flow of control → Conditional, Loops } DPP

5-6 hrs ⇒ 40%

Questions → 60%

MCQs on

Conditionals in C

What is the correct output of the following code?

```
int a = 5, b = 10;
if (a < b) {
    if (a > 0)
        printf("A");
    else
        printf("B");
} else
    printf("C");
```

a = 5

b = 10

a) A

b) B

c) C

d) No output

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

Single Statement

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

Logical AND

a=7 b=0

a AND B

↓

True && 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;

if (a == 5 && b == 7)
    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)

b) if (a == 5)

c) if (a = 5 && b == 10)

d) if (a == 5 & b == 10)

Assignment

True/false

Assign

Bitwise AND

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.

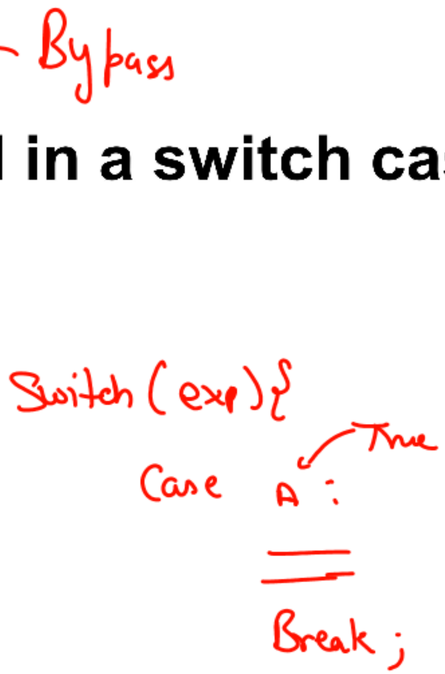
What happens if the break statement is omitted in a switch case?

a) Compilation error

b) Program crashes

c) Case execution continues to the next case

d) Only the matched case is executed



Which of the following causes a runtime error?

a) if (x > 0)

b) if (x < 0)

c) if (x == 0)

d) if (x / 0)

No

No

No

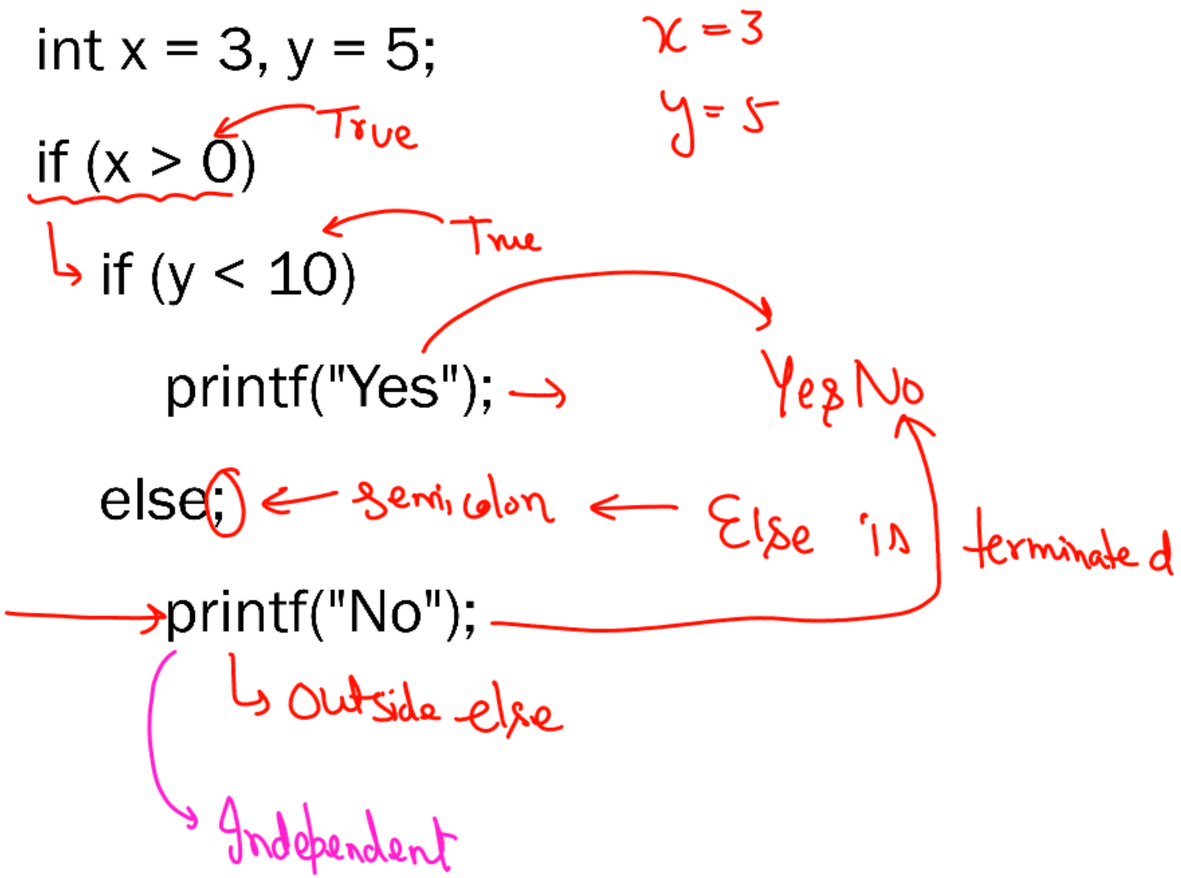
True / false

Error

$\frac{n}{0} \Rightarrow$ Error

What is the output of the following code?

```
int x = 3, y = 5;
if (x > 0)
    if (y < 10)
        printf("Yes");
    else;
printf("No");
```



a) YesNo

b) Yes

c) No

d) Compilation error

What will be the output of the following code?

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

a) 10

b) 5

c) 0

d) Undefined behavior

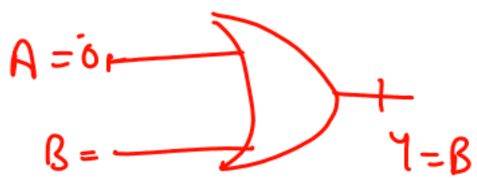
Which statement about short-circuit evaluation in C is correct?

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

a) X 1 9

b) Y 1 9

c) X 2 8

d) Compilation error

What will be the output of the following code?

