Section 1

Snippet 1

```
public class InfiniteForLoop {
public static void main(String[] args) {
for (int i = 0; i < 10; i--) {
 System.out.println(i);
 }
 }
}
```

Error to investigate: Why does this loop run infinitely? How should the loop control variable be adjusted?

The loop runs infinitely as in the condition I < 10 I will never be greater or equal to zero so it will be incremented by i++

Correct Code

```
public class InfiniteForLoop{
 public static void main(String[] args) {
 for (int i = 0; i < 10; i++) {
 System.out.println(i);
}
}
}
```

Snippet 2

```
public class IncorrectWhileCondition {
 public static void main(String[] args) {
int count = 5;
while (count = 0) {
 System.out.println(count);
 count--;
 }
 }
 }
```

Error to investigate: Why does the loop not execute as expected? What is the issue with the condition in the `while` loop?

Error : incompatible types .In condition (count=0) assignment operator is used rather than Comparison Op.

Correct code

```
public class IncorrectWhileCondition {
    public static void main(String[] args) {
        int count = 5;
        while (count > 0) {
            System.out.println(count);
            count--;
        }
    }
}
```

Snippet 3

```
public class DoWhileIncorrectCondition {
 public static void main(String[] args) {
 int num = 0;
 do {
 System.out.println(num);
 num++;
 }
 while (num > 0 ); //
}
}
```

Error to investigate: Why does the loop only execute once? What is wrong with the loop condition in the `do while` loop?

Num is incremented to 1 but the condition num>0 is false as loop terminate, hence loop will be print only once

Correct Code

```
public class DoWhileIncorrectCondition {
 public static void main(String[] args) {
 int num = 1;
 do {
 System.out.println(num);// 1 2345678910
 num++; // 2345678910
 }
 while (num <= 10); //
}
}
```

Snippet 4

```
public class OffByOneErrorForLoop {
public static void main(String[] args) {
 for (int i = 1; i < 10; i++) {
System.out.println(i);
}
 Expected: 10 iterations with numbers 1 to 10 //
Actual: Prints numbers 1 to 10, but the task expected only 1 to 9 } }
```

// Error to investigate: What is the issue with the loop boundaries? How should the loop be adjusted to meet the expected output?

Change loop condition that is i<=10 to i<10 that means it will print the value less than 10 not equal to 10…It will print from 1 to 9

Correct Code

```
public class OffByOneErrorForLoop {
public static void main(String[] args) {
for (int i = 1; i < 10; i++) {
 System.out.println(i);
 }
```

Snippet 5

```
public class WrongInitializationForLoop {
public static void main(String[] args) {
 for (int i = 10; i >= 0; i++) {
 System.out.println(i);
}
 }
 }
```

Error to investigate: Why does this loop not print numbers in the expected order? What is the problem with the initialization and update statements in the `for` loop?

As the the I will be always greater than zero and there condition is also i++ so it will cause a infinity loop

Correct Code

```
public class WrongInitializationForLoop {
   public static void main(String[] args) {
      for (int i = 10; i >= 0; i--) {
         System.out.println(i); // 10 9 8 7 6 5 4 3 2 1 0
      }
   }
}
```

Snippet 6

```
public class MisplacedForLoopBody {
 public static void main(String[] args) {
 for (int i = 0; i < 5; i++)
 System.out.println(i);
System.out.println("Done");
}
 }
```

Error to investigate: Why does "Done" print only once, outside the loop? How should the loop body be enclosed to include all statements within the loop?

ANS: System.out.println executes one time…not a part of loop

Enclose both sop(i) and sop(“Done”) within brackets ‘{‘

Correct Code

```
public class MisplacedForLoopBody {
public static void main(String[] args) {
for (int i = 0; i < 5; i++){
 System.out.println(i);
 System.out.println("Done");
}
}
 }
```

Snippet 7

```
public class UninitializedWhileLoop {
public static void main(String[] args) {
int count;
 while (count < 10) {
System.out.println(count);
 count++;
}
 }
 }
```

Error to investigate: Why does this code produce a compilation error? What needs to be done to initialize the loop variable properly?

```
Error: UninitializedWhileLoop.java:6: error: variable count might not have been initialized
        while (count < 10) {
```

Correct Code

```
public class UninitializedWhileLoop {
public static void main(String[] args) {
int count=0;
while (count < 10) {
System.out.println(count);
count++;
 }
}
}
```

Snippet 8

```
public class OffByOneDoWhileLoop {
 public static void main(String[] args) {
 int num = 1;
do { System.out.println(num);
num--;
} while (num > 0);
}
}
```

Error to investigate: Why does this loop print unexpected numbers? What adjustments are needed to print the numbers from 1 to 5?

