번호 ÷ 2의 나머지가 1이면 홀수, 0이면 짝수.

번호 14는 어느 줄에 있을까요?

- ① ① 왼쪽 줄
- ② ② 오른쪽 줄
- ③ ③ 가운데 줄
- ④ ④ 알 수 없음

**정답: ② 오른쪽 줄**

$14 \div 2 = 7$ , 나머지 0 → 짝수 → 오른쪽 줄. '나머지(mod)'는 나누고 남은 수예요.

컴퓨터는 '나머지가 0인가?'를 가지고 짝수·홀수, 배수 판정을 아주 자주 해요.

**Q64** 중첩 반복

중첩 반복으로 별을 찍어요.

반복  $i = 1$ 부터 3까지 [

반복  $j = 1$ 부터  $i$ 까지 [ ★ 찍기 ]

줄 바꾸기

]

화면에 별은 모두 몇 개 찍힐까요?

**반복 중첩 (Nested Loop)**



- ① ① 3개
- ② ② 5개
- ③ ③ 6개
- ④ ④ 9개

**정답: ③ 6개**

바깥  $i$ 가 1,2,3일 때 안쪽이 각각 1,2,3번 도므로  $1 + 2 + 3 = 6$ 개. 별이 삼각형 모양으로 늘어나요.

이렇게 1부터 차례로 더한 수(1,3,6,10...)를 '삼각수'라고 불러요.

Q65 나만의 블록 / 함수 입문

값을 돌려주는 '함수'를 만들었어요.

함수 넓이(가로, 세로):

돌려주기 → 가로 × 세로

호출: 넓이(4, 3)

이 호출의 결과 값은?

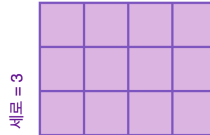
함수 넓이(가로, 세로)

정의 함수 넓이(가로, 세로):  
돌려주기 가로 × 세로

호출: 넓이(4, 3)



가로 = 4



결과  
12

$$4 \times 3 = 12 \text{ (칸 수)}$$

■ 한 칸 × 12 = 넓이 12

- ① ① 7
- ② ② 12
- ③ ③ 14
- ④ ④ 24

🎯 정답: ② 12

📖 가로=4, 세로=3을 넣으면  $4 \times 3 = 12$ 를 돌려줘요. 함수는 입력을 넣으면 결과를 만들어 주는 '나만의 블록'이에요.

💡 한 번 만든 함수는 어떤 직사각형이든 가로·세로만 넣으면 넓이를 바로 구해 줘요.

Q66 의사코드 읽기

의사코드를 따라가요.

곱 ← 1

반복 k = 1부터 4까지 [ 곱 ← 곱 × k ]

반복이 끝났을 때 '곱'의 값은?

변수 추적: 곱

변수 상자  
곱 = 24

k	곱
1	1
2	2
3	6
4	24

- ① ① 10
- ② ② 16
- ③ ③ 24
- ④ ④ 64

🎯 정답: ③ 24

📖 곱을 1로 시작해서  $1 \times 1 = 1$ ,  $1 \times 2 = 2$ ,  $2 \times 3 = 6$ ,  $6 \times 4 = 24$ . 더하기 대신 곱하기를 쌓는 '곱 누적'이에요.

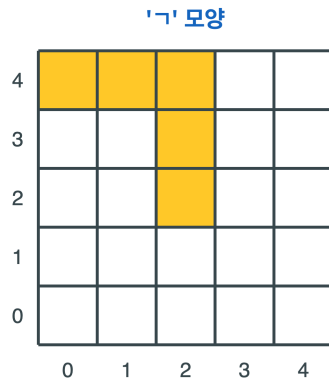
💡 1부터 n까지 모두 곱한 값을 '팩토리얼'이라 하고, 줄을 세우는 경우의 수를 셀 때 쓰여요.

**Q67** 좌표·픽셀아트

5×5 격자(좌하 원점, 가로·세로 0~4)에 아래 좌표를 칠해요.

(0,4) (1,4) (2,4) (2,3) (2,2)

어떤 글자 모양이 만들어질까요?



- ① ① ㄱ
- ② ② ㄴ
- ③ ③ ㄷ
- ④ ④ ㅁ

**정답:** ① ㄱ

맨 위 가로줄 (0,4)-(1,4)-(2,4)와, 오른쪽 끝에서 아래로 내려오는 (2,3)-(2,2)가 합쳐져 'ㄱ' 모양이 돼요.

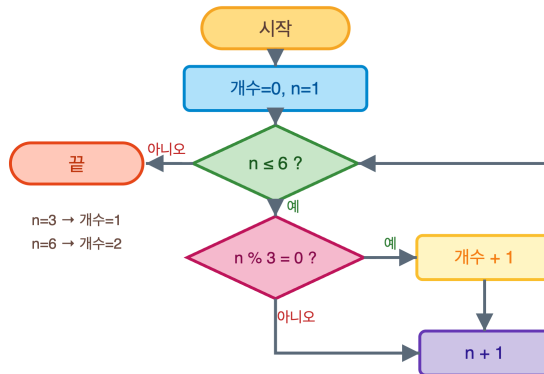
화면의 그림은 모두 이렇게 좌표마다 점(픽셀)을 칠해 만든 거예요.

**Q68** 순서도와 흐름

순서도로 1부터 6까지 보면서 3의 배수만 세요.

시작 → [개수=0, n=1] → (n ≤ 6?) 예 → (n ÷ 3의 나머지 = 0?) 예 → [개수=개수+1] 아니오 → 그냥 → [n=n+1] → 되돌아감 / (n ≤ 6?) 아니오 → 끝.

끝났을 때 '개수'는?



