

flow of control
↳ Conditional
↳ Loops

DPP

5-6 hrs \Rightarrow 40%

Questions \rightarrow 60%

MCQs on
Conditionals in C

What is the correct output of the following code?

```
int a = 5, b = 10;
```

~~if (a < b) { True~~

~~if (a > 0) True~~

~~printf("A");~~

else

 printf("B");

} else

 printf("C");

~~if (a > 0) printf(" A");~~

Single Statement

a = 5

b = 10

~~a) A~~

b) B

c) C

d) No output

if (condition) {
 ≡
}

 ↳ necessary ?
 ↓

Yes → If there are multiple lines in block

* If there is a single statement inside if or else
then block is not necessary.

What is the correct output of the following code?

```
int x = 0;                                x= 0  
if (x == 0) {  
    if (x == 1) false  
        printf("True");  
    else  
        printf("False");  
}
```

- a) True
- b) False
- c) No output
- d) Compilation error

Which of the following is correct about the if-else statement in C?

- a) ~~The else part is optional.~~
- b) The else part must be present.
- c) There can be multiple else blocks.
- d) You cannot nest if-else statements.

if → else

[one if → one else]

What is the correct syntax for the ternary conditional operator?

- a) condition ? expr1;~~expr2~~
- ~~b) condition ? expr1 : expr2~~
- c) condition : expr1 ? Expr2
- d) condition;~~expr1 : expr2~~

What is the output of the following code?

```
int a = 7, b = 0;  
if (a && b) Logical AND  
    printf("True");  
else  
    printf("False");
```

$a = 7 \quad b = 0$
 $a \text{ AND } b$
 \downarrow
True $\&\amp;$ $\not\rightarrow$ False
= False

True AND True = True

Any Non Zero int = True
Zero = False

- a) True
- b) False**
- c) Undefined Behaviour
- d) Compilation error

What is the output of the following code?

```
int a = 5, b = 7;
```

a=5, b=7

```
if (a == 5 && b == 7) → T && T = True  
    printf("Condition 1");
```

```
else if (a == 5 || b == 5)
```

```
    printf("Condition 2");
```

```
else
```

```
    printf("Condition 3");
```

a) Condition 1

b) Condition 2

c) Condition 3

d) Compilation error

Which of the following is a valid conditional expression?

- a) if (a = 5) *Assignment*
 - b) if (a == 5) *True/false*
 - c) if (a =_x 5 && b == 10) *Assign*
 - d) if (a == 5 & b == 10) *Bitwise AND* $\bigcirc \times$
- if (Condition) *True / False*

How many else if clauses can be associated with one if statement?

- a) One
- b) Two
- ~~c)~~ No limit
- d) Only 'else' is allowed after 'if'

Which of the following statements is correct about nested if statements?

- a) else block is required in every nested if. ~~X~~
- b) Nested if statements can only be used once.
- c) Nested if statements can have more than one else.
- d) ~~Nested~~ if statements are allowed in C.

By pass

What happens if the break statement is omitted in a switch case?

- a) Compilation error
- b) Program crashes
- c) Case execution continues to the next case
- d) Only the matched case is executed

```
Switch (exp){  
Case A :  
    True  
    ==  
    Break ;
```

Which of the following causes a runtime error?

- a) if ($x > 0$) No
 - b) if ($x < 0$) No
 - c) if ($x == 0$) No
 - ~~d) if ($x / 0$)~~ Error
- } True/false

$$\frac{n}{0} \rightarrow \underline{\text{Error}}$$

What is the output of the following code?

```
int x = 3, y = 5;           x = 3  
if (x > 0) {                y = 5  
    if (y < 10) {           True  
        printf("Yes");     True  
    } else {                Yes  
        printf("No");      No  
    }                         terminated  
}                            outside else  
                             Independent
```

- a) YesNo
- b) Yes
- c) No
- d) Compilation error

What will be the output of the following code?

```
a = 10  
int a = 10;  
if (a % 2 == 0 && (a = a / 2))  
    printf("%d", a);  
else  
    printf("%d", a);
```

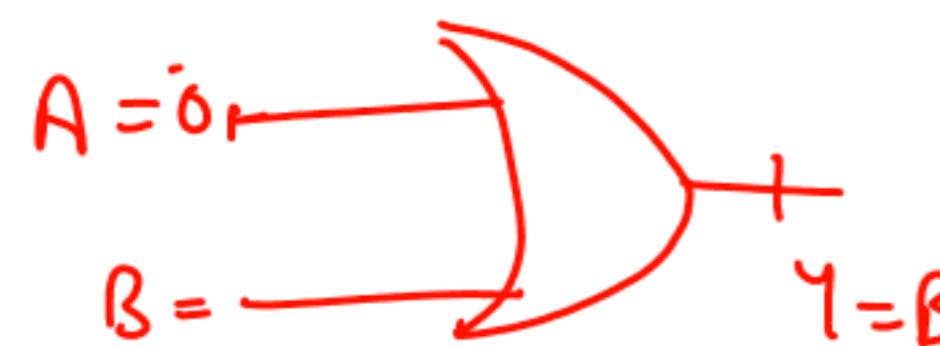
Handwritten annotations:

- True && True
- 5 ← Non zero (True)
- ↓
5
- a = 10/2
- a = 5

- a) 10
- b) 5
- c) 0
- d) Undefined behavior

Which statement about short-circuit evaluation in C is correct?

- a) Both operands of logical `&&` are always evaluated.
- b) Both operands of logical `||` are always evaluated.
- c) The second operand of `&&` is only evaluated if the first operand is false.
- d) ~~The second operand of `||` is only evaluated if the first operand is false.~~



What is the result of the following code?

```
int x = 2, y = 8;
```

```
if (x- && ++y)  
    ↪ True
```

```
    printf("X");
```

```
else
```

```
    printf("Y");
```

```
printf("%d %d", x, y);
```

$x = 2$

$y = 9$

$2 \&& 9$

$T \&& T = True$

a) X 1 9

b) Y 1 9

c) X 2 8

d) Compilation error

Output = X 1 9

What will be the output of the following code?

```
int a = 2, b = 0, c = 4;  
if (a && (b || c++))  
    printf("%d", c);
```

else

```
    printf("False");
```

a) 4

~~b) 5~~

c) False

d) Compilation error

$a = 2$
 $b = 0$
 $c = 4$

$\text{if } (a \&\& (b \text{ || } c++))$

$a \downarrow$
 $2 \downarrow$
 $\text{True} \&\& \text{True}$
 $b \downarrow$
 $0 \text{ || } 4 \downarrow$
 $\text{True} \text{ || } \text{True}$
 $= \text{True}$

OR → Any one input is True the O/P = True

AND → All inputs must be true to get True output

Which of the following is true for this code snippet?

```
int x = 1, y = -1;
```

```
if (x > 0 || y++ < 0)
```

```
    ↳ printf("%d", y);
```

```
else
```

```
    printf("%d", y++);
```

$x = 1, y = -1$

~~a) -1~~

b) 0

c) Undefined behavior

d) Compilation error

$\text{if } (x > 0 \text{ } || \text{ } y++ < 0)$

True OR Skip ← Short Circuit Eval.