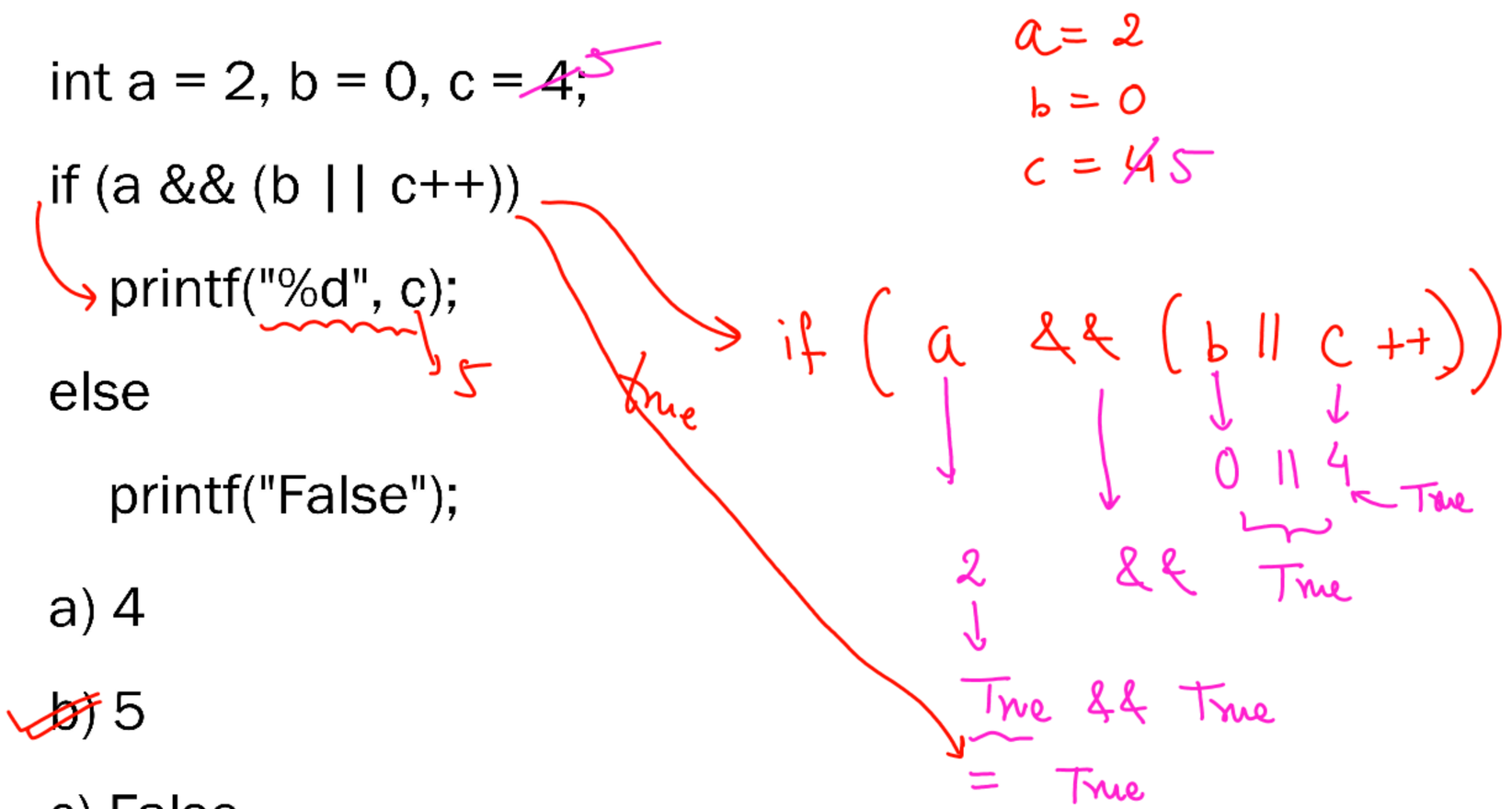
```
int a = 2, b = 0, c = 4;
if (a && (b || c++))
    printf("%d", c);
else
    printf("False");
```

a) 4

b) 5

c) False

d) Compilation error



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

-1

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

True OR Skip ← Short Circuit Eval.

True

a) -1

b) 0

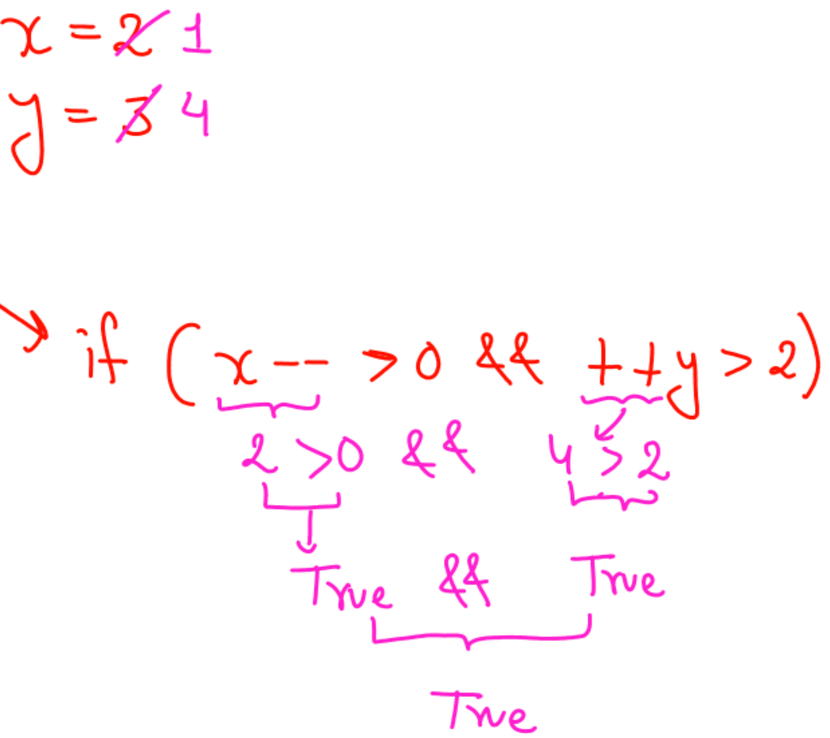
c) Undefined behavior

d) Compilation error

What will the following code print?