- ① ① 1
- ② ② 2
- ③ ③ 3
- ④ ④ 6

**정답: ② 2**

**📖** n=1~6 중 3으로 나눈 나머지가 0인 수는 3과 6, 두 개예요. 조건이 맞을 때만 개수를 1 늘려서 개수=2.

**💡** '나머지=0인가?'만 바꾸면 2의 배수, 5의 배수도 같은 순서도로 셀 수 있어요.

**Q69** 버그 찾기

1부터 5까지 더하려는 코드인데 멈추지 않고 영원히 돌아요(무한 반복).

합 ← 0

n ← 1

반복 (n ≤ 5 인 동안) [

합 ← 합 + n

]

무엇을 빠뜨려서 생긴 버그일까요?

- ① ① 합의 초기값을 0으로 둔 것
- ② ② 반복 안에서 n ← n + 1 을 빠뜨린 것
- ③ ③ n을 1로 시작한 것
- ④ ④ 조건을 n ≤ 5 로 한 것

**정답: ② 반복 안에서 n ← n + 1 을 빠뜨린 것**

**📖** n이 계속 1이라 'n ≤ 5'가 항상 참 → 영원히 반복해요. 반복 안에 n ← n + 1 을 넣어 n을 키워야 5를 넘겨 멈춰요.

**💡** 멈추는 조건이 절대 거짓이 안 되면 '무한 루프'에 빠져요. 반복 변수의 갱신을 꼭 확인하세요!

**Q70** 변수와 계산

변수 세 개 A=1, B=2, C=3. 아래 순서대로 실행해요.

임시 ← A

A ← B

B ← C

C ← 임시

끝난 뒤 (A, B, C)는?

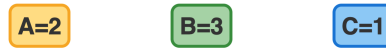
변수 상자 옮기기



명령 순서

- ① 임시 ← A
- ② A ← B
- ③ B ← C
- ④ C ← 임시

끝 상태



- ① ① (1, 2, 3)
- ② ② (2, 3, 1)
- ③ ③ (3, 1, 2)
- ④ ④ (2, 1, 3)

**정답: ② (2, 3, 1)**

임시=1로 A를 잠깐 보관. A=B=2, B=C=3, C=임시=1. 값이 한 칸씩 '돌아가는' 회전이에요. 임시 변수 하나로 세 값도 돌릴 수 있어요.

임시 변수가 없으면 첫 줄에서 A 값이 덮여 사라져, 회전이 망가지요.

**Q71** 나만의 블록 / 함수 입문

별을 정해진 수만큼 찍는 함수를 만들었어요.

함수 별찍기(개수):

반복 개수 번 [ ★ 찍기 ]

호출 순서: 별찍기(2), 별찍기(3)

화면에 찍힌 별은 모두 몇 개일까요?



- ① ① 3개
- ② ② 5개
- ③ ③ 6개
- ④ ④ 9개

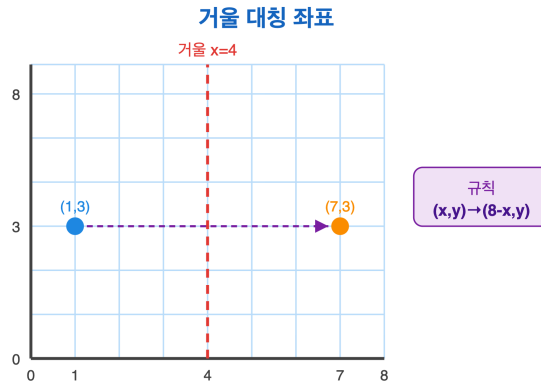
**정답: ② 5개**

별찍기(2)는 ★★ 2개, 별찍기(3)은 ★★★ 3개. 합쳐서 5개. 같은 함수라도 입력(개수)을 바꾸면 다르게 동작해요.

입력값(매개변수)을 받는 함수 하나면, 별 100개도 별찍기(100) 한 줄로 끝나요.

**Q72** 좌표·픽셀아트

픽셀아트 모양을 세로선  $x=4$ 를 거울로 삼아 좌우로 뒤집는 규칙은  $(x, y) \rightarrow (8 - x, y)$  야. 점  $(1, 3)$ 에 이 규칙을 쓰면 어디로 갈까?



- ① ①  $(7, 3)$
- ② ②  $(1, 5)$
- ③ ③  $(7, 5)$
- ④ ④  $(3, 3)$

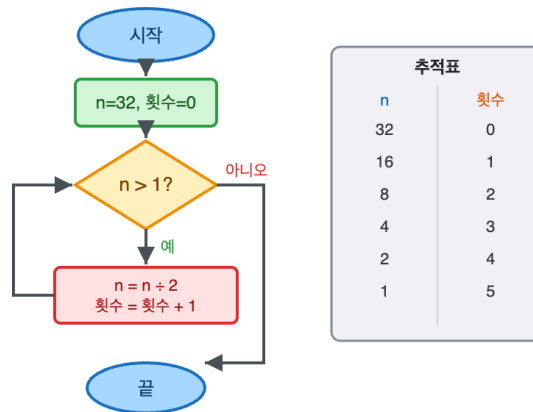
**정답: ①  $(7, 3)$**

규칙  $(x, y) \rightarrow (8 - x, y)$ 에  $x=1, y=3$ 을 넣어. 새  $x = 8 - 1 = 7$ , 새  $y$ 는 그대로 3. 그래서  $(7, 3)$ .  $y$ 는 안 바뀌고  $x$ 만 거울 반대편으로 가니, 모양이 좌우로 뒤집혀.

그림 프로그램의 '좌우 반전' 단추가 바로 이 규칙이야. 높이는 그대로 두고 가로 위치만 거울처럼 바꿔.