Correct Code

```
public class OffByOneDoWhileLoop {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.println(num);
            num++;
        } while (num <= 5);
    }
}
```

Snippet 9

```
public class InfiniteForLoopUpdate {
 public static void main(String[] args) {
for (int i = 0; i < 5; i += 2) {
 System.out.println(i);
 }
 }
}
```

Error to investigate: Why does the loop print unexpected results or run infinitely? How should the loop update expression be corrected?

i was incremented by 2 that why the result was 0,2,4 and 1and 3 excuded,if we want all sequence wise numbers to be print then increment I by one.

Correct code:

```
public class InfiniteForLoopUpdate {
    public static void main(String[] args) {
        for (int i = 0; i < 5; i++) {
            System.out.println(i);
        }
    }
}
```

Snippet 10

```
public class IncorrectWhileLoopControl {
public static void main(String[] args) {
int num = 10;
 while (num = 10) {
System.out.println(num);
 num--;
}
 }
}
```

Why does the loop execute indefinitely? What is wrong with the loop condition?

Error: IncorrectWhileLoopControl.java:4: error: incompatible types: int cannot be converted to Boolean

Correct Code

```
public class IncorrectWhileLoopControl {
    public static void main(String[] args) {
        int num = 10;
        while (num >= 0) {
            System.out.println(num);
            num--;
        }
    }
}
```

Snippet 11

```
public class IncorrectLoopUpdate {
 public static void main(String[] args) {
 int i = 0;
 while (i < 5) {
 System.out.println(i);
 i += 2;
```

Error: This may cause unexpected results in output } } }

// Error to investigate:

What will be the output of this loop? How should the loop variable be updated to achieve the desired result?

Correct Code

```
public class IncorrectLoopUpdate {
    public static void main(String[] args) {
        int i = 0;
        while (i < 5) {
            System.out.println(i);
            i++;
        }
    }
}
```

Snippet 12

```
public class LoopVariableScope {
public static void main(String[] args) {
for (int i = 0; i < 5; i++) {
 int x = i * 2;
 }
System.out.println(x);
```

Error: 'x' is not accessible here } } // Error to investigate: Why does the variable 'x' cause a compilation error? How does scope

```
Error : LoopVariableScope.java:6: error: cannot find symbol
        System.out.println(x); // Error: 'x' is not accessible here
                           ^
  symbol:   variable x
  location: class LoopVariableScope
```