True

What will the following code print?

```
int x = 2, y = 3;
```

```
if (x-- > 0 && ++y > 2) True
```

```
    printf("X is %d and Y is %d", x, y);
```

```
else
```

```
    printf("Condition failed");
```

$x = 2$
 $y = 3$

$\text{if } (x-- > 0 \&\& ++y > 2)$
 $2 > 0 \&\& 4 > 2$
True & True
True

- a) X is 1 and Y is 4
- b) X is 1 and Y is 3
- c) Condition failed
- d) Undefined behavior

~~0/0~~ X is 1 and Y is 4

Which of the following will result in an error in a conditional expression?

```
int x = 10, y = 20;
```

```
if (x == y || x < 5 && ++y) {
```

```
    printf("Pass");
```

```
else
```

```
    printf("Fail");
```

a) Fail

b) Pass

c) Undefined behavior

d) Compilation error

$$\begin{aligned}x &= 10 \\y &= \cancel{20} \rightarrow 21\end{aligned}$$

$\text{if } (x == y \text{ || } x < 5 \text{ && } ++y)$

$$x == y \text{ || } x < 5 \text{ && } 21$$

$$\text{False} \text{ || } \text{True} \text{ && } \cancel{21} \rightarrow \text{True}$$

$$\text{False} \text{ || } \text{True}$$

True

Relational ↑
AND
OR

Which of the following is true about the ternary operator in C?

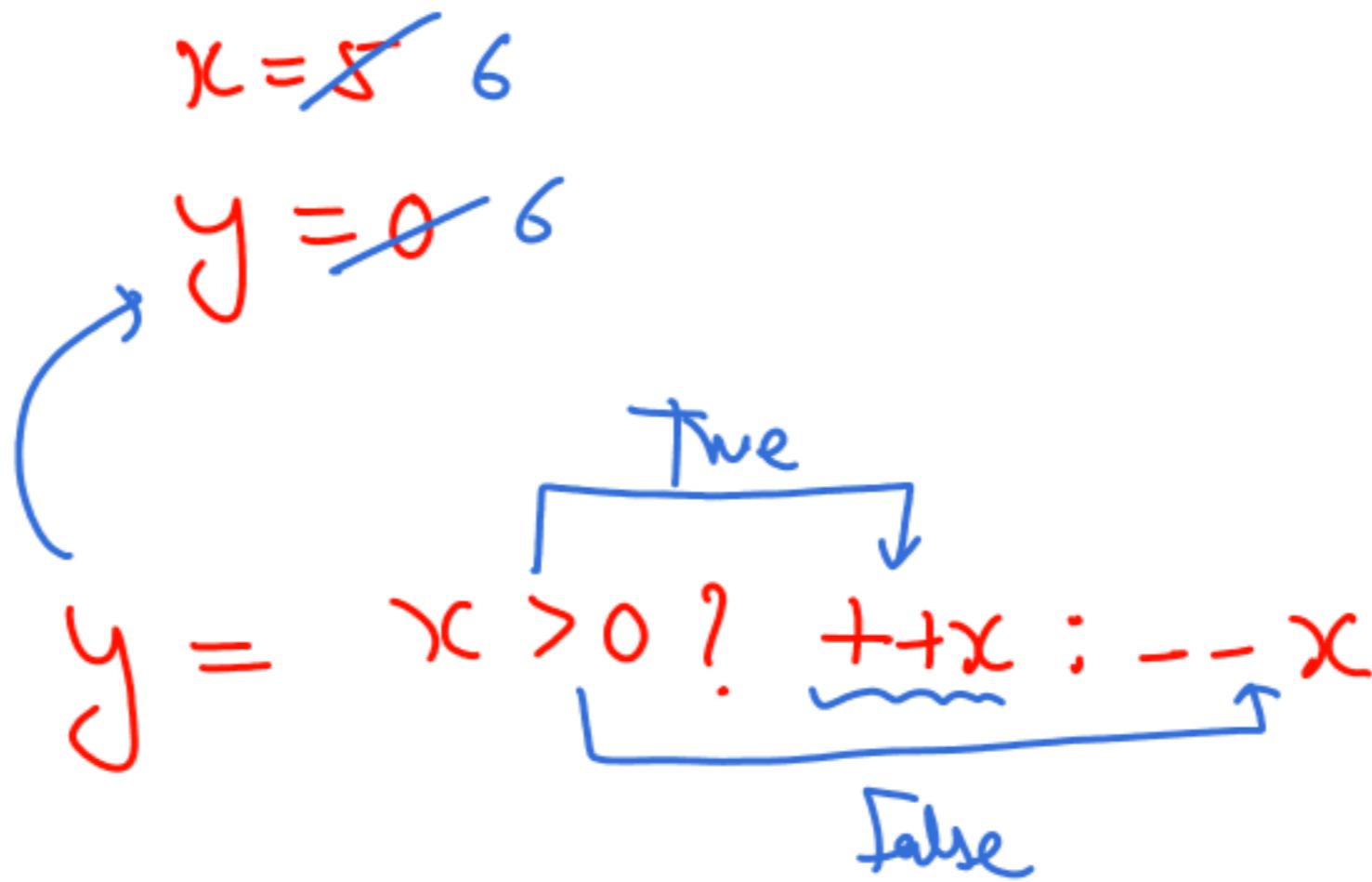
- a) It can evaluate only one expression per branch. ✓
- b) The expression on both sides of : must be of the same type. ✓
- ~~c) You can use it in place of an if-else without any restrictions.~~
- d) It must be nested within an if-else structure. ✗

What is the output of this code?

```
int x = 5, y = 0;
```

```
y = x > 0 ? ++x : --x;
```

```
printf("%d %d", x, y);
```



- a) 6 6
- b) 6 5
- c) 5 6
- d) Undefined behavior

What is the output of this code?

```
int a = 5, b = 10;
```

```
if (a-- > 0) → a-- > 0
```

```
  ↳ if (++b > 10)
```

```
    ↳ printf("A");
```

```
else
```

```
    printf("B");
```

```
printf("%d %d", a, b);
```

4
!!

a = 4

b = 10 11

if (++b > 10)

11 > 10 - True

✓ a) A 4 11

b) B 4 11

c) B 4 10

d) A 5 11

Op

A 4 11

What is the output of this code?

```
int x = 5, y = 0;
```

```
if (x && y++)
```

```
    printf("True");
```

```
else
```

```
    printf("False");
```

```
→ printf("%d", y);
```

a) False 0

b) True 1

c) False 1

d) Undefined behavior

$x = 5$

$y = 0 \uparrow$

if ($x \&& y++$)

\downarrow
 \downarrow

$5 \&& 0 \Rightarrow \text{True} \&& \text{False} \Rightarrow \text{False}$

o/p

False 1

Which of the following expressions evaluates to true?

a) $!(1 \text{ || } 0)$ → false

b) $!1 \text{ && } 0$ → false

c) $!1 \text{ || } 0 \Rightarrow 0 \text{ OR } 0 = \text{False}$

~~d) $!(1 \text{ && } 0) \Rightarrow !(\underbrace{1 \text{ && } 0}_{!0})$~~
 \downarrow
 $!0 \Rightarrow 1$

Not (false) = True

What is the output of this code?

```
int a = 4, b = 7;
```

```
if ((a < 5 && b > 6) || !(a < 5))  
    ↘ True  
    ↘ printf("True");
```

```
else
```

```
    printf("False");
```

- a) True
- b) False
- c) Compilation error
- d) Undefined behavior

a = 4

b = 7

Relational
AND
OR

if ((a < 5 && b > 6) || !(a < 5))
 ↓ ↓
 (True && True) || ! (True)
 ↓ ↓
 True False
 ↓
 True

Which of the following will print "False"?