**Q73** 순서도와 흐름

순서도를 따라가 보.  $n=32$ , 횃수=0으로 시작  $\rightarrow (n > 1?)$  예이면 ' $n = n \div 2$ , 횃수 = 횃수 + 1' 후 다시 조건으로  $\rightarrow$  아니오이면 끝. 끝났을 때 횃수는 얼마일까?



- ① ① 4번
- ② ② 5번
- ③ ③ 6번
- ④ ④ 32번

**정답: ② 5번**

$n$ 이 1보다 큰 동안 반으로 줄이며 횃수를 1씩 늘려.  $32 \rightarrow 16(1) \rightarrow 8(2) \rightarrow 4(3) \rightarrow 2(4) \rightarrow 1(5)$ . 이제  $n=1$ 이라  $1 > 1$ 이 거짓이 되어 멈춰. 횃수는 5야.

수를 반씩 줄여 1까지 가는 횃수가 바로 '2를 몇 번 곱했나(밑 2 로그)'와 같아.  $32 = 2 \times 2 \times 2 \times 2 \times 2$  라 5번이야.

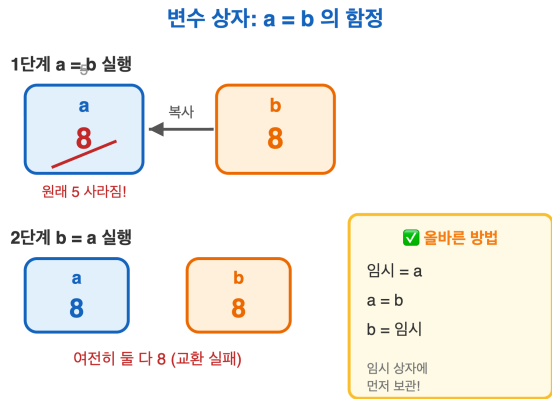
**Q74** 버그 찾기

변수  $a=5$ ,  $b=8$  의 값을 서로 바꾸려고 이렇게 썼어.

$a = b$

$b = a$

그런데 실행하면  $a=8$ ,  $b=8$  이 되어 버려. 올바르게 바꾸려면 어떻게 고쳐야 할까?



- ① ① 임시 변수에  $a$ 를 먼저 저장한 뒤  $a=b$ ,  $b$ =임시 순으로 바꾼다
- ② ② 순서를  $b=a$  먼저,  $a=b$  나중에 바꾼다
- ③ ③  $a=a+b$  를 한 번만 실행한다
- ④ ④ 반복을 한 번 더 추가한다

**정답:** ① 임시 변수에  $a$ 를 먼저 저장한 뒤  $a=b$ ,  $b$ =임시 순으로 바꾼다

' $a=b$ '를 하는 순간  $a$ 의 원래 값 5가 사라져서 나중에  $b$ 에 넣을 5가 없어서. 그래서 둘 다 8이 돼. 먼저 임시= $a$ 로 5를 안전하게 보관하고,  $a=b(=8)$ ,  $b$ =임시( $=5$ ) 순으로 하면 제대로 바뀌어.

값을 안전하게 보관하는 임시 변수는 '컵 두 개의 물을 바꿔 담을 때 쓰는 빈 컵'과 똑같은 역할이야.

**Q75** 중첩 반복

다음 블록을 실행해.

반복  $i=1..4$  [

반복  $j=1..i$  [

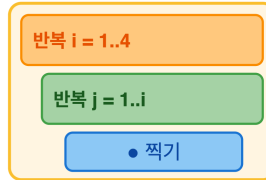
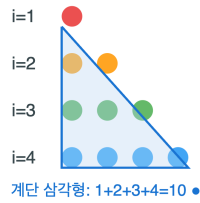
출력 •

]

]

출력되는 • 은 모두 몇 개일까?

**중첩 반복: 안쪽 j가 i만큼**



바깥 1번 들 때마다  
안쪽이 i번 반복

- ① ① 7개
- ② ② 10개
- ③ ③ 16개
- ④ ④ 4개

**정답: ② 10개**

📖 안쪽 반복이 바깥쪽  $i$  값만큼 돌아.  $i=1$ 일 때 1개,  $i=2$ 일 때 2개,  $i=3$ 일 때 3개,  $i=4$ 일 때 4개. 모두 더하면  $1+2+3+4 = 10$ 개야. 안쪽 횟수가 고정인 아니라 바깥 값을 따라 늘어나는 게 핵심이야.

💡  $1+2+3+\dots+n$  처럼 차례로 늘어나는 점 개수를 '삼각수'라고 불러. 볼링 핀이 1,2,3,4로 놓인 것도 삼각수야.

**Q76** 변수와 계산

변수  $x$ 가 있어. 다음을 실행하면 끝났을 때  $x$ 는 얼마일까?

$$x = 2$$

반복 4번 [

$$x = x + x$$

]

**변수  $x$  추적표 (매번 두 배)**



매번 자기 자신을 더해 두 배! → 32 도착 🎯

- ① ① 16
- ② ② 32
- ③ ③ 8
- ④ ④ 64

🎯 **정답: ② 32**

📖  $x = x + x$  는 자기 자신을 더하니 두 배가 돼. 시작 2에서 네 번 두 배: 2→4→8→16→32. 그래서  $x = 32$ . 반복할 때마다 변수가 어떻게 갱신되는지 한 줄씩 따라가는 게 중요해.

💡 두 배를 거듭하면 수가 폭발적으로 커져. 종이를 4번 접으면 두께가 16배가 되는 것과 같은 원리야.

**Q77** 의사코드 읽기

다음 의사코드를 따라가 보.