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

if (x-- > 0 && ++y > 2)
    printf("X is %d and Y is %d", x, y);
else
    printf("Condition failed");
```

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



o/p X is 1 and Y is 4

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

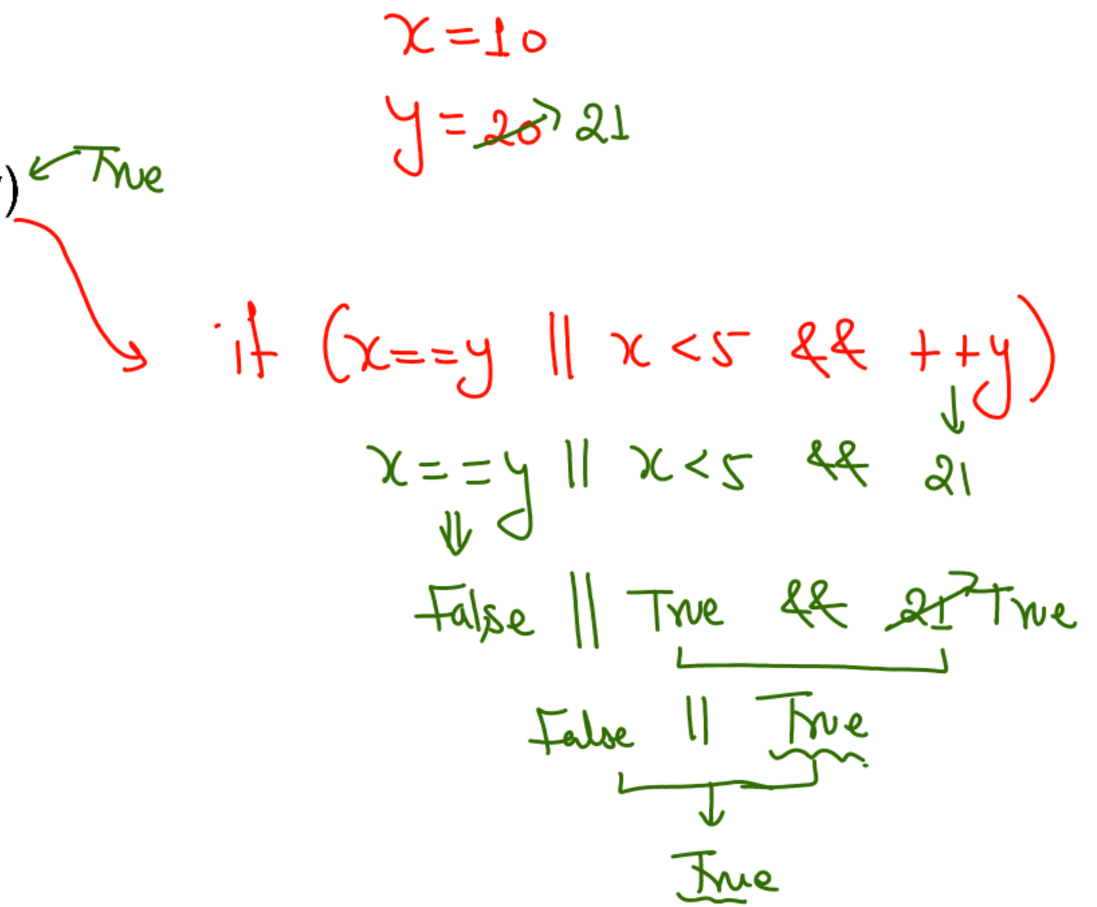
```
int x = 10, y = 20;
if (x == y || x < 5 && ++y)
    printf("Pass");
else
    printf("Fail");
```

a) Fail

b) Pass

c) Undefined behavior

d) Compilation error



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

a) It can evaluate only one expression per branch.

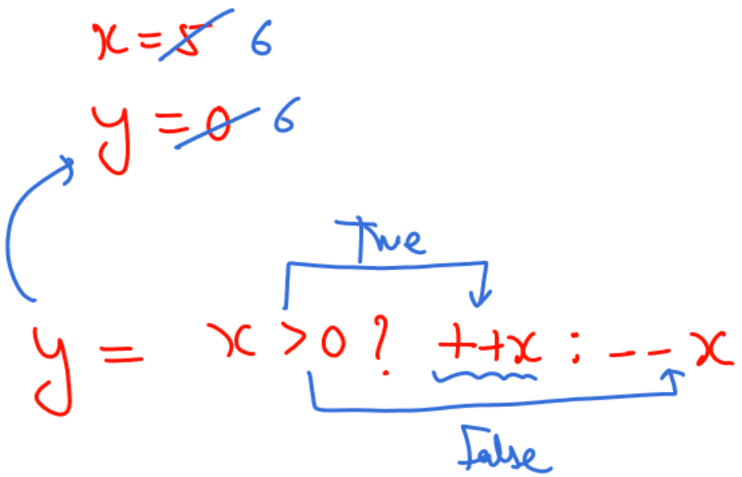
b) The expression on both sides of : must be of the same type.

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)
    if (++b > 10)
        printf("A");
    else
        printf("B");

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

a) A 4 11

b) B 4 11

c) B 4 10

d) A 5 11

What is the output of this code?

```
int x = 5, y = 0;
if (x && y++)
    printf("True");
else
    printf("False");
printf(" %d", y);
```

x = 5
y = 0 → 1

if (x && y++) → false

5 && 0 ⇒ True && False ⇒ False

o/p
False 1

a) False 0

b) True 1

c) False 1

d) Undefined behavior

Which of the following expressions evaluates to true?

a) !(1 || 0) → false

b) !1 && 0 → false

c) !1 || 0 ⇒ 0 OR 0 = False

d) !(1 && 0) → ! (1 && 0)

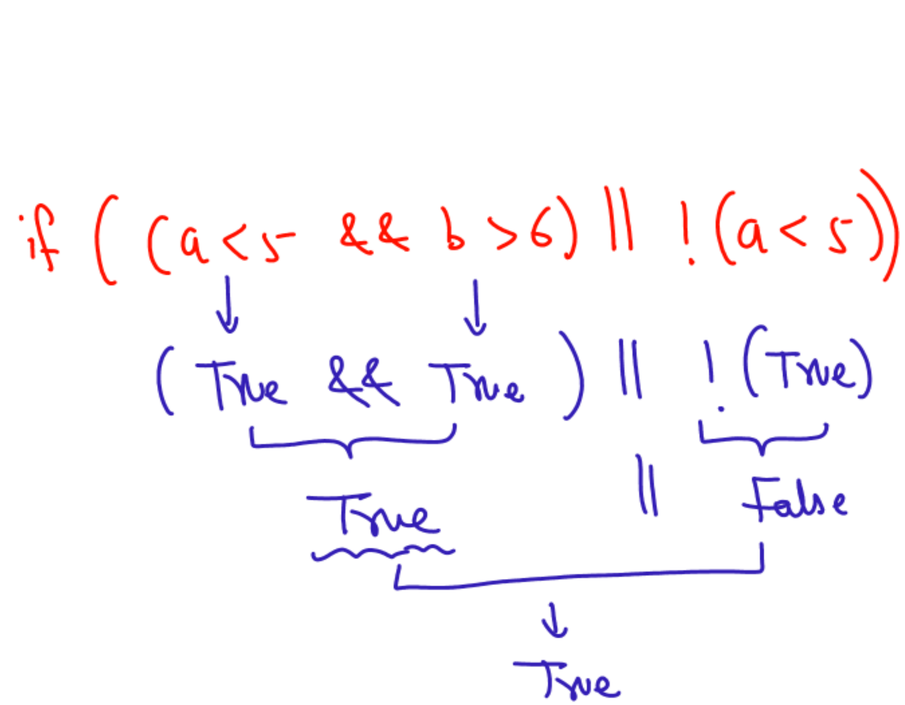
! (0) ⇒ 1

Not (false) = True

What is the output of this code?