- a) if (5 == 5 || 0) ← True
- b) if (5 == 6 && 1) ← false
false &
- c) if (5 != 5 || 1) → False
false || True
- d) if (5 > 4 && !1)
 True AND False
 False

What will happen when the following code is executed?

```
int a = 5, b = 0;  
if (++a && ++b) True  
    ↴ printf("True");  
else  
    printf("False");  
printf("%d %d", a, b);  
      ↪ 6  
      ↪ 1
```

$$a = 6$$

$$b = 1$$

if (++a && ++b)
6 && 1 = True

Output = True 6 1

- a) True 6 1
- b) False 6 0
- c) False 5 0
- d) Compilation error

What will be the output of the following code?

Short Circuit

```
int x = 10, y = 0;
```

$x = 10$

$y = 0$

```
if (x || ++y) skip ← True OR
```

```
    10 printf("X is %d and Y is %d", x, y);  
    10 0
```

```
else
```

```
    printf("False");
```

- a) X is 10 and Y is 1
- b) X is 10 and Y is 0
- c) False
- d) Compilation error

What will be the output of the following code?

```
int a = 5, b = 10;  
if ((a > b) ? a++ : b--) True  
    ↳ printf("A = %d, B = %d", a, b); 5 9  
else  
    printf("No Change");
```

~~a) A = 5, B = 9~~

- b) A = 6, B = 10
- c) A = 6, B = 9
- d) No Change

$a = 5$
 $b = 10$ 9

if ((a > b) ? a++ : b--)
 ↳ True ↓
 ↳ False ↓
 ↳ 10
if (10)
 ↳ Non Zero Number ← True

What will be printed by the following code?

```
int a = 5, b = 3;
```

```
if ((a < b) ? a-- : b++)
```

```
    printf("%d %d", a, b);
```

```
else
```

```
    printf("No Change");
```

$a = 5$
 $b = 4$

if [$(a < b) ? a-- : b++$]

True
False

if (3)

Non Zero Number \leftarrow True

- a) 5 4
- b) 4 3
- c) 5 3
- d) No Change

Loops in C

What will be the output of the following code?

init Var \Rightarrow $i = \cancel{0} \cancel{1} \cancel{2} 3$
 $0 \rightarrow 4$

```
int i = 0;  
for((); i < 5; i++) {  
    if (i == 3)  
        break;  
     $\Rightarrow$  printf("%d ", i);  
}
```

- a) 0 1 2
b) 0 1 2 3
c) 0 1 2 3 4
d) Infinite loop

Cases

① $i = 0$
 $i < 5 (\top)$

\hookrightarrow if ($i == 3$) \Rightarrow false
 $i++ \Rightarrow i = 1$

② $i = 1, i < 5 (\top)$
 $\hookrightarrow (i == 3) (\text{False})$
 $i++ \Rightarrow i = 2$

③ $i = 2, i < 5 (\top)$
 $\hookrightarrow i == 3 (\text{F})$
 $i++ \Rightarrow i = 3$

④ $i = 3, i < 5 (\top)$
 $\hookrightarrow i == 3 (\top)$
 $\hookrightarrow \text{Break} \leftarrow \underline{\text{terminate}}$

Output
0:1:2

What will be the output of the following code?

$i = \phi - 1$

```
int i = 0;  
do {  
    printf("%d ", i++);  
} while (i < 0);
```

$i++ \leftarrow$ Return value first then increment

$i < 0$ (false)

0
1

- a) 0
- b) 0 1
- c) 1
- d) No output

What will be the output of the following code?

```
int i;  
for (i = 0; i < 3; i++) {  
    i++;  
    printf("%d ", i);  
}
```

① $i=0, \quad i < 3$ (True)

$i++ \rightarrow i=1$

$i++ \rightarrow i=2$

② $i=2, \quad i < 3$ (True)

$\hookrightarrow i++ \rightarrow i=3$

printf(i)

$i++ \Rightarrow i=4$

③ $i=4, \quad i < 3$ (F)

$\hookrightarrow \times$

$[0,1,2]$
 \downarrow
 $i=0, 1, 2, 3, 4$
 $0 \rightarrow 2$

Output
1 3

- a) 0 1 2
- b) 1 3
- c) 1 2 3
- d) 1 2

Which of the following conditions will cause an infinite loop?

```
int i = 1;           i <= 5 will always true  
while (i <= 5) {  
    printf("%d", i);  
    i--;  
}
```

- a) No infinite loop
- b) Infinite loop
- c) Compilation error
- d) Undefined behavior