$a = 1$

$b = 1$

반복 4번 [

$c = a + b$

$a = b$

$b = c$

]

반복이 끝난 뒤  $b$ 를 출력하면 무엇이 나올까?

변수 추적 표 (피보나치)

반복	$c=a+b$	$a$	$b$
시작	-	1	1
1회	2	1	2
2회	3	2	3
3회	5	3	5
4회	8	5	<b><math>b=8</math></b>

수열: 1 1 2 3 5 8  
↑ 마지막  $b$

규칙:  $c = a + b$ ,  $a \leftarrow b$ (이전),  $b \leftarrow c$

- ① ① 5
- ② ② 8
- ③ ③ 13
- ④ ④ 7

**정답: ② 8**

앞 두 수를 더해 다음 수를 만드는 규칙이야.  $c=a+b$ 로 새 수를 구하고,  $a$ 에는 옛  $b$ 를,  $b$ 에는 새 수  $c$ 를 넣어.  $1, 1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 8$ 로 네 번 만들면  $b=8$ . 순서대로 한 줄씩 값을 옮겨 적는 게 비결이야.

이렇게 앞 두 수를 더해 이어지는 수열을 '피보나치 수열'이라고 해. 꽃잎이나 솔방울 무늬에서도 보여.

**Q78** 나만의 블록 / 함수 입문

함수 두 개를 정의했어.

함수 제곱(x): 돌려주기  $x \times x$

함수 제곱합(n):

합 = 0

반복  $k=1..n$  [

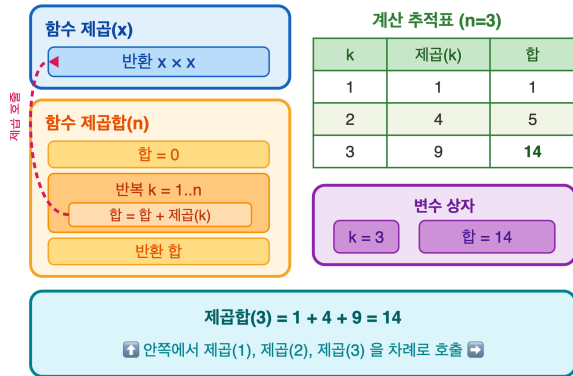
합 = 합 + 제곱(k)

]

돌려주기 합

그러면 제곱합(3)의 결과는 얼마일까?

**함수: 제곱과 제곱합**



- ① ① 9
- ② ② 14
- ③ ③ 36
- ④ ④ 6

**정답: ② 14**

제곱합 함수는 k를 1부터 3까지 바꾸며 제곱(k)를 더해 모아. k=1: 제곱(1)=1 → 합 1, k=2: 제곱(2)=4 → 합 5, k=3: 제곱(3)=9 → 합 14. 함수 안에서 다른 함수를 불러 쓸 수 있다는 게 핵심이야.

1<sup>2</sup>+2<sup>2</sup>+3<sup>2</sup>처럼 제곱을 차례로 더한 값을 '제곱수의 합'이라고 해. 큰 함수가 작은 함수를 부르면 복잡한 계산도 깔끔해져.

**Q79** 조건 분기 (여러 갈래)

놀이기구 탑승 규칙이야.

만약 (키 >= 120) 그리고 (어른과 함께) 이면 '탈 수 있음'

아니면 '탈 수 없음'

키가 125cm인데 혼자 왔다면(어른과 함께 = 아니오) 결과는 무엇일까?

- ① ① 탈 수 있음
- ② ② 탈 수 없음
- ③ ③ 알 수 없음
- ④ ④ 키만 보면 됨

**정답: ② 탈 수 없음**

'그리고(AND)'는 두 조건이 모두 참일 때만 전체가 참이 돼. 키 125 >= 120은 참이지만, '어른과 함께'가 거짓이라 둘을 묶으면 거짓이야. 그래서 '탈 수 없음'으로 결정돼.

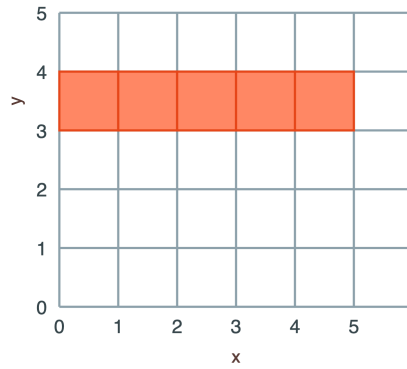
'그리고'는 둘 다 참이어야 통과, '또는'은 하나만 참이어도 통과야. 두 단어 차이로 결과가 완전히 달라져.

**Q80** 좌표·픽셀아트

규칙에 따라 격자의 칸을 칠해.

칠할 좌표: (0,3), (1,3), (2,3), (3,3), (4,3)

이렇게 칠하면 어떤 모양이 될까?



- ① ① 가로(수평)선
- ② ② 세로(수직)선
- ③ ③ 대각선
- ④ ④ 점 하나

**정답: ① 가로(수평)선**

**📖** 칠한 칸을 보면 y는 모두 3으로 같고 x만 0,1,2,3,4로 변해. 높이는 그대로인데 옆으로만 나란히 이어지니 가로(수평)선이 돼. 좌표에서 y가 같으면 가로줄, x가 같으면 세로줄이야.

**💡** 화면의 모든 그림은 이렇게 작은 칸(픽셀)을 좌표로 하나씩 칠해서 만들어져. 가로선 하나도 점 여러 개의 모임이야.

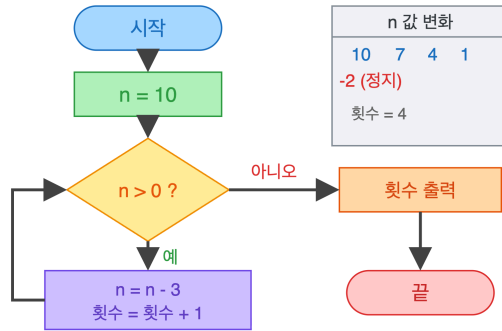
## 초6 코딩

총 20문제 · 문제와 정답·풀이 포함

### Q81 순서도와 흐름

순서도를 따라가 보자. '변수'는 값을 담는 상자야.