Correct Code

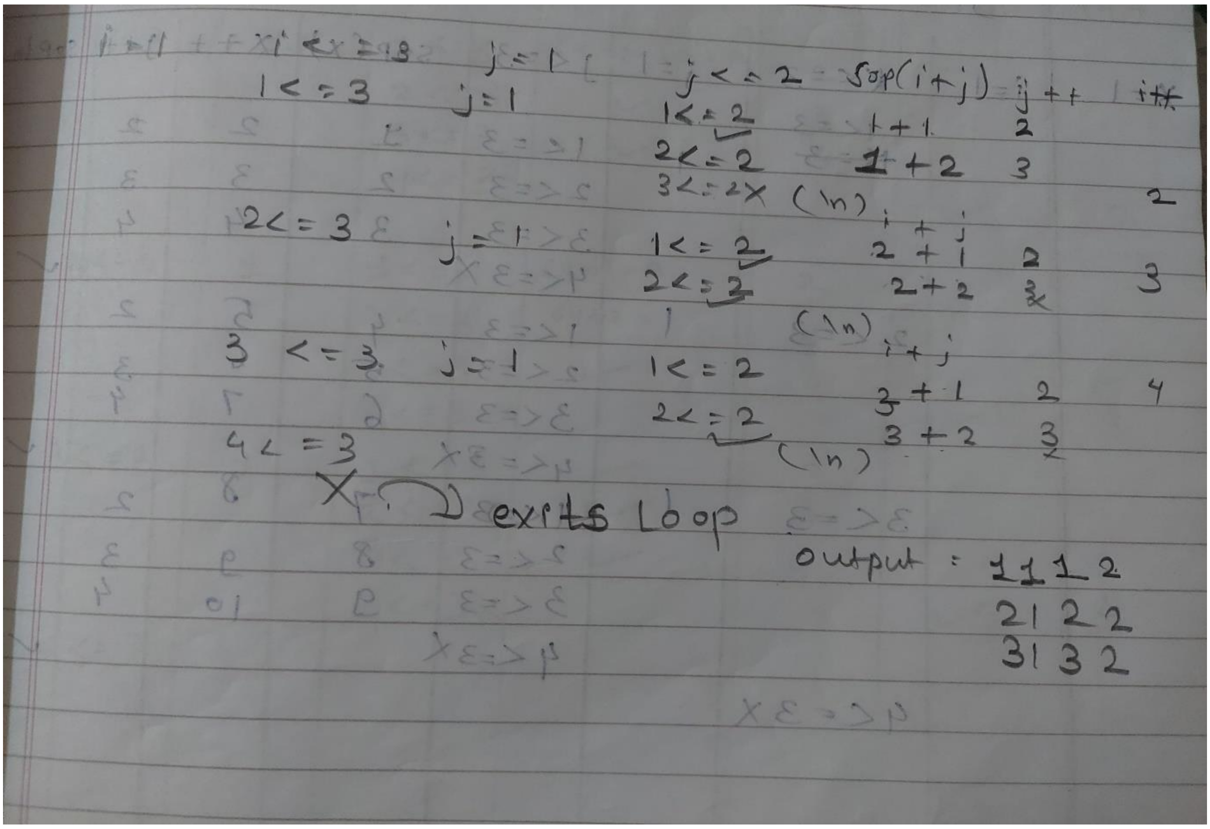
```
public class LoopVariableScope {
    public static void main(String[] args) {
        int x = 0;
        for (int i = 0; i < 5; i++) {
            x = i * 2;
        }
        System.out.println(x); // Error: 'x' is not accessible here
    }
}
```

SECTION 2: Guess the Output

Snippet 1

```
public class NestedLoopOutput {
public static void main(String[] args) {
for (int i = 1; i <= 3; i++) {
for (int j = 1; j <= 2; j++) {
System.out.print(i + " " + j + " ");
}
System.out.println();
}
 }
}
```

// Guess the output of this nested loop



	j = 1	j <= 2	Sop(i+j)	j ++	i++
1<=3	j=1	1<=2	1+1	2	
		2<=2	1+2	3	
		3<=2X	(\n)		2
2<= 3	j =	1<= 2	i + j 2 + 1	2	
		2<=2	2+2	3	3
			(\n)		
3 <= 3	j = 1	1<= 2	i+j 3 + 1	2	4
		2<=2	3 + 2	3	
			(\n)		
4<=3					