① $i = 1, i \leq 5$ (True)
 $\hookrightarrow \text{print}(i)$
 $i--$

② $i = 0, i \leq 5$ (True)
 $\hookrightarrow \text{print}(i)$
 $i--$

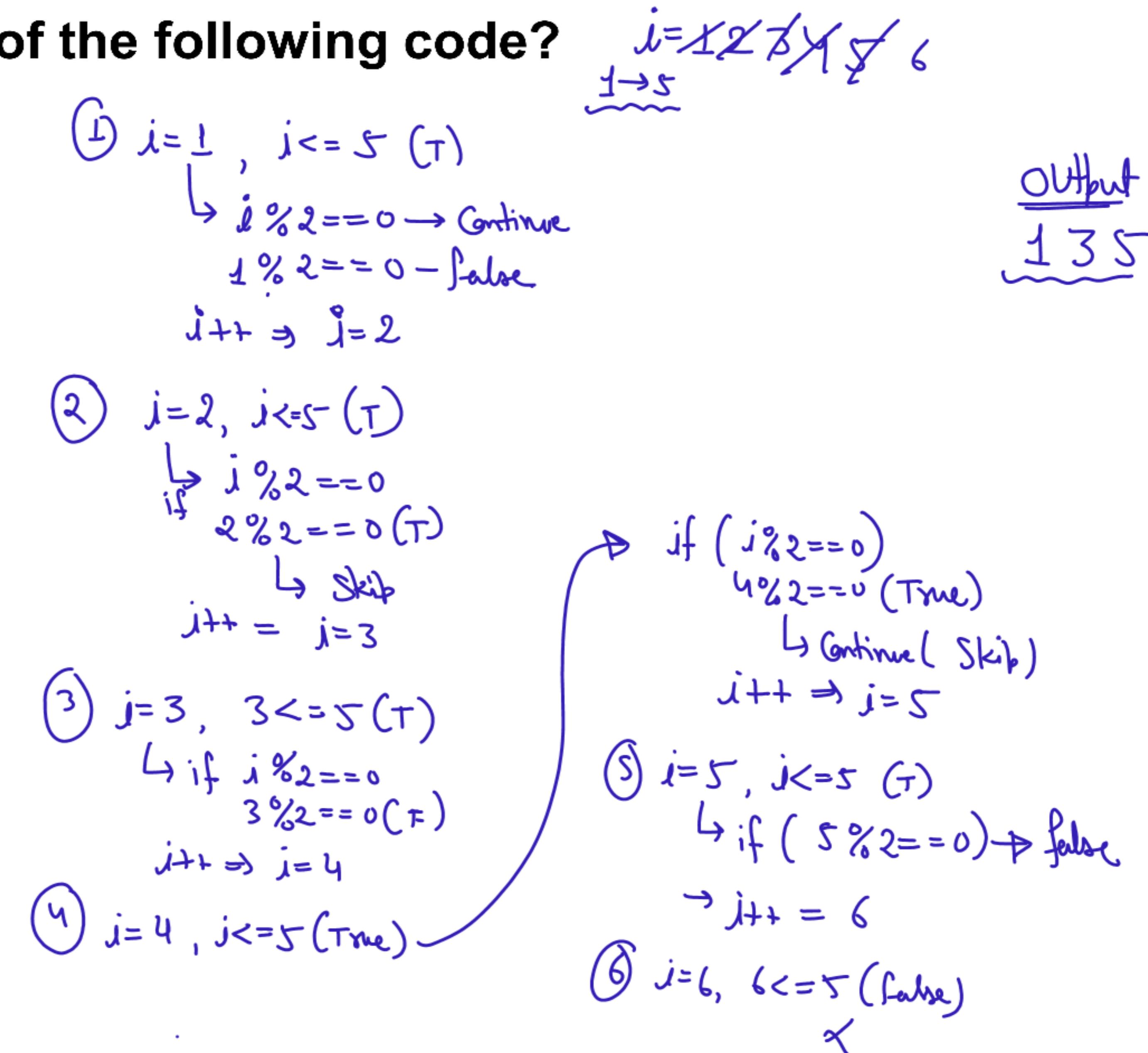
③ $i = -1, i \leq 5$ (True)
 $i = -2$ (T)
 $i = -3$ (T)
 \vdots
 $i = -\infty$ (T)

Output
10

What will be the output of the following code?

```
for (int i = 1; i <= 5; i++) {  
    if (i % 2 == 0)  
        continue;  
    printf("%d ", i);  
}
```

- a) 1 2 3 4 5
- b) 1 3 5
- c) 2 4
- d) Compilation error



What will be the output of the following code?

```
int i = 1;  
while (i <= 10) {  
    if (i == 5)  
        i += 2;  
    printf("%d ", i);  
    i++; ✓  
}
```

- a) 1 2 3 4 7 8 9 10
- b) 1 2 3 4 5 6 7 8 9 10
- c) 1 2 3 4 6 7 8 9 10
- d) Infinite loop

① $i=1, i \leq 10$ (True)

↳ if ($i==5$) X
 $i++ \Rightarrow i=2$

② $i=2, 2 \leq 10$ (T)

↳ if ($i==5$) X
 $i++ \Rightarrow i=3$

③ $i=3, 3 \leq 10$ (T)

↳ if ($i==5$) X
 $i++ \Rightarrow i=4$

④ $i=4, 4 \leq 10$ (T)

↳ if ($i==5$) X
 $i++ \Rightarrow i=5$

⑤ $i=5, 5 \leq 10$ (T)

↳ if ($i==5$) → True
 $i+=2$

$i=1, 2, 3, 4, 7, 8, 9, 10, 11$
 $i \rightarrow 10$

Output

1; 2; 3; 4; 7; 8; 9;
10

$j = i+2$

$i = 5+2=7$

$j = 7$

$j++ \Rightarrow j=8$

⑥ $i=8; 8 \leq 10$ (T)

↳ $i++ \Rightarrow i=9$

⑦

$j=9, 9 \leq 10$ (T)

↳ $j++ \Rightarrow j=10$

⑧

$i=10, 10 \leq 10$ (T)

↳ $j++ \Rightarrow j=11$

⑨ $i=11, 11 \leq 10$

(false)

Break

What is the output of this code?

[Any Non zero number = True
and zero = False]

```
int i = 5;  
while (i > 0) { false  
    printf("%d ", i);  
}
```

- a) 5 4 3 2 1
- b) 5 4 3 2 1 0
- c) ~~4 3 2 1 0~~
- d) Infinite loop

- ① i=5 →
- ② i=4
- ③ i=3
- ④ i=2
- ⑤ i=1
- ⑥ i=0

i = ~~5 4 3 2 1 0~~ -1

output

4|3|2|1|0

What will be the output of the following code?

A) for (int i = 0; i < 5; i++) {
 B) for (int j = 0; j < i; j++) {
 printf("%d ", j);
 }
 printf("\n");
}

- a) 0 1 2 3 4
- b) 0 1 2 3
- c) 0
- d) Prints a triangle pattern of numbers

① A) $j=0$; $0 < 5$ (True)
 B) ① $j=0$; $j < i$ (False)
 X

② A) $j=1$, $j < 5$ (True)
 B) $j=0$, $j < 1$ (True)
 print(0)
 $j++ \rightarrow j=1$
 $j=1$, $j < 1$ ($1 < 1$) - false

③ A) $j=2$, $j < 5$ (True)
 B) $j=0$, $j < i$ (True) \rightarrow print(j), $j++ = j=1$
 $j=1$, $j < i$ (True) \rightarrow print(j), $j++ \rightarrow j=2$
 $j=2$, $2 < 5$ (True) \rightarrow print(j), $j++ \rightarrow j=3$

A) $j=0$; $0 < 5$ (True)
 B) $j=0$
 0 \rightarrow j

④ A) $j=3$, $3 < 5$ (True)
 B) $j=0$, $j < i$ (True) \rightarrow print(j), $j++$
 $j=1$, $j < i$ (True) \rightarrow print(j), $j++$
 $j=2$, $2 < 3$ (True) \rightarrow print(j), $j++$
 $j=3$, $3 < 3$ (False)