시작 →  $n=10$  → 마름모 ' $n>0?$ ' → (예)  $n=n-3$  하고 화살표가 마름모로 되돌아감 → (아니오) 끝, '뺀 횟수' 출력.  
 $n$ 에서 3을 몇 번 빼면 멈출까?



- ① ① 3번
- ② ② 4번
- ③ ③ 5번
- ④ ④ 6번

정답: ② 4번

핵심:  $n$ 이 0보다 큰 동안만 3을 뺀다.  $n$ 을 추적:  $10 \rightarrow 7(1\text{번}) \rightarrow 4(2\text{번}) \rightarrow 1(3\text{번}) \rightarrow -2(4\text{번})$ . -2는 0보다 크지 않으므로 멈춘다. 따라서 4번 뺐다.

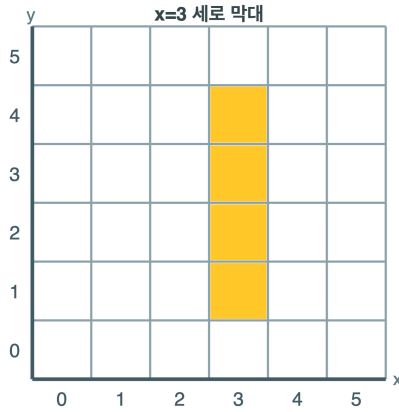
이렇게 '조건이 참인 동안 반복'하는 흐름을 컴퓨터에서는 '조건 반복'이라고 불러.

**Q82** 좌표·픽셀아트

격자에 다음 좌표 칸을 모두 칠했어. (가로,세로) 순서야.

(3,1), (3,2), (3,3), (3,4)

칠해진 모양은 무엇일까?



- ① ① 가로선 (옆으로 누운 선)
- ② ② 세로선 (위아래로 선 선)
- ③ ③ 대각선 (비스듬한 선)
- ④ ④ 점 하나

**정답: ② 세로선 (위아래로 선 선)**

**핵심:** 가로값 x는 모두 3으로 같고, 세로값 y만 1,2,3,4로 늘어난다. x가 고정이고 y만 변하면 위아래로 곧게 선 '세로선'이 된다.

**💡** x가 고정이면 세로선, y가 고정이면 가로선! 좌표 규칙만 봐도 모양을 짐작할 수 있어.

**Q83** 중첩 반복

안쪽 반복과 바깥쪽 반복이 겹친 '중첩 반복' 블록이야.

반복  $i=1..4$  [

반복  $j=1..i$  [ ☆ 찍기 ]

]

별(☆)은 모두 몇 개 찍힐까?



- ① ① 4개
- ② ② 7개
- ③ ③ 10개
- ④ ④ 16개

**정답: ③ 10개**

**핵심:** 안쪽 반복은  $i$ 번 돈다.  $i=1$ 일 때 1개,  $i=2$ 일 때 2개,  $i=3$ 일 때 3개,  $i=4$ 일 때 4개. 모두 더하면  $1+2+3+4=10$ 개.

**💡** 1부터 차례로 더한 1,3,6,10... 같은 수를 '삼각수'라고 해. 별 삼각형 개수가 딱 삼각수야!

**Q84** 버그 찾기

1부터 5까지 출력하려고 만든 의사코드인데 1,2,3,4까지만 나와. 버그를 어떻게 고칠까?

변수  $n = 1$

반복: 만약  $n < 5$  이면

$n$  출력

$n = n + 1$

- ① ①  $n = 1$  을  $n = 0$  으로 바꾼다
- ② ②  $n < 5$  를  $n \leq 5$  로 바꾼다
- ③ ③  $n = n + 1$  을  $n = n + 2$  로 바꾼다
- ④ ④  $n$  출력 줄을 지운다

**정답: ②  $n < 5$  를  $n \leq 5$  로 바꾼다**

**핵심:** ' $n < 5$ '는  $n$ 이 5가 되는 순간 거짓이 되어 5를 출력하기 전에 멈춘다(한끝 차이, off-by-one). '<='로 바꾸면  $n=5$ 일 때도 조건이 참이라 5까지 출력한다.

**💡** '<'와 '<='를 헷갈려 생기는 한끝 차이 버그는 프로그래머가 가장 자주 만나는 실수 중 하나야.

**Q85** 변수와 계산

임시 상자 없이 두 변수의 값을 바꾸는 마법 같은 의사코드야. 처음  $a=8$ ,  $b=5$  일 때, 모두 실행한 뒤  $a$ 와  $b$ 는?

$$a = a + b$$

$$b = a - b$$

$$a = a - b$$

변수 교환:  $a, b$



단계	a	b
시작	8	5
$a = a + b$ 후	13	5
$b = a - b$ 후	13	8
$a = a - b$ 후	5	8

- ① ①  $a=5$ ,  $b=8$
- ② ②  $a=8$ ,  $b=5$
- ③ ③  $a=13$ ,  $b=8$
- ④ ④  $a=5$ ,  $b=5$

**정답: ①  $a=5$ ,  $b=8$**

**핵심:** 더하고 빼면서 임시 변수 없이 값을 맞바꾼다. ① $a=8+5=13$ . ② $b=13-5=8$ . ③ $a=13-8=5$ . 최종  $a=5$ ,  $b=8$  으로 처음과 비교해 값이 서로 바뀌었다.