```
int a = 4, b = 7;
if ((a < 5 && b > 6) || !(a < 5))
    printf("True");
else
    printf("False");
```

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



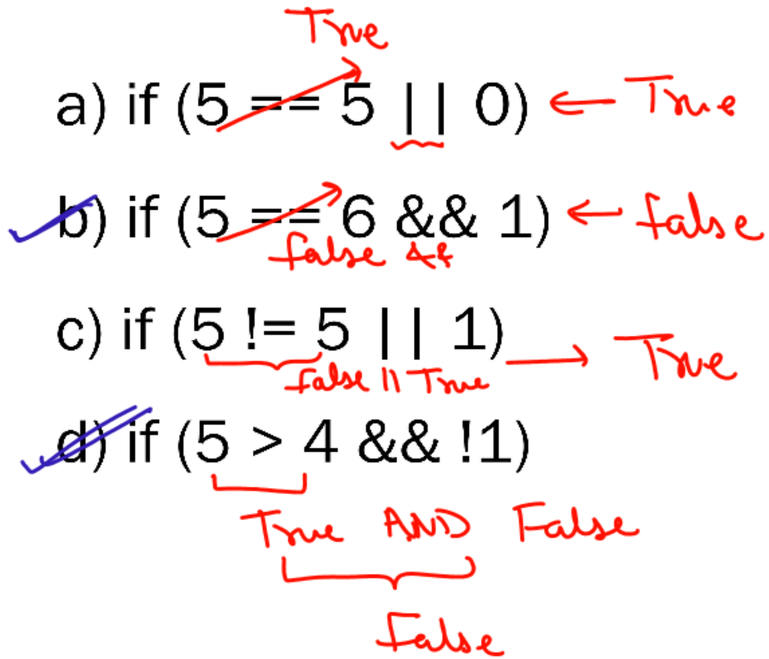
Which of the following will print "False"?

a) if (5 == 5 || 0)

b) if (5 == 6 && 1)

c) if (5 != 5 || 1)

d) if (5 > 4 && !1)



What will happen when the following code is executed?

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

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;

if (x || ++y)
    printf("X is %d and Y is %d", x, y);
else
    printf("False");
```