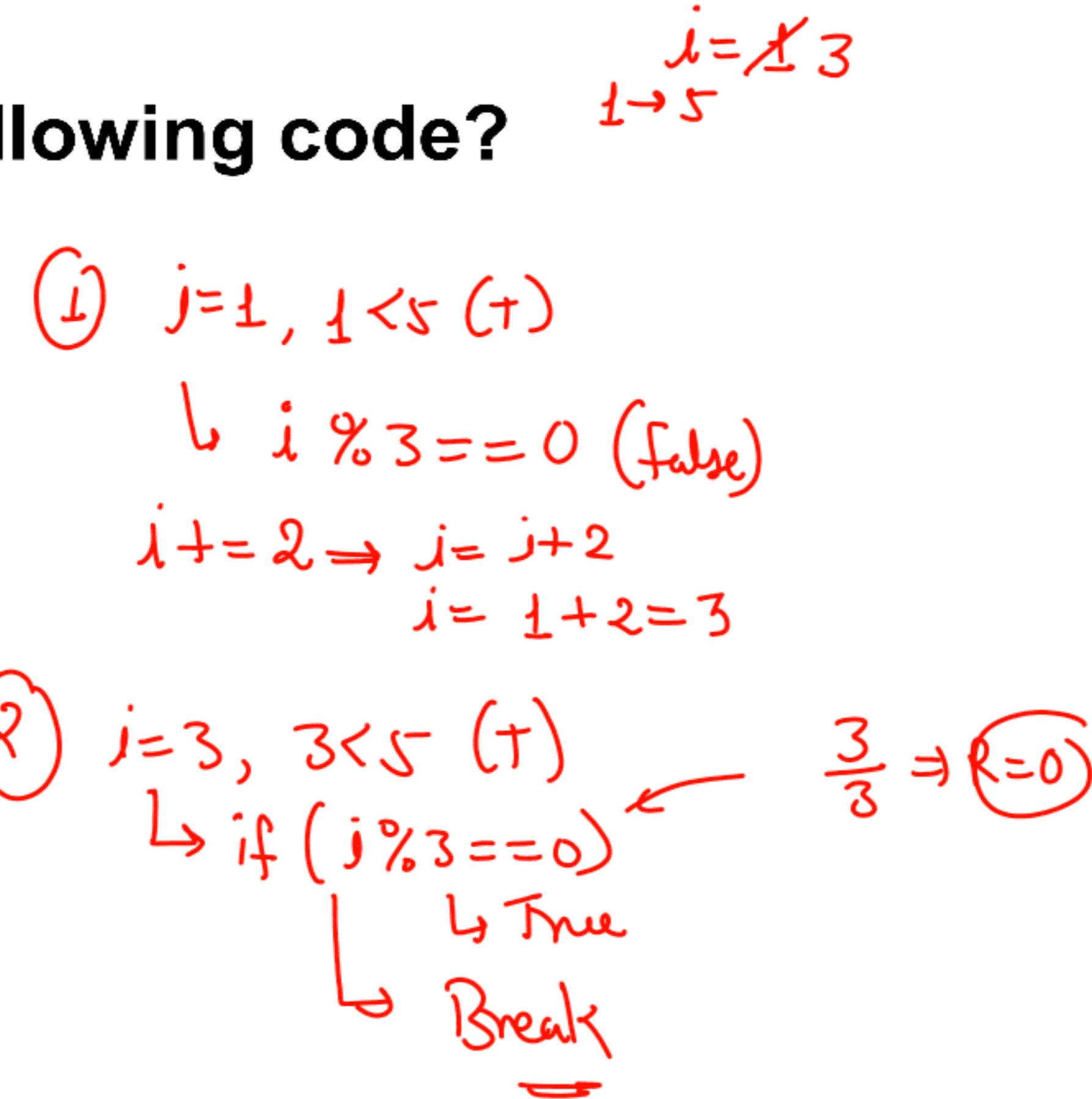
Output

0:
0 1
0 1 2

What will be printed by the following code?

```
int i;  
for (i = 1; i <= 5; i += 2) {  
    printf("%d ", i);  
    if (i % 3 == 0)  
        break;  
}
```

- a) 1 3
- b) 1 3 5
- c) 1 5
- d) 3 5



Output
1 3

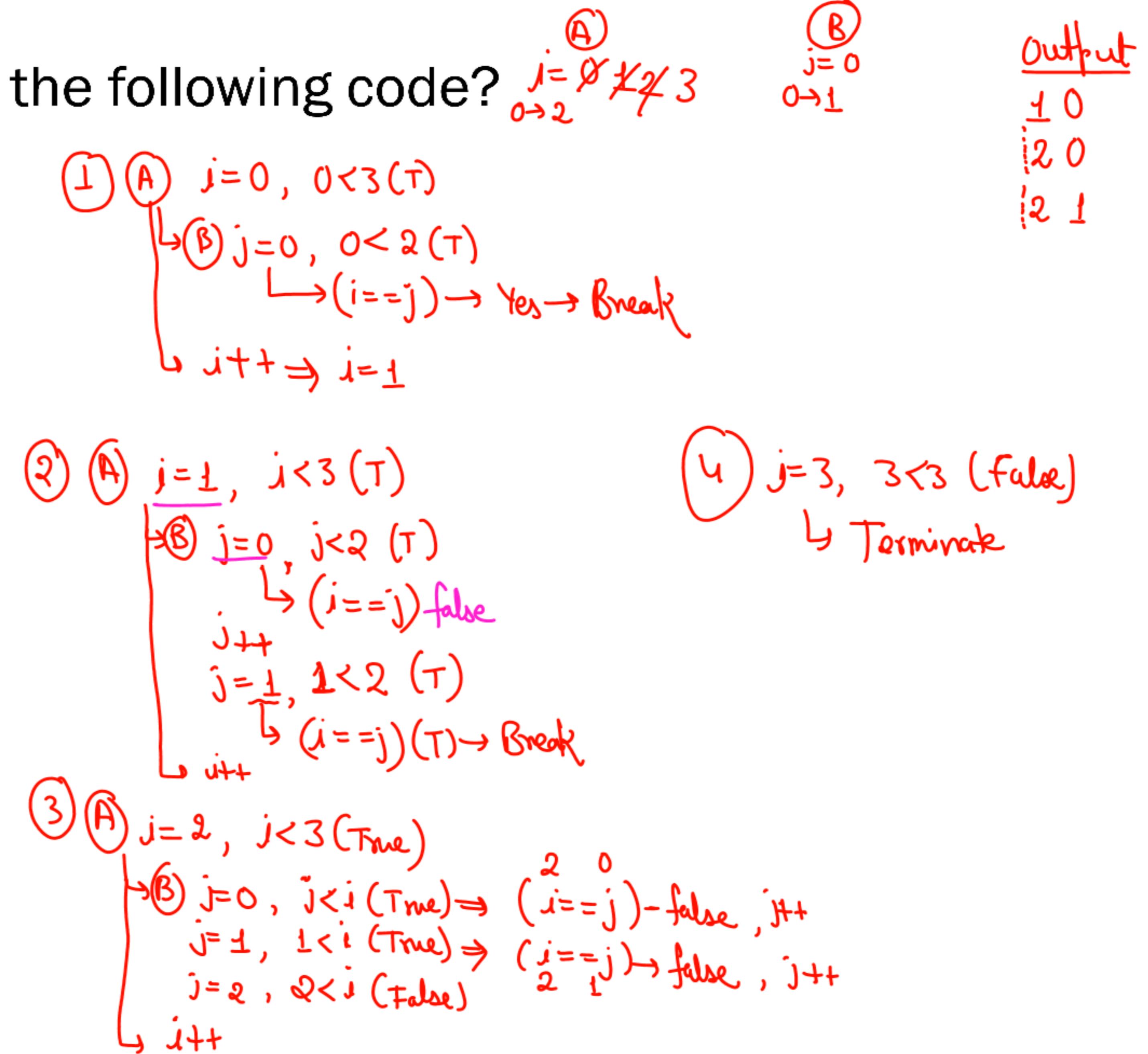
What will be the output of the following code?

```

① A
for (int i = 0; i < 3; i++) {
    ② B
    for (int j = 0; j < 2; j++) {
        if (i == j)
            break;
        printf("%d %d\n", i, j);
    }
}

```

- a) 0 0
- b) 1 0
- c) 1 1
- d) 0 1 1 0



What will be printed by the following code?