**💡** 이 '덧셈-뺄셈 스왑'은 임시 상자를 안 써서 메모리를 아끼지만, 요즘은 읽기 쉬운 임시 변수 방식을 더 많이 써.

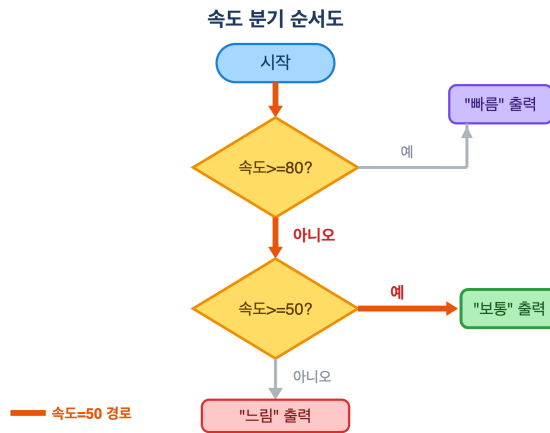
**Q86** 조건 분기 (여러 갈래)

바람의 속도에 따라 알림을 정하는 의사코드야. 속도 = 50 일 때 출력은?

만약 속도  $\geq 80$  이면 "빠름"

아니면 만약 속도  $\geq 50$  이면 "보통"

아니면 "느림"



- ① ① 빠름
- ② ② 보통
- ③ ③ 느림
- ④ ④ 아무것도 출력 안 됨

**정답: ② 보통**

**핵심:** 위에서부터 차례로 확인한다. 속도=50은 80 이상이 아니라 첫 갈래는 거짓. 다음 '50 이상?'은 참(50은 50과 같음)이라 '보통'을 출력하고 멈춘다.

**💡** '아니면 만약'은 위 조건이 거짓일 때만 확인해. 그래서 순서가 바뀌면 결과도 달라질 수 있어.

**Q87** 의사코드 읽기

의사코드를 따라가 보자. '합'에 짝수만 더한다.

변수 합 = 0

반복 i = 1 부터 6 까지 [

만약 i 가 짝수이면 합 = 합 + i

]

반복이 끝났을 때 '합'은?

- ① ① 9
- ② ② 12
- ③ ③ 15
- ④ ④ 21

**정답: ② 12**

**핵심:** 1~6 중 짝수만 골라 더한다. 짝수는 2, 4, 6. 합 = 2 + 4 + 6 = 12. (홀수 1,3,5는 조건이 거짓이라 건너뛴다.)

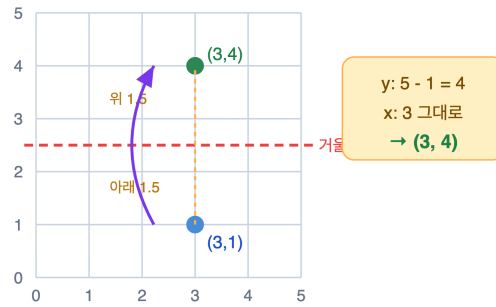
**💡** 짝수인지 보려면 2로 나눈 나머지가 0인지 확인하면 돼. 이것 'mod 연산'이라고 불러.



Q90 좌표·픽셀아트

세로 0~5 격자에서 점을 '상하 뒤집기' 하면 위아래가 거꾸로 돼. 규칙은  $(x, y) \rightarrow (x, 5 - y)$  야.  
점 (3, 1) 을 상하 뒤집기 하면 어디로 갈까?

상하 뒤집기  $(x,y) \rightarrow (x,5-y)$



- ① ① (3, 2)
- ② ② (3, 4)
- ③ ③ (1, 3)
- ④ ④ (2, 4)

🎯 정답: ② (3, 4)

📖 핵심: 가로값 x는 그대로, 세로값 y만 5-y로 바뀐다. x=3은 유지, y=1 → 5-1=4. 따라서 (3,1) → (3,4).

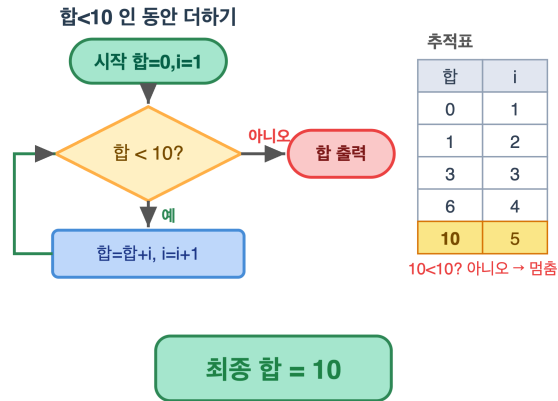
💡 x는 그대로 두고 y만 뒤집으면 상하 반사, 반대로 하면 좌우 반사야. 거울 방향이 달라지는 거지.

**Q91** 순서도와 흐름

순서도를 따라가 보자. '합'이 10보다 작은 동안만  $i$ 를 더해.

시작  $\rightarrow$  합=0,  $i=1 \rightarrow$  마음모 '합 < 10?'  $\rightarrow$  (예) 합=합+i,  $i=i+1$ , 마음모로 되돌아감  $\rightarrow$  (아니오) 끝, 합 출력.

최종 '합'은 얼마일까?