X → exits Loop

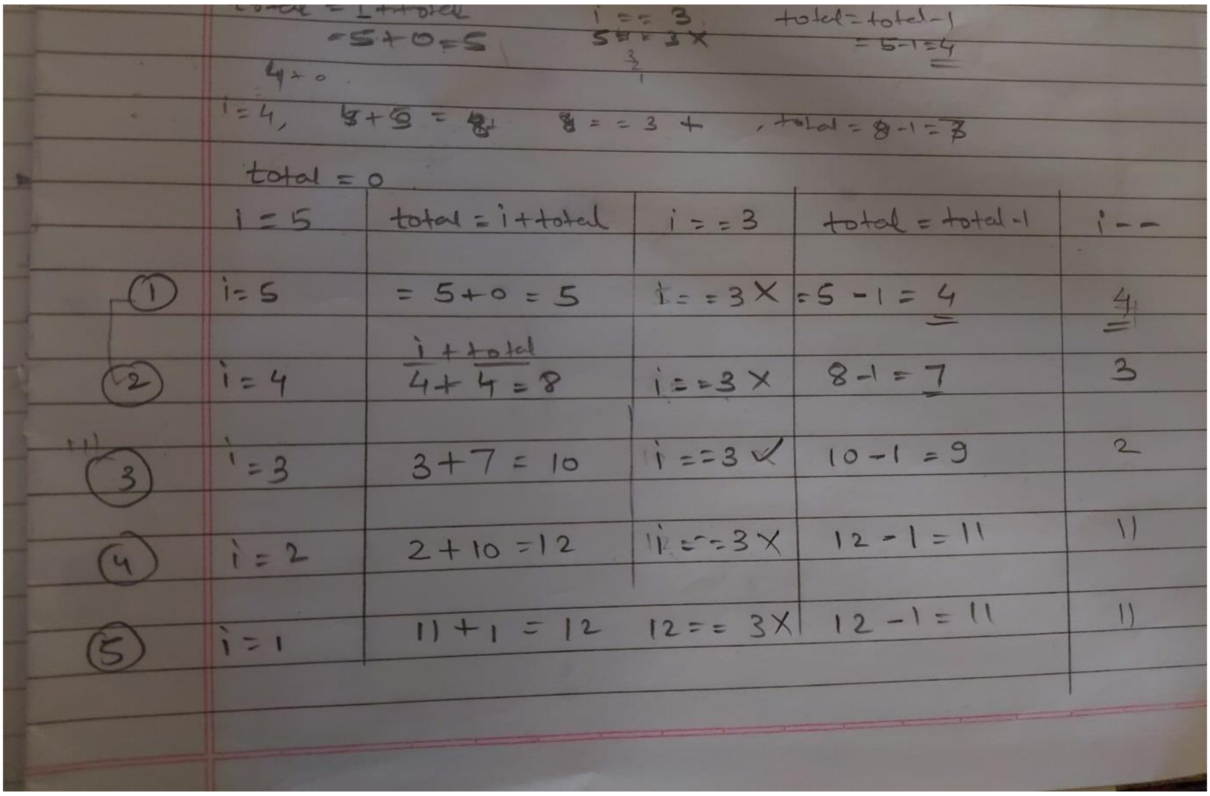
output : 1112
2122
3132

Snippet 2

```
public class DecrementingLoop {
public static void main(String[] args) {
int total = 0;
 for (int i = 5; i > 0; i--) {
 total += i;
if (i == 3) continue;
total -= 1;
```

```
 }
 System.out.println(total);
}
 }
```