$j = \cancel{0} \neq 9$
 $0 \rightarrow 9$

```
int i = 0;  
for (; i < 10; i += 3) {  
    if (i == 6)  
        continue;  
    printf("%d", i);  
}
```

- a) 0 3 6 9
- b) 0 3 9
- c) 0 3 6
- d) 3 6 9

① $j = 0, 0 < 10$ (True)
 $\downarrow j == 6 \rightarrow \text{false}$
 $i += 3 \Rightarrow j = j + 3$
 $j = 0 + 3 = 3$

② $j = 3, 3 < 10$ (True)
 $\downarrow (j == 6) \leftarrow \text{false}$
 $i += 3 \Rightarrow j = j + 3$
 $j = 3 + 3 = 6$

③ $j = 6, 6 < 10 \rightarrow \text{True}$
 $\downarrow (j == 6) - \text{True}$
 $\downarrow \text{Continue}$
 $i += 3, j = j + 3$
 $j = 6 + 3 = 9$
 $j = 9$

④ $j = 9, 9 < 10$ (True)
 $\downarrow i = i + 3$
 $i = 9 + 3 = 12$

⑤ $j = 12, 12 < 10$ (False)
 \downarrow
 Terminate

Output

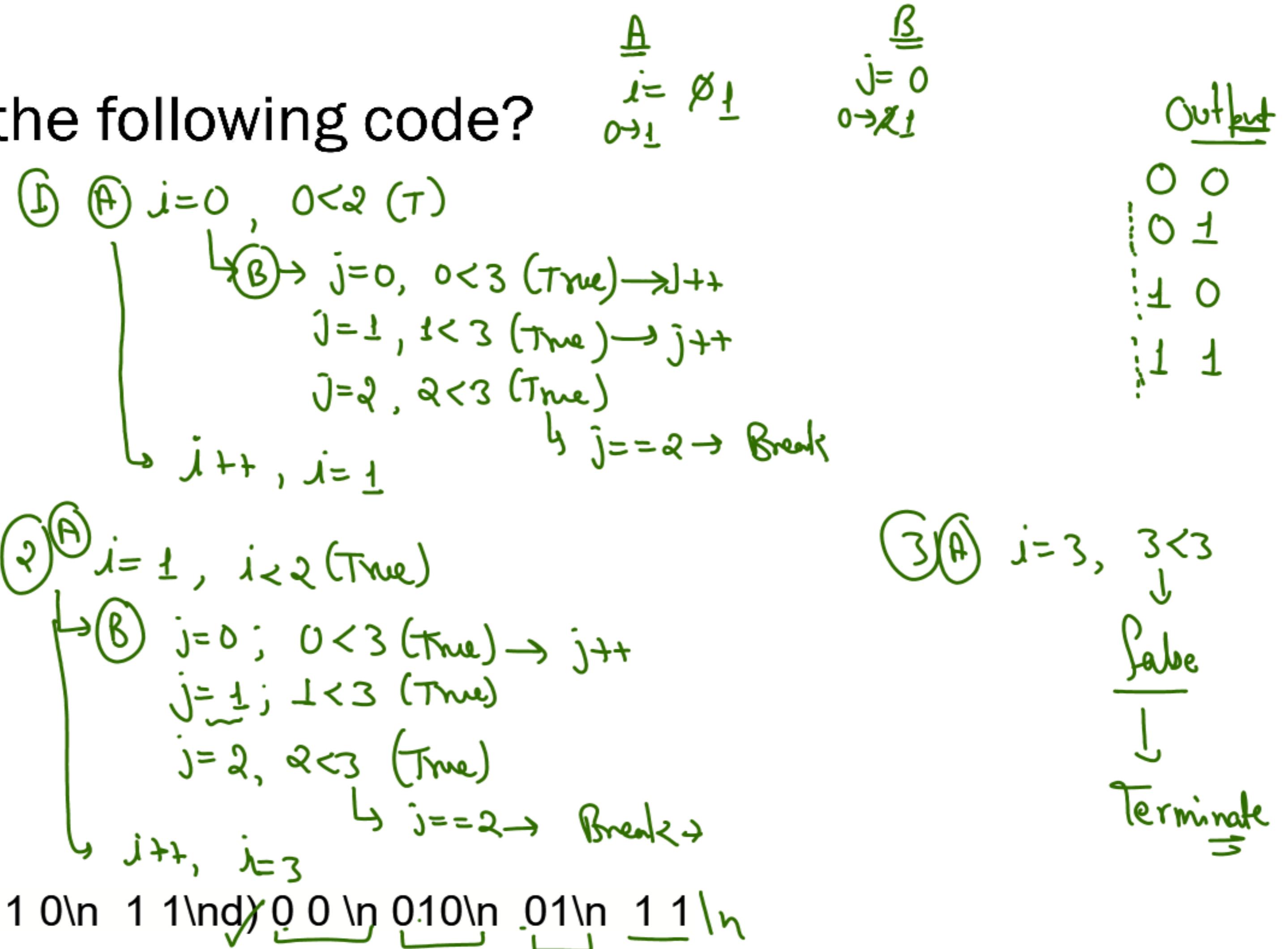
0 3 9

What will be printed by the following code?

```

int i = 0, j = 0;
①(A) for (i = 0; i < 2; i++) {
    ②(B) for (j = 0; j < 3; j++) {
        if (j == 2)
            break;
        →printf("%d %d\n", i, j);
    }
}

```



- a) 0 0 \nb) 0 0 \n 1 0c) 0 0\n 0 1\n 1 0\n 1 1\nd) 0 0 \n 0 1 \n 0 1 \n 1 1 \n
- b) 0 0 1 0
- c) 0 0 0 1 1 0 1 1
- d) 0 0 0 1 1 0 1 1

What will be the output of the following code?

~~j=0 & j=3~~
~~j=1 & j=3~~

```
for (int i = 0, j = 0; i + j < 5; i++, j++) {  
    printf("%d ", i + j);  
}
```

Output
0 2 4

- a) Compilation error
- ~~b) 0 2 4~~
- c) 0 1 2 3 4
- d) 0 2

① $i=0$
 $j=0$
 $i+j < 5$ (True)
↳ $i=1, j=1$
 $i+j < 5$
 $2 < 5 \leftarrow \text{True}$
↳ $\text{print}(i+j)$

③ $i=2, j=2$
 $i+j < 5 \Rightarrow 2+2 < 5 \Rightarrow 4 < 5$ (True)
↳ $\text{print}(i+j)$

④ $i=3, j=3$
 $i+j < 5 \Rightarrow 6 < 5$ (False)

What will be the output of the following code?

```
int i = 2, j = 0;  
while (i++ < 5) {  
    j += i;  
}  
printf("%d", j);  
  
a) 12  
b) 15  
c) 10  
d) 9
```

$$\begin{array}{l} i=2 \cancel{3} \cancel{4} \cancel{5} 6 \\ j=\cancel{2} \cancel{3} \cancel{4} \underline{\underline{12}} \end{array}$$

- ① $i++ < 5$
 $2 < 5 \rightarrow \text{True}$
 \downarrow
 $j += i \Rightarrow j = j + i$
 $j = 0 + 3 = 3$
- ② $i=3, j=3$
 $(i++ < 5)$
 $(3 < 5) \rightarrow \text{True}$
 \downarrow
 $j += i \Rightarrow j = j + i$
 $j = 3 + 4$
 $j = 7$