- ① ① 6
- ② ② 10
- ③ ③ 15
- ④ ④ 3

**정답: ② 10**

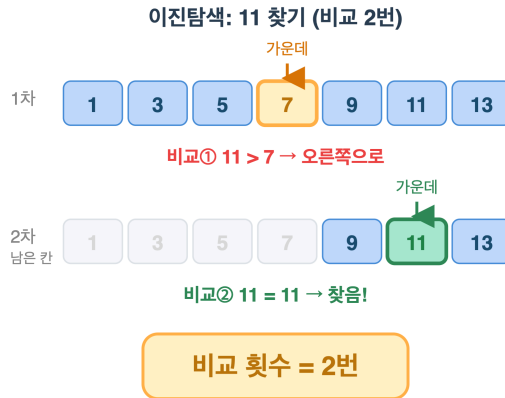
**핵심:** 합이 10보다 작은 동안만 더한다. (합,  $i$ )를 추적: 0, 1  $\rightarrow$  더해서 합=1,  $i=2 \rightarrow$  합=3,  $i=3 \rightarrow$  합=6,  $i=4 \rightarrow$  합=10,  $i=5$ . 이제 '합 < 10?'이 거짓(10은 10보다 작지 않음)이라 멈춘다. 최종 합=10.

**💡** 멈추는 조건이 잘못되면 영원히 도는 '무한 반복'이 돼. 반복엔 꼭 끝나는 출구가 있어야 해.

Q92 알고리즘 입문

'이진탐색'은 정렬된 목록에서 가운데를 보고 절반씩 줄여가며 찾는 방법이야.

정렬된 목록 [1, 3, 5, 7, 9, 11, 13] 에서 11 을 이진탐색으로 찾을 때, 값을 비교하는 횟수는 몇 번일까? (가운데 값부터 비교)



- ① ① 1번
- ② ② 2번
- ③ ③ 4번
- ④ ④ 7번

**정답: ② 2번**

**핵심:** 매번 가운데와 비교해 범위를 절반으로 줄인다. 1차 가운데=7,  $11 > 7$  이므로 오른쪽 [9,11,13]로. 2차 가운데=11, 일치! 비교는 2번이면 충분하다.

**💡** 이진탐색은 한 번 비교할 때마다 후보를 절반으로 줄여. 그래서 1000개 중에서도 10번이면 찾을 수 있어!

Q93 변수와 계산

변수 두 개 a, b가 있어. 처음에 a=5, b=2야. 아래 세 줄을 차례로 실행하면 마지막 a의 값은?

블록:

$a = a + b$

$b = a + b$

$a = a + b$

- ① ① 7
- ② ② 9
- ③ ③ 12
- ④ ④ 16

**정답: ④ 16**

**핵심:** 변수는 '지금 들어있는 값'으로 계산한 뒤 새 값을 덮어쓴다.

1)  $a = a + b \rightarrow 5 + 2 = 7$  (a=7, b=2)

2)  $b = a + b \rightarrow 7 + 2 = 9$  (a=7, b=9)

3)  $a = a + b \rightarrow 7 + 9 = 16$  (a=16). 따라서 마지막 a=16.

**💡** 이렇게 '이전 값에 더해 새 값 만들기'를 누적이라고 해. 게임 점수 쌓기와 똑같아!

**Q94** 조건 분기 (여러 갈래)

밝기 센서 값에 따라 글자를 출력하는 조건 블록이야.

만약 빛 < 3 이면 '어두움'

아니면 만약 빛 < 7 이면 '보통'

아니면 '밝음'

빛 = 7 일 때 출력되는 글자는?

- ① ① 어두움
- ② ② 보통
- ③ ③ 밝음
- ④ ④ 아무것도 안 나옴

 **정답: ③ 밝음**

 **핵심:** '< (작다)'는 같은 값을 포함하지 않는다(경계값 주의).

빛=7 → 7<3? 거짓 → 7<7? 거짓(같아도 작은 건 아님) → 둘 다 아니므로 마지막 '밝음'.

 7<7 이 거짓인 걸 깜빡하면 버그가 생겨. 경계값(딱 그 수)은 항상 따로 확인해!

Q95 중첩 반복

바깥 반복과 안쪽 반복이 겹친 블록이야.

반복 i = 2..4 [

반복 j = 1..3 [

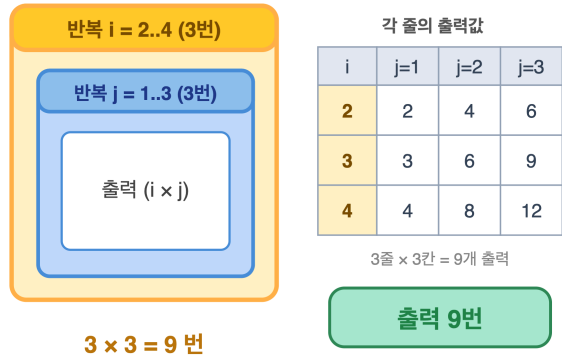
출력 (i × j)

]

]

'출력'은 모두 몇 번 일어날까?

중첩 반복: 출력 횟수



- ① ① 3번
- ② ② 6번
- ③ ③ 9번
- ④ ④ 12번

정답: ③ 9번

핵심: 중첩 반복의 총 횟수는 (바깥 횟수)×(안쪽 횟수).

바깥 i는 2,3,4로 3번, 그 안에서 안쪽 j는 1,2,3으로 3번. 3×3=9번 출력.

💡 구구단 표가 바로 이 중첩 반복이야. 바깥은 '몇 단', 안쪽은 '몇 곱하기'!