// Guess the output of this loop



total = i+total
=5+0=5
4+0
i == 3
5 == 3 ×
total=total-1
=5-1=4

i=4, 3+5 = 8 8 == 3 × , total = 8-1 = 7

total = 0

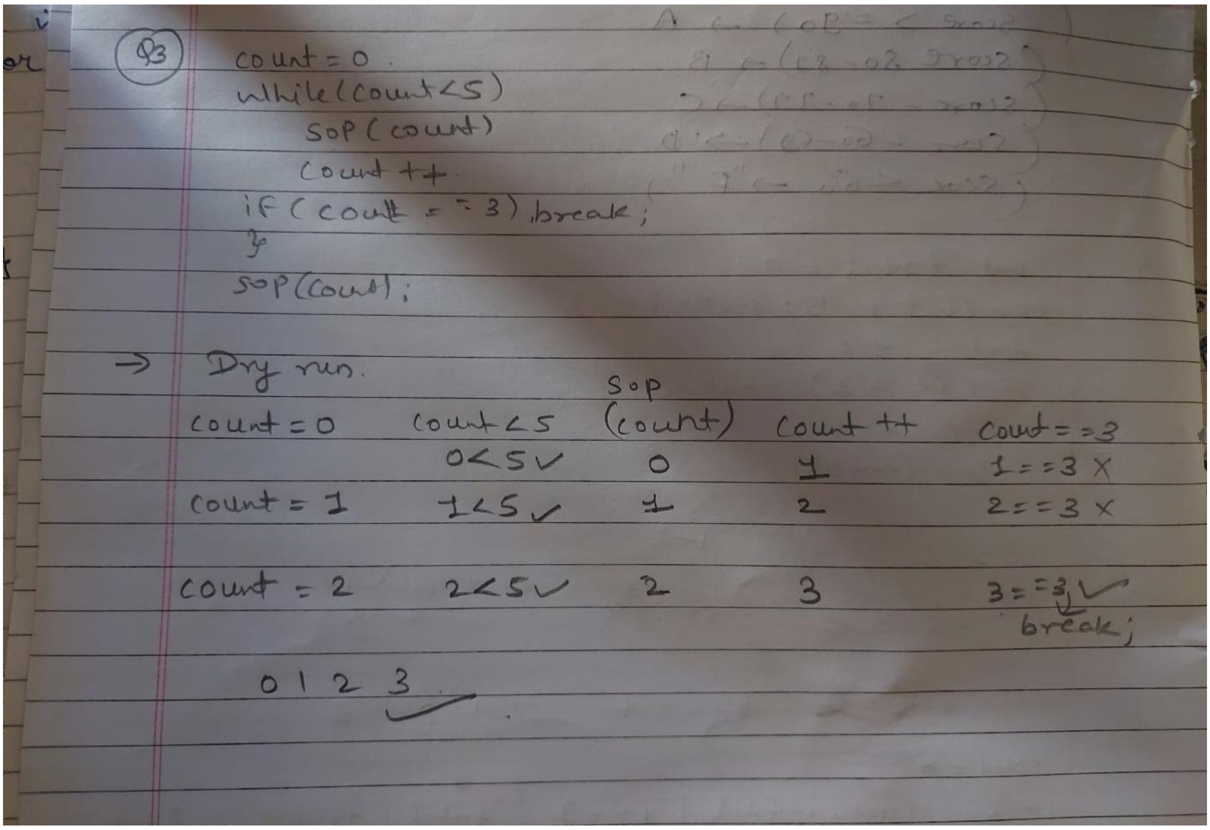
	i = 5	total = i+total	i = = 3	total = total -1	i - -
1	i= 5	= 5+0 = 5	i = = 3 ×	=5 - 1 = 4	4
2	i = 4	i + total 4+ 4 = 8	i = = 3 ×	8 -1 = 7	3
3	i = 3	3+7 = 10	i = = 3 ✓	10 - 1 = 9	2
4	i = 2	2 + 10 = 12	i = = 3 ×	12 - 1 = 11	1)
5	i = 1	11 + 1 = 12	12 = = 3 ×	12 - 1 = 11	1)

Snippet 3

```
public class WhileLoopBreak {
public static void main(String[] args) {
int count = 0;
 while (count < 5) {
System.out.println(count + " ");
count++;
```

```
 if (count == 3)
 break;
}
 System.out.println(count);
 }
}
```

// Guess the output of this while loop



Q3

```
count = 0
while(count<5)
    sop(count)
    count ++
if(count == 3) break;
}
sop(count);
```

→ Dry run.

count = 0	count<5	Sop (count)	count ++	count==3
	0<5 ✓	0	1	1==3 ✗
count = 1	1<5 ✓	1	2	2==3 ✗
count = 2	2<5 ✓	2	3	3==3 ✓ break;