③ $(i++ < 5)$
 \downarrow
 $(4 < 5) \rightarrow \text{True}$
 \downarrow
 $j = j + i$
 $j = 7 + 5 \Rightarrow \underline{\underline{12}}$

④ $(i++ < 5)$
 \downarrow
 $(5 < 5) \rightarrow \text{False}$
 \downarrow
Terminate

What the following program do?

```
int i = 0;  
for (i = 10; i != 0; i--) {  
    if (i % 2 == 1)  
        i++;  
}
```

- a) Infinite loop
- b) No infinite loop
- c) Compilation error
- d) Undefined behavior

$$j = \cancel{0} \cancel{1} \cancel{2} \cancel{3} \cancel{4} \cancel{5} \cancel{6} \cancel{7} \cancel{8} \cancel{9}$$

① $i \neq 0$ (True)
 \hookrightarrow if ($i \% 2 == 1$) \rightarrow false

② $j--$
 $i = 9, j \neq 0$ (True)
 \hookrightarrow if ($i \% 2 == 1$)
 \hookrightarrow $i++ \rightarrow j = 10$

$$\frac{9}{2} \Rightarrow R=1$$

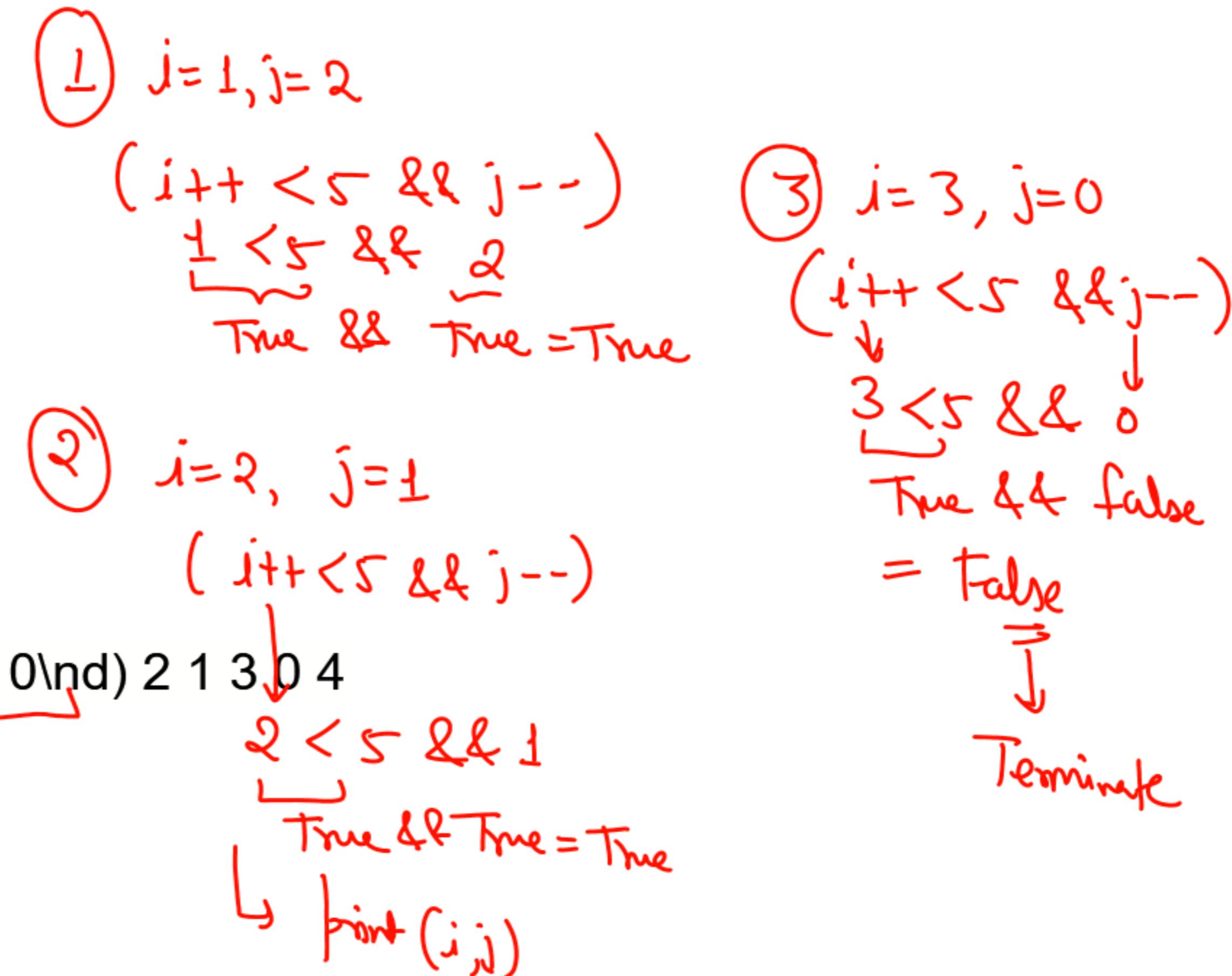
③ $i--, j = 10$, $i \neq 0$ ($10 \neq 0$) True
 \hookrightarrow if ($i \% 2 == 1$)
 \hookrightarrow $i++$, $i = 10$
 $i-- \Rightarrow j = 9$

What will be printed by the following code?

```
int i = 1, j = 2;
```

```
while (i++ < 5 && j--) {  
    printf("%d %d\n", i, j);  
}
```

- a) 2 1\nb) 2 1 \n 3 0\n 4 -1c) 2 1 \n 3 0\nd) 2 1 3 0 4



$i=1 2 3 4$
 $j=2 1 0 -1$

Output
2 1
3 0

What will be printed by the following code?

```
int i = 1;  
while (i++ <= 5) {  
    printf("%d ", i);  
    i++;  
}
```

- a) 1 2 3 4 5
- ~~b) 2 4 6~~
- c) 2 3 4 5
- d) 3 5

~~j = 1 2 3 4 5 6 7 8~~

Output
2 4 6

① $j = 1$
 $(i++ \leq 5)$
 \downarrow
 $(1 \leq 5) \Rightarrow \text{True}$
 \downarrow
 $i++$

③ $j = 5$
 $(i++ \leq 5)$
 \downarrow
 $(5 \leq 5) \Rightarrow \text{True}$
 \downarrow
 $i++$

② $j = 3$
 $(i++ \leq 5)$
 \downarrow
 $(3 \leq 5) \Rightarrow \text{True}$
 \downarrow
 $i++$

④ $j = 7$
 $(i++ \leq 5)$
 \downarrow
 $(7 \leq 5) \leftarrow \text{false}$
Terminate

What will be printed by the following code?

```
int i, j, sum = 0;  
for (i = 0; i <= 5; i++) {  
    for (j = 0; j < 5; j++) {  
        sum += i * 5 + j;  
    }  
}  
printf("%d ", sum)
```

= 350

① A j = 0, i <= 5 (True)