skip

True OR

x = 10

y = 0

10

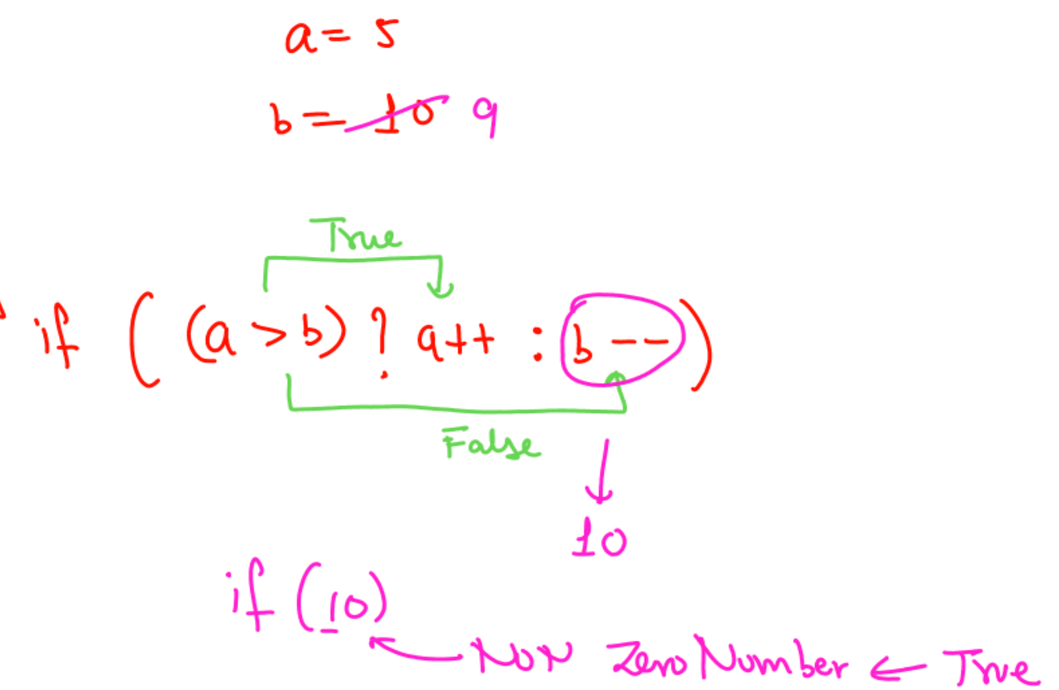
10

0

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--)
    printf("A = %d, B = %d", a, b);
else
    printf("No Change");
```



a) A = 5, B = 9

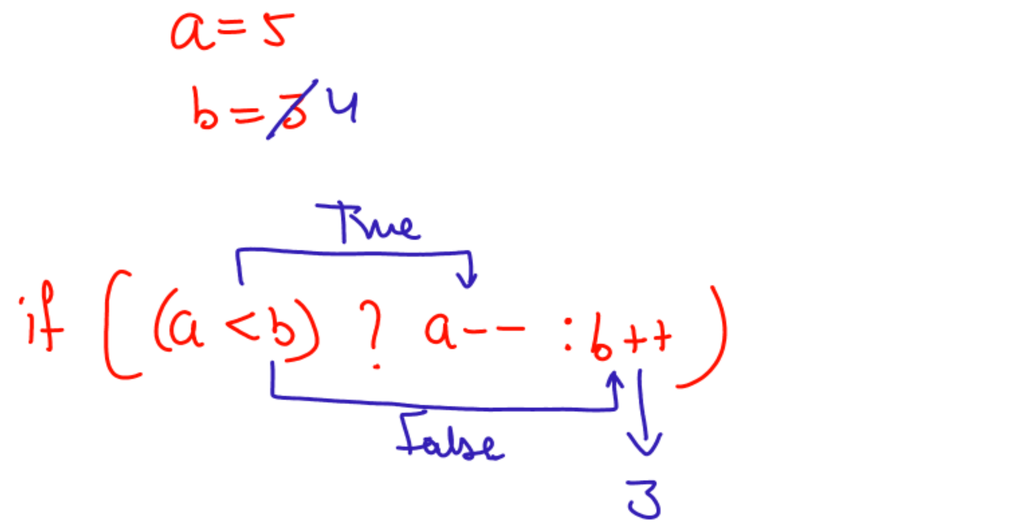
b) A = 6, B = 10

c) A = 6, B = 9

d) No Change

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

Loops in C

What will be the output of the following code?

init Var ⇒ i (0→4) = ~~0~~ ~~1~~ ~~2~~ 3

output: 0 1 2

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

Allowed

Cases

① i=0
i<5 (T)
↳ if (i==3) ⇒ false
i++ ⇒ i=1

② i=1, i<5 (T)
↳ (i==3) (False)
i++ ⇒ i=2

③ i=2, i<5 (T)
↳ i==3 (F)
i++ ⇒ i=3

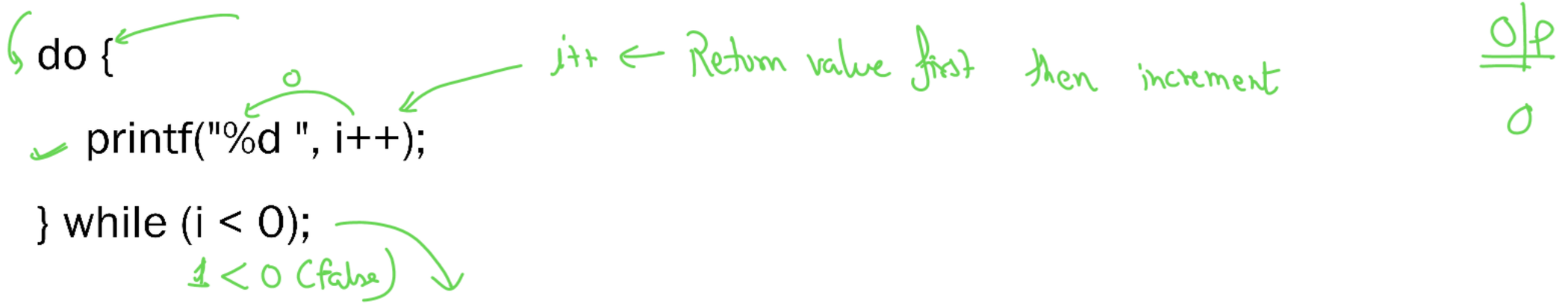
④ i=3, i<5 (T)
↳ i==3 (T)
↳ Break ← terminate

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

What will be the output of the following code?

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

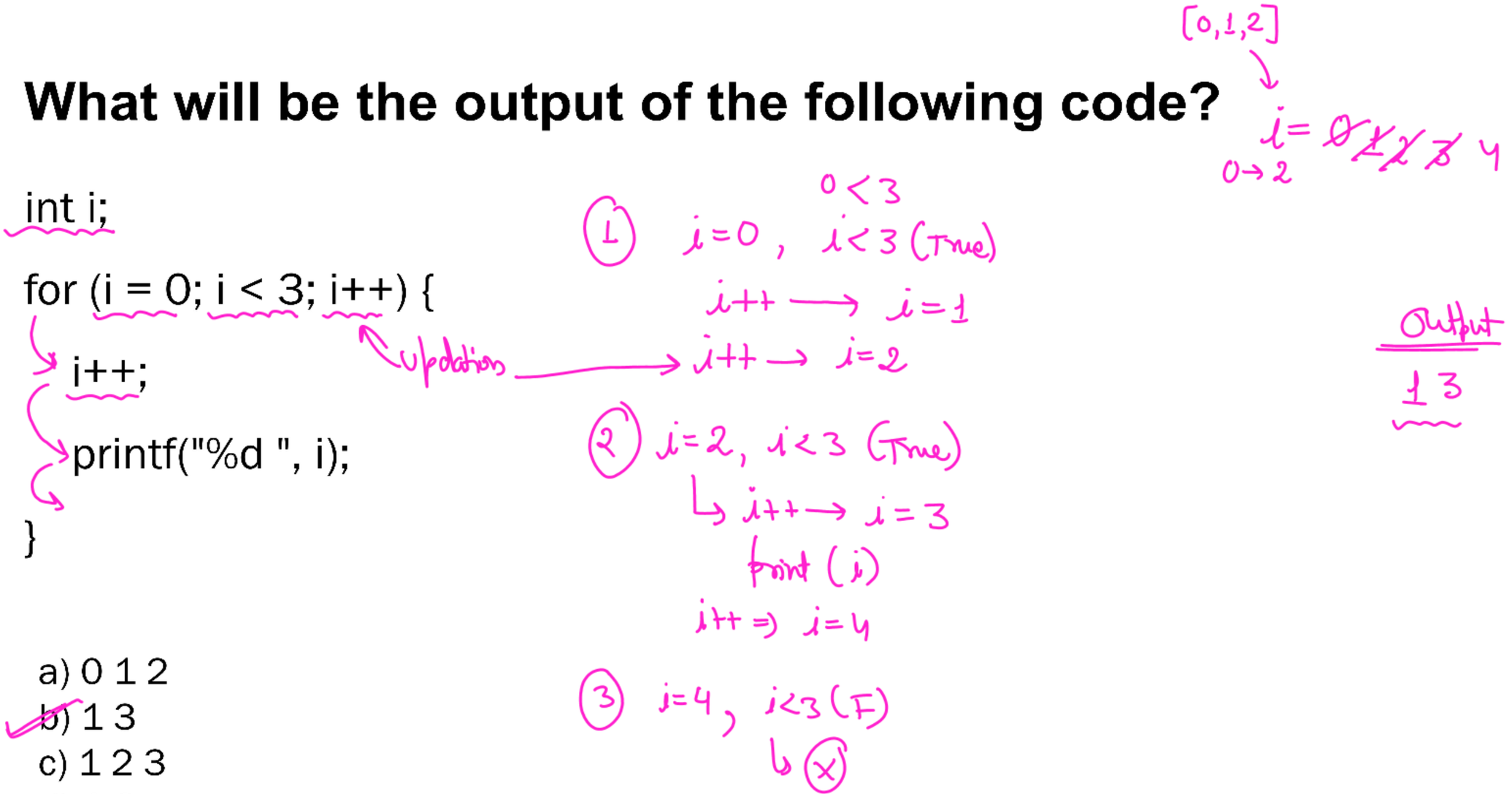
a) 0
b) 0 1
c) 1
d) No output



What will be the output of the following code?