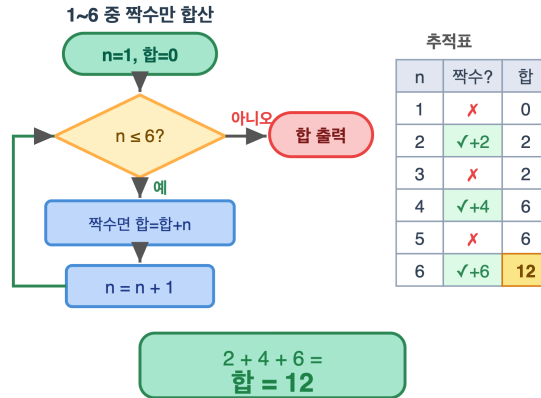
**Q96** 순서도와 흐름

아래 순서도를 따라가면 '합'의 마지막 값은?

시작 →  $n=1$ , 합=0 →  $[n \leq 6 ?]$

예 → 만약  $n$ 이 짝수면 합 = 합 +  $n$  →  $n = n + 1$  → 다시  $[n \leq 6 ?]$

아니오 → 합 출력 → 끝



- ① ① 9
- ② ② 12
- ③ ③ 15
- ④ ④ 21

**정답: ② 12**

책상:  $n$ 을 1부터 6까지 돌며 짝수만 더한다.

$n=1..6$  중 짝수는 2, 4, 6. 합 =  $2+4+6 = 12$ . (홀수 1,3,5는 더하지 않음.)

💡 순서도의 마름모는 '갈림길'이야. 예/아니오로 흐름이 갈라지지!

**Q97** 조건 분기 (여러 갈래)

놀이공원 입장료를 정하는 조건 블록이야. ('그리고'는 둘 다 참, '또는'은 하나만 참이어도 참)

만약 (나이 < 8) 또는 (나이 ≥ 65) 이면 '무료'

아니면 만약 (나이 < 19) 그리고 (학생 = 참) 이면 '반값'

아니면 '정상'

나이 = 15, 학생 = 참 일 때 결과는?

- ① ① 무료
- ② ② 반값
- ③ ③ 정상
- ④ ④ 알 수 없음

**정답: ② 반값**

책상: '또는'은 하나라도 참이면 참, '그리고'는 둘 다 참이어야 참.

1)  $(15 < 8)$  거짓 또는  $(15 \geq 65)$  거짓 → 거짓, '무료' 아님.

2)  $(15 < 19)$  참 그리고 (학생=참) 참 → 둘 다 참이라 참 → '반값'.

💡 '그리고/또는'을 논리 연산이라고 해. 검색창에서 단어 두 개로 거를 때도 이게 작동해!

Q98 알고리즘 입문

목록 [8, 3, 7, 2, 9, 5]에서 가장 작은 수를 찾는 알고리즘이야.

최소 = 첫 번째 값(8)

목록을 차례로 보며, 지금 값이 최소보다 작으면 최소 = 지금 값

끝까지 보는 동안 '최소' 값이 새로 바뀌는(갱신되는) 횟수는 몇 번?



- ① ① 1번
- ② ② 2번
- ③ ③ 3번
- ④ ④ 5번

정답: ② 2번

핵심: 더 작은 값을 만날 때만 갱신된다.

최소=8 시작 → 3(작음, 갱신1) → 7(아니오) → 2(작음, 갱신2) → 9(아니오) → 5(아니오). 총 2번 갱신, 최종 최소=2.

가장 큰 수 찾기도 똑같은 방법이야. '<'를 '>'로, 시작값만 바꾸면 끝!

Q99 의사코드 읽기

아래 의사코드를 실행하면 '개수'에 출력되는 값은?

개수 = 0

반복 n = 1..20 [

만약 (n ÷ 3 의 나머지 = 0) 이면 개수 = 개수 + 1

]

출력 개수

- ① ① 5
- ② ② 6
- ③ ③ 7
- ④ ④ 9

정답: ② 6

핵심: '3으로 나눈 나머지가 0' = 3의 배수. 그걸 셀 때마다 카운터를 1 늘린다.

1~20 중 3의 배수: 3, 6, 9, 12, 15, 18 → 모두 6개. 그래서 개수=6.

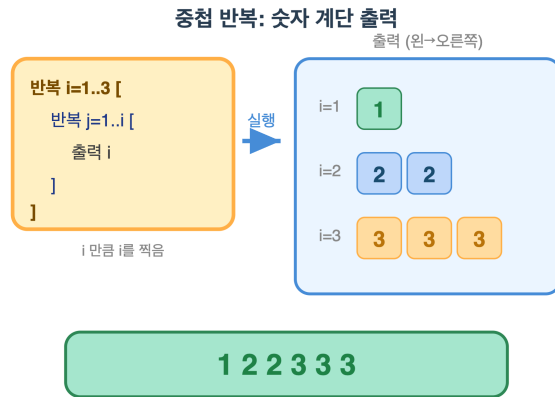
'나머지(mod)'는 짝/홀 판정, 배수 찾기, 시계 계산에 두루 쓰이는 마법 도구야!

**Q100** 중첩 반복

별 대신 '숫자'를 출력하는 중첩 반복이야.

```
반복 i = 1..3 [  
반복 j = 1..i [  
출력 i  
]  
]
```

출력되는 순서를 왼쪽부터 바르게 적은 것은?



- ① ① 1 2 3
- ② ② 1 2 2 3 3 3
- ③ ③ 1 1 2 2 3 3
- ④ ④ 3 3 3 2 2 1

**정답: ② 1 2 2 3 3 3**

**핵심:** 안쪽 반복이  $j=1..i$ 라서  $i$ 가 커질수록 그 숫자를  $i$ 번 출력한다.

$i=1$ : 안쪽 1번 → 1

$i=2$ : 안쪽 2번 → 2 2

$i=3$ : 안쪽 3번 → 3 3 3

이어 붙이면 1 2 2 3 3 3.

💡 안쪽 끝값을 바깥 변수  $i$ 로 두면 삼각형·계단 모양이 자동으로 만들어져!