 ② B j = 0, 0 < 5 (True), j++

 sum += j * 5 + j

 sum += 0 * 5 + 0

 sum += 1 * 5 + 1

 sum += 0 * 5 + 1 \Rightarrow sum += 1

 sum += 1 * 5 + 2

 sum += 0 * 5 + 2 \Rightarrow sum = sum + 2

 sum += 2 * 5 + 3

 sum += 1 * 5 + 4

 sum += 0 * 5 + 5

 sum += 5 * 5 + 6

 sum += 4 * 5 + 7

 sum += 3 * 5 + 8

 sum += 2 * 5 + 9

 sum += 1 * 5 + 10

 sum += 0 * 5 + 11

 sum += 5 * 5 + 12

 sum += 4 * 5 + 13

 sum += 3 * 5 + 14

 sum += 2 * 5 + 15

 sum += 1 * 5 + 16

 sum += 0 * 5 + 17

 sum += 5 * 5 + 18

 sum += 4 * 5 + 19

 sum += 3 * 5 + 20

 sum += 2 * 5 + 21

 sum += 1 * 5 + 22

 sum += 0 * 5 + 23

 sum += 5 * 5 + 24

 sum += 4 * 5 + 25

 sum += 3 * 5 + 26

 sum += 2 * 5 + 27

 sum += 1 * 5 + 28

 sum += 0 * 5 + 29

 sum += 5 * 5 + 30

 sum += 4 * 5 + 31

 sum += 3 * 5 + 32

 sum += 2 * 5 + 33

 sum += 1 * 5 + 34

 sum += 0 * 5 + 35

 sum += 5 * 5 + 36

 sum += 4 * 5 + 37

 sum += 3 * 5 + 38

 sum += 2 * 5 + 39

 sum += 1 * 5 + 40

 sum += 0 * 5 + 41

 sum += 5 * 5 + 42

 sum += 4 * 5 + 43

 sum += 3 * 5 + 44

 sum += 2 * 5 + 45

 sum += 1 * 5 + 46

 sum += 0 * 5 + 47

 sum += 5 * 5 + 48

 sum += 4 * 5 + 49

 sum += 3 * 5 + 50

 sum += 2 * 5 + 51

 sum += 1 * 5 + 52

 sum += 0 * 5 + 53

 sum += 5 * 5 + 54

 sum += 4 * 5 + 55

 sum += 3 * 5 + 56

 sum += 2 * 5 + 57

 sum += 1 * 5 + 58

 sum += 0 * 5 + 59

 sum += 5 * 5 + 60

 sum += 4 * 5 + 61

 sum += 3 * 5 + 62

 sum += 2 * 5 + 63

 sum += 1 * 5 + 64

 sum += 0 * 5 + 65

 sum += 5 * 5 + 66

 sum += 4 * 5 + 67

 sum += 3 * 5 + 68

 sum += 2 * 5 + 69

 sum += 1 * 5 + 70

 sum += 0 * 5 + 71

 sum += 5 * 5 + 72

 sum += 4 * 5 + 73

 sum += 3 * 5 + 74

 sum += 2 * 5 + 75

 sum += 1 * 5 + 76

 sum += 0 * 5 + 77

 sum += 5 * 5 + 78

 sum += 4 * 5 + 79

 sum += 3 * 5 + 80

 sum += 2 * 5 + 81

 sum += 1 * 5 + 82

 sum += 0 * 5 + 83

 sum += 5 * 5 + 84

 sum += 4 * 5 + 85

 sum += 3 * 5 + 86

 sum += 2 * 5 + 87

 sum += 1 * 5 + 88

 sum += 0 * 5 + 89

 sum += 5 * 5 + 90

 sum += 4 * 5 + 91

 sum += 3 * 5 + 92

 sum += 2 * 5 + 93

 sum += 1 * 5 + 94

 sum += 0 * 5 + 95

 sum += 5 * 5 + 96

 sum += 4 * 5 + 97

 sum += 3 * 5 + 98

 sum += 2 * 5 + 99

 sum += 1 * 5 + 100

 sum += 0 * 5 + 101

 sum += 5 * 5 + 102

 sum += 4 * 5 + 103

 sum += 3 * 5 + 104

 sum += 2 * 5 + 105

 sum += 1 * 5 + 106

 sum += 0 * 5 + 107

 sum += 5 * 5 + 108

 sum += 4 * 5 + 109

 sum += 3 * 5 + 110

 sum += 2 * 5 + 111

 sum += 1 * 5 + 112

 sum += 0 * 5 + 113

 sum += 5 * 5 + 114

 sum += 4 * 5 + 115

 sum += 3 * 5 + 116

 sum += 2 * 5 + 117

 sum += 1 * 5 + 118

 sum += 0 * 5 + 119

 sum += 5 * 5 + 120

 sum += 4 * 5 + 121

 sum += 3 * 5 + 122

 sum += 2 * 5 + 123

 sum += 1 * 5 + 124

 sum += 0 * 5 + 125

 sum += 5 * 5 + 126

 sum += 4 * 5 + 127

 sum += 3 * 5 + 128

 sum += 2 * 5 + 129

 sum += 1 * 5 + 130

 sum += 0 * 5 + 131

 sum += 5 * 5 + 132

 sum += 4 * 5 + 133

 sum += 3 * 5 + 134

 sum += 2 * 5 + 135

 sum += 1 * 5 + 136

 sum += 0 * 5 + 137

 sum += 5 * 5 + 138

 sum +=

$$\textcircled{A} \quad i=0 \text{ to } 4 \Rightarrow \quad i=0, \quad \text{num} = i * 5 + j$$

$$j=0 \text{ to } 4$$

$$\stackrel{i=0}{\Rightarrow} \cancel{0 \times 5 + 0} + \cancel{0 \times 5 + 1} + \cancel{0 \times 5 + 2} + \cancel{0 \times 5 + 3} + \cancel{0 \times 5 + 4}$$

$$\textcircled{1} \quad 0 + 1 + 2 + 3 + 4$$

$$\begin{array}{c} \cancel{0+1+2+3+4+5+6+7+8+9} \\ \cancel{+10+11+12+13+14} \\ \Rightarrow a_1 = 0 \end{array}$$

$$a_n = ?$$

$$\begin{aligned} \textcircled{2} \quad j=1, \rightarrow \quad & 1 \times 5 + 0 + 1 \times 5 + 1 + 1 \times 5 + 2 + 1 \times 5 + 3 + 1 \times 5 + 4 \\ & \Rightarrow 5 + 0 + 5 + 1 + 5 + 2 + 5 + 3 + 5 + 4 \\ & = 5 + 6 + 7 + 8 + 9 \end{aligned}$$

$$S_n = \frac{n(n+1)}{2}$$

$$\Rightarrow \frac{25(25+1)}{2}$$

$$\begin{aligned} \textcircled{3} \quad i=2, \quad & (2 \times 5 + 0) + (2 \times 5 + 1) + (2 \times 5 + 2) + (2 \times 5 + 3) + (2 \times 5 + 4) \\ & 10 + 11 + 12 + 13 + 14 \end{aligned}$$

$$\Rightarrow \frac{25 \times 26}{2}$$

$$\Rightarrow 25 \times 14$$

$$\Rightarrow 250 + 100$$

$$= \boxed{350}$$