```
int i;

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

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

i<=5 will always true

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

$i = 1 \; 0 \; -1$

① $i = 1$, $i <= 5$ (True)
↳ print(i)
$i--$

② $i = 0$, $i <= 5$ (True)
↳ print(i)
$i--$

③ $i = -1$, $i <= 5$ (True)
$i = -2$ (T)
$i = -3$ (T)
⋮
$i = -\infty$ (T)

output
10

$i = 0 \rightarrow -\infty$
↳

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

i = 1 2 3 4 5 6
1 → 5

① i = 1 , i <= 5 (T)
↳ i % 2 == 0 → Continue
1 % 2 == 0 – false
i++ ⇒ i = 2

② i = 2, i <= 5 (T)
↳ if i % 2 == 0
2 % 2 == 0 (T)
↳ Skip
i++ = i = 3

③ i = 3, 3 <= 5 (T)
↳ if i % 2 == 0
3 % 2 == 0 (F)
i++ ⇒ i = 4

④ i = 4 , i <= 5 (True)
if (i % 2 == 0)
4 % 2 == 0 (True)
↳ Continue (Skip)
i++ ⇒ i = 5

⑤ i = 5, i <= 5 (T)
↳ if (5 % 2 == 0) → false
→ i++ = 6

⑥ i = 6, 6 <= 5 (false)
✗

Output
1 3 5

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

i = 1 2 3 4 7 8 9 10 11
1 → 10

output
1 2 3 4 7 8 9 10

① i=1, i<=10 (True)
↳ if (i==5) X
i++ ⇒ i=2

② i=2, 2<=10 (T)
↳ if (i==5) X
i++ → i=3

③ i=3, 3<10 (T)
↳ if (i==5) X
i++ → i=4

④ i=4, 4<=10 (T)
↳ i==5 X
i++ → i=5

⑤ i=5, 5≤10 (T)
↳ if(i==5) → True
↳ i+=2
j = i+2
i = 5+2=7
i=7
i++ ⇒ i=8

⑥ i=8; 8≤10 (T)
↳ i++ ⇒ i=9

⑦ j=9, 9≤10 (T)
↳ i++ → i=10

⑧ i=10, 10<=10 (T)
↳ i++ → i=11

⑨ i=11, 11<=10 (false)
↓
Break

What is the output of this code?

[Any Non Zero number = True
and zero = false]

```
int i = 5;
while (i--) {
    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

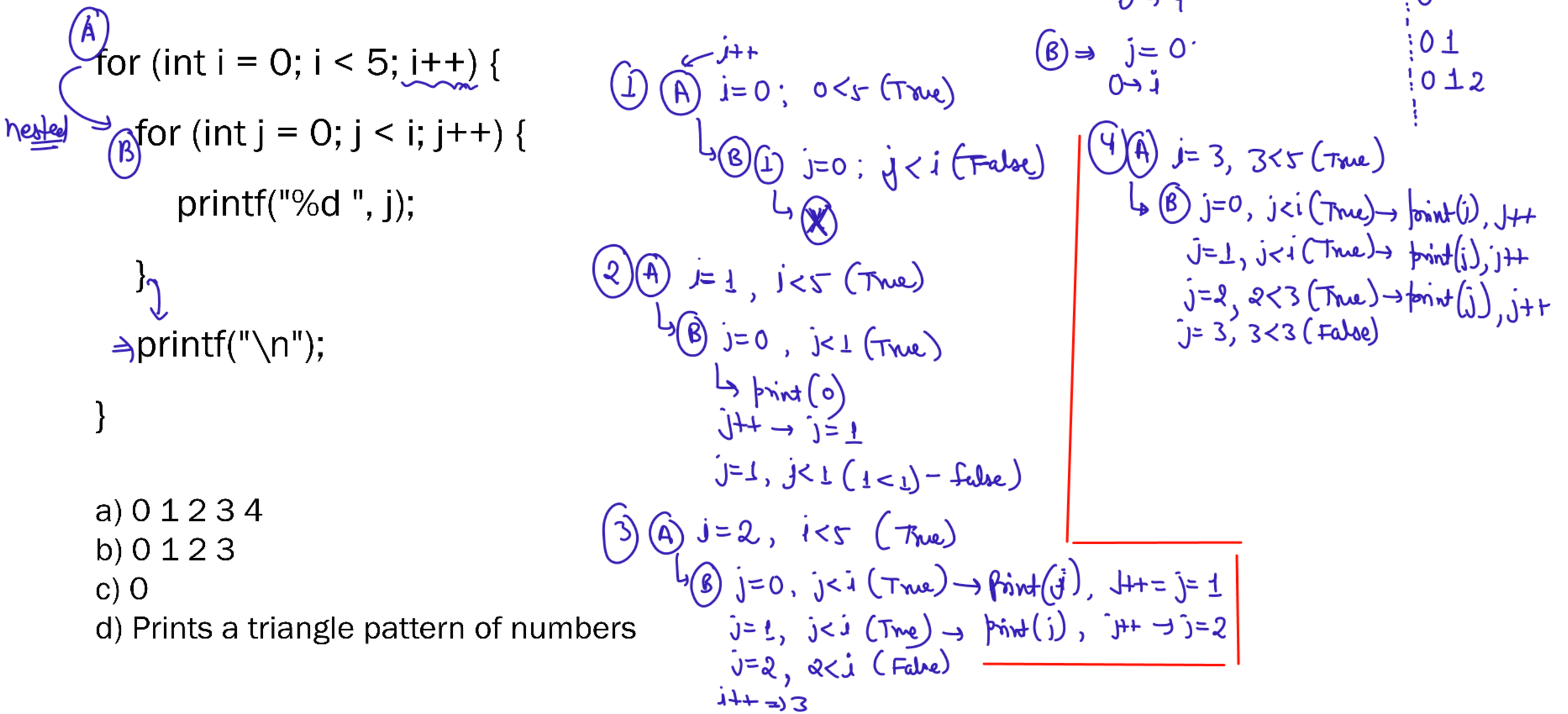
output

4 3 2 1 0

What will be the output of the following code?

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



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

$i = 1 \to 3$
$1 \to 5$

① $i = 1, 1 < 5$ (T)
↳ $i \% 3 == 0$ (false)
$i += 2 \Rightarrow i = i + 2$
$i = 1 + 2 = 3$

② $i = 3, 3 < 5$ (T)
↳ if ($i \% 3 == 0$) ← $\frac{3}{3} \Rightarrow R = 0$
↳ True
↳ Break

Output
1 3

What will be the output of the following code?