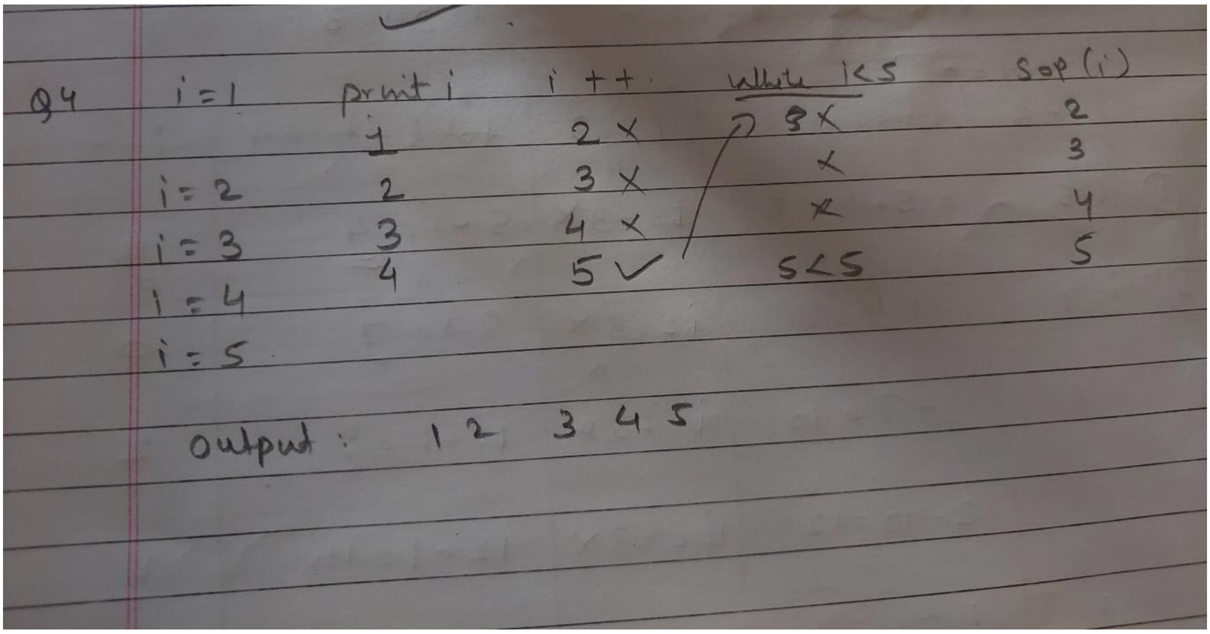
0 1 2 3 ✓

Snippet 4

```
public class DoWhileLoop {
public static void main(String[] args) {
int i = 1;
```

```
 do {
 System.out.print(i + " ");
i++;
}
while (i < 5);
System.out.println(i);
 }
}
```

// Guess the output of this do-while loop



Q4	i=1	print i	i ++	While i<5	Sop (i)
		1	2 ×	3 ×	2
	i = 2	2	3 ×	×	3
	i = 3	3	4 ×	×	4
		4	5 ✓	5<5	5
	i = 4				
	i = 5				

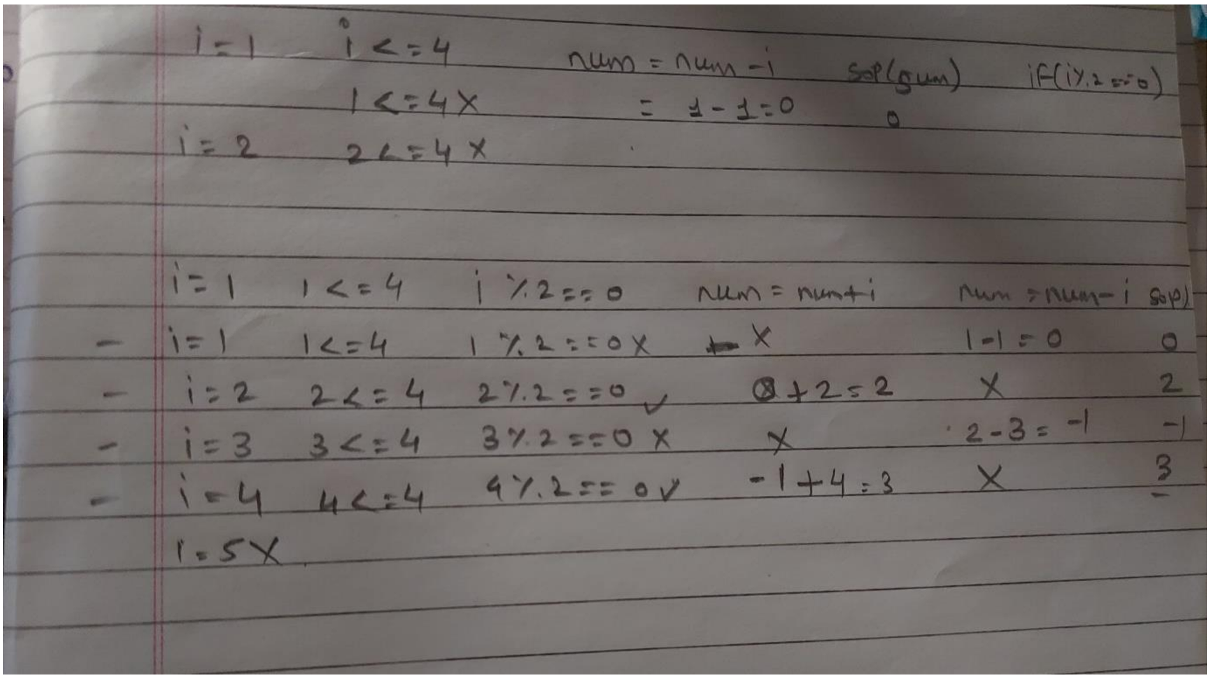
output : 1 2 3 4 5

Snippet 5

```
public class ConditionalLoopOutput {
public static void main(String[] args) {
int num = 1;
 for (int i = 1; i <= 4; i++) {
 if (i % 2 == 0) {
num += i;
```

```
 } else {
num -= i;
 }
 }
System.out.println(num);
}
}
```