```
for (int i = 0; i < 3; i++) {
    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~~

1 0
2 0
2 1

A: i = 0 → 2 ~~1~~ ~~2~~ ~~3~~
B: j = 0 → 1

Output
1 0
2 0
2 1

1) A: i=0, 0<3 (T)
   B: j=0, 0<2 (T) → (i==j) → Yes → Break
   i++ ⇒ i=1

2) A: i=1, 1<3 (T)
   B: j=0, j<2 (T) → (i==j) false
   j++
   j=1, 1<2 (T) → (i==j) (T) → Break
   i++

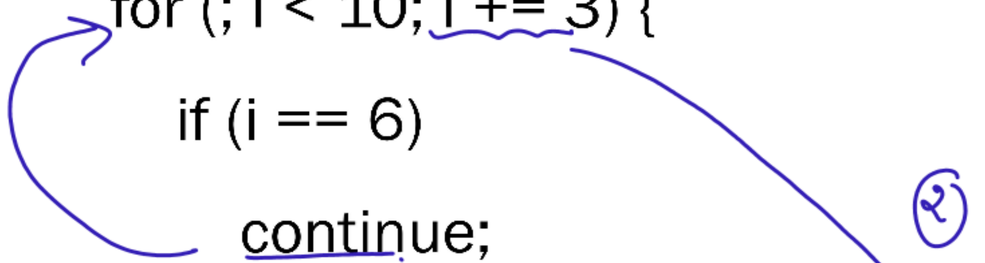
3) A: i=2, 2<3 (True)
   B: j=0, j<i (True) ⇒ (2==0) - false, j++
   j=1, 1<i (True) ⇒ (2==1) → false, j++
   j=2, 2<i (False)
   i++

4) j=3, 3<3 (false)
   ↳ Terminate

What will be printed by the following code?

```
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



i = 0 3 6 9
0 → 9

① i=0, 0<10 (True)
↳ i==6 → false
i+=3 ⇒ i = i+3
i = 0+3 = 3

② i=3, 3<10 (True)
↳ (i==6) ← false
i+=3 ⇒ i = i+3
i = 3+3 = 6

③ i=6, 6<10 → True
↳ (i==6) — True
↳ Continue
i+=3, i = i+3
i = 6+3 = 9
i=9

④ i=9, 9<10 (True)
↳ i = i+3
i = 9+3 = 12

⑤ i=12 12<10 (false)
↓
Terminate

Output
0 3 9

What will be printed by the following code?

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

A: i = 0 → 1

B: j = 0 → 1

(1) (A) i=0, 0<2 (T)
↳ (B) → j=0, 0<3 (True) → j++
j=1, 1<3 (True) → j++
j=2, 2<3 (True)
↳ j==2 → Break
↳ i++, i=1

(2) (A) i=1, i<2 (True)
↳ (B) j=0; 0<3 (True) → j++
j=1; 1<3 (True)
j=2, 2<3 (True)
↳ j==2 → Break
↳ i++, i=3

(3) (A) i=3, 3<3
↓
False
↓
Terminate

Output:
0 0
0 1
1 0
1 1

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 0 \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?

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

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

i = 0 ~~1~~ ~~2~~ 3
j = 0 ~~1~~ ~~2~~ 3

Output
0 2 4

① i = 0
j = 0
i+j < 5 (True)

② i = 1, j = 1
i+j < 5
2 < 5 ← True
↳ print(i+j)

③ i = 2, j = 2
i+j < 5 ⇒ 2+2 < 5 ⇒ 4 < 5 (True)
↳ print(i+j)

④ i = 3, j = 3
i+j < 5 ⇒ 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

i = 2 3 4 5 6
j = 0 3 7 12

(1) i++ < 5
↓
2 < 5 ⇒ True
↳ j += i ⇒ j = j + i
j = 0 + 3 = 3

(2) i = 3, j = 3
(i++ < 5)
(3 < 5) → True
↳ j += i ⇒ j = j + i
j = 3 + 4
j = 7

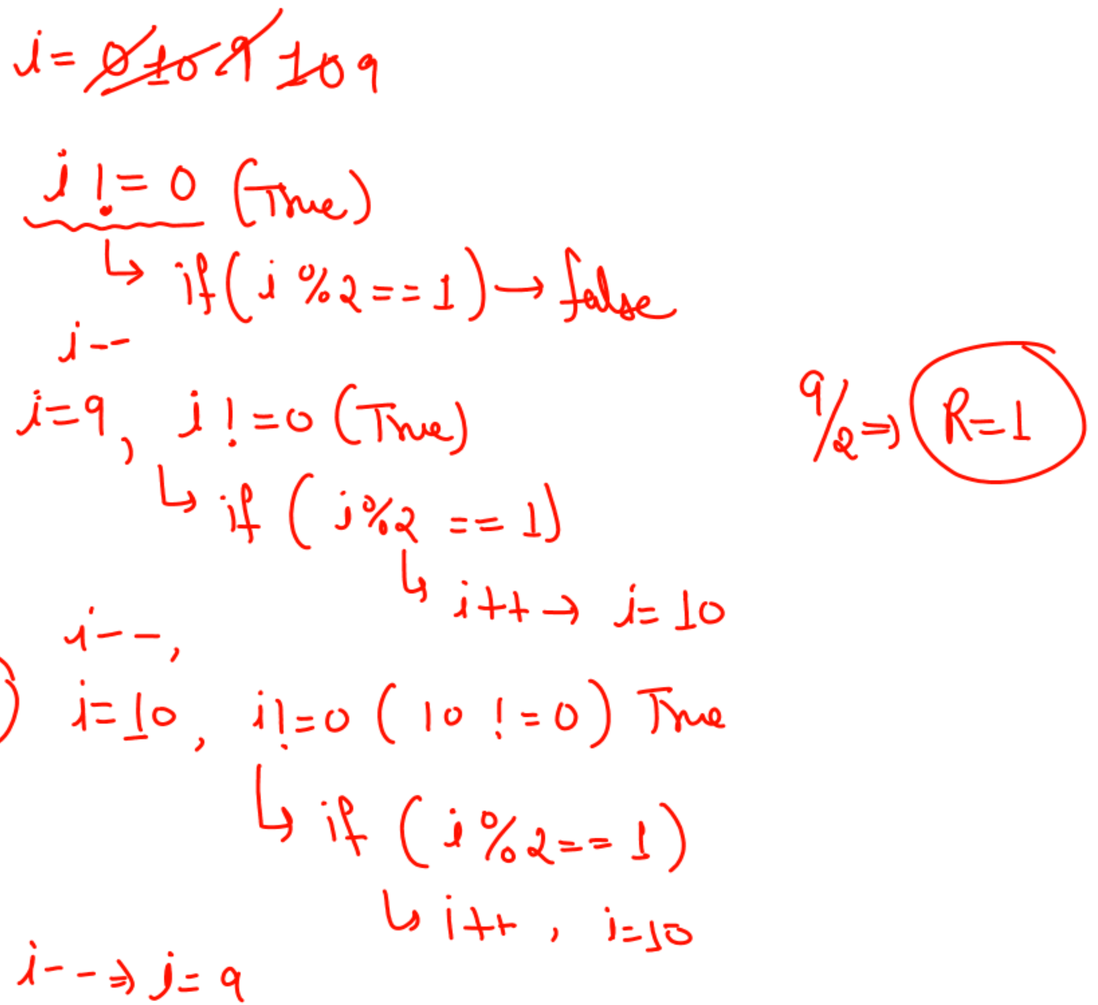
(3) (i++ < 5)
↓
(4 < 5) ← True
↳ j = j + i
j = 7 + 5 ⇒ 12

(4) (i++ < 5)
↓
(5 < 5) → False
↓
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



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

b) 2 1 3 0 4 -1

c) 2 1 3 0

d) 2 1 3 0 4

i = ~~1~~ ~~2~~ ~~3~~ 4

j = ~~2~~ ~~1~~ ~~0~~ -1

Output

2 1

3 0

1) i=1, j=2

(i++ < 5 && j--)

1 < 5 && 2

True && True = True

2) i=2, j=1

(i++ < 5 && j--)

2 < 5 && 1

True && True = True

print (i, j)

3) i=3, j=0

(i++ < 5 && j--)

3 < 5 && 0

True && false

= False

Terminate

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

Output
2 4 6

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

1) i = 1
(i++ <= 5)
(1 <= 5) ⇒ True
i++

2) i = 3
(i++ <= 5)
(3 <= 5) ⇒ True
i++

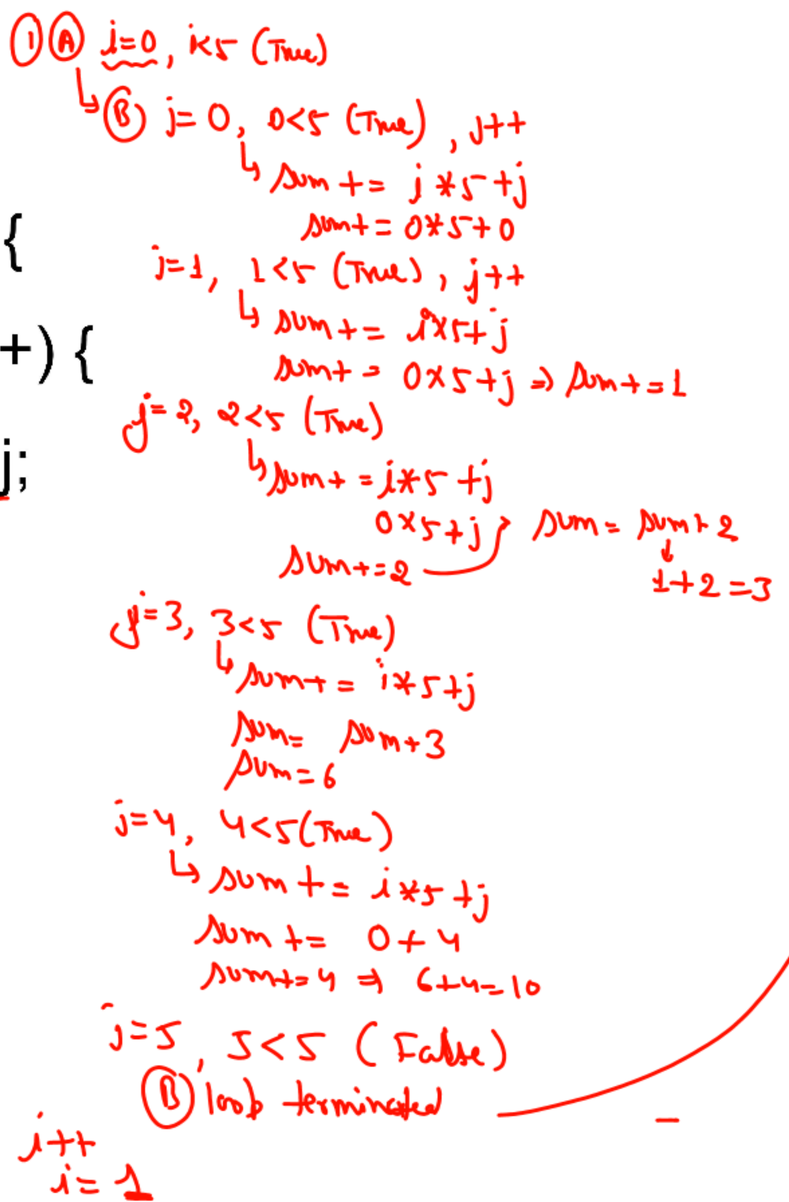
3) i = 5
(i++ <= 5)
(5 <= 5) ⇒ True
i++

4) i = 7
(i++ <= 5)
(7 <= 5) ← 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



(A) $i \Rightarrow 0$ to $4 \Rightarrow$ $i=0$, $sum += i*5+j$

$j \Rightarrow 0$ to 4

$i \equiv 0$

$\Rightarrow$ ~~$0\times5+0$~~ + ~~$0\times5+1$~~ + ~~$0\times5+2$~~ + ~~$0\times5+3$~~ + ~~$0\times5+4$~~

(1) $0+1+2+3+4$

(2) $i=1, \rightarrow$ $1\times5+0 + 1\times5+1 + 1\times5+2 + 1\times5+3 + 1\times5+4$

$\Rightarrow 5+0 + 5+1 + 5+2 + 5+3 + 5+4$

$= 5+6+7+8+9$

(3) $i=2,$ $(2\times5+0) + (2\times5+1) + (2\times5+2) + (2\times5+3) + (2\times5+4)$

$10 + 11 + 12 + 13 + 14$

$0+1+2+3+4+5+6+7+8+9$
$+10+11+12+13+14$

$\Rightarrow a_1 = 0$

$a_n = a_5$

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

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

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

$\Rightarrow 25 \times 14$

$\Rightarrow 250 + 100$

$= \boxed{350}$