// Guess the output of this loop

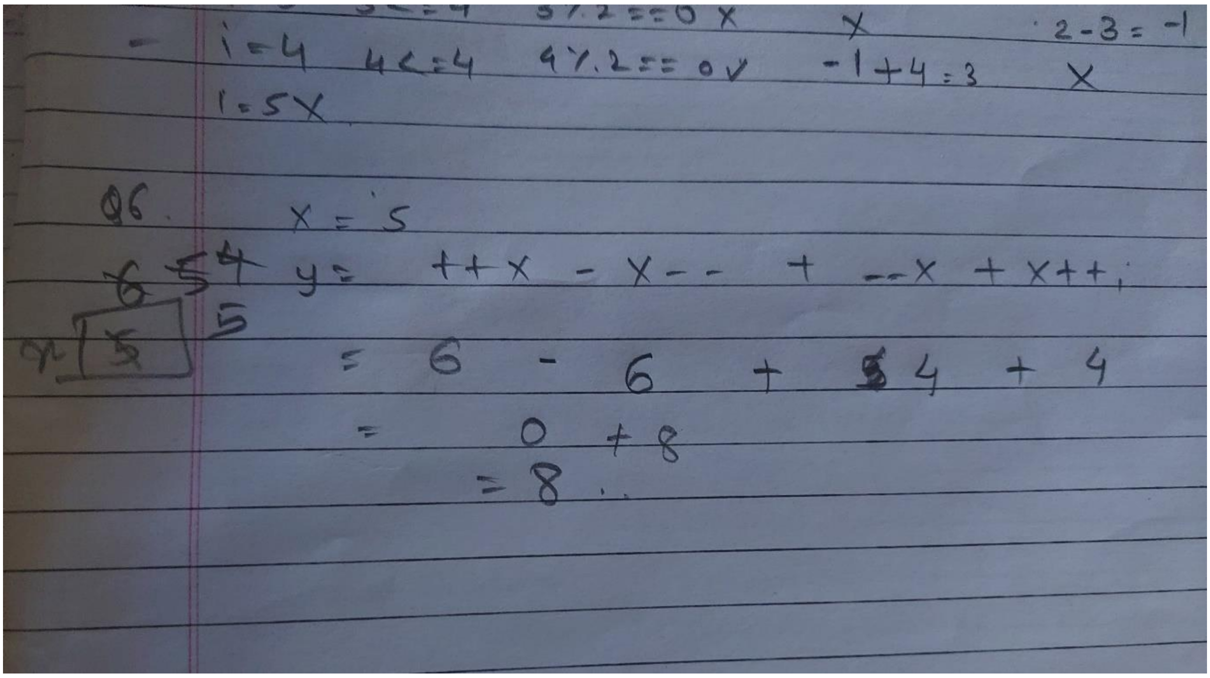


i=1	i<=4	num = num -i	sop(sum)	if(i%2==0)
	1<=4X	= 1-1=0	0	
i=2	2<=4 X			

i=1	1<=4	i %2==0	num = num+i	num = num-i	sop)
i=1	1<=4	1 %2==0 X	X	1-1 = 0	0
i=2	2<=4	2%2==0 ✓	0+2=2	X	2
i=3	3<=4	3%2==0 X	X	2-3= -1	-1
i=4	4<=4	4%2== 0 ✓	-1+4=3	X	3
i=5X					

Snippet 6

```
public class IncrementDecrement {
public static void main(String[] args) {
 int x = 5;
 int y = ++x - x-- + --x + x++;
System.out.println(y);
}
 }
 // Guess the output of this code snippet
```



3<=4   3%2==0 X   X   2-3 = -1

i=4   4<=4   4%2==0 ✓   -1+4=3   X

i=5X

Q6.   x = 5

~~6~~ 5 ~~4~~ y = ++x - x-- + --x + x++;

6   5

x | ~~5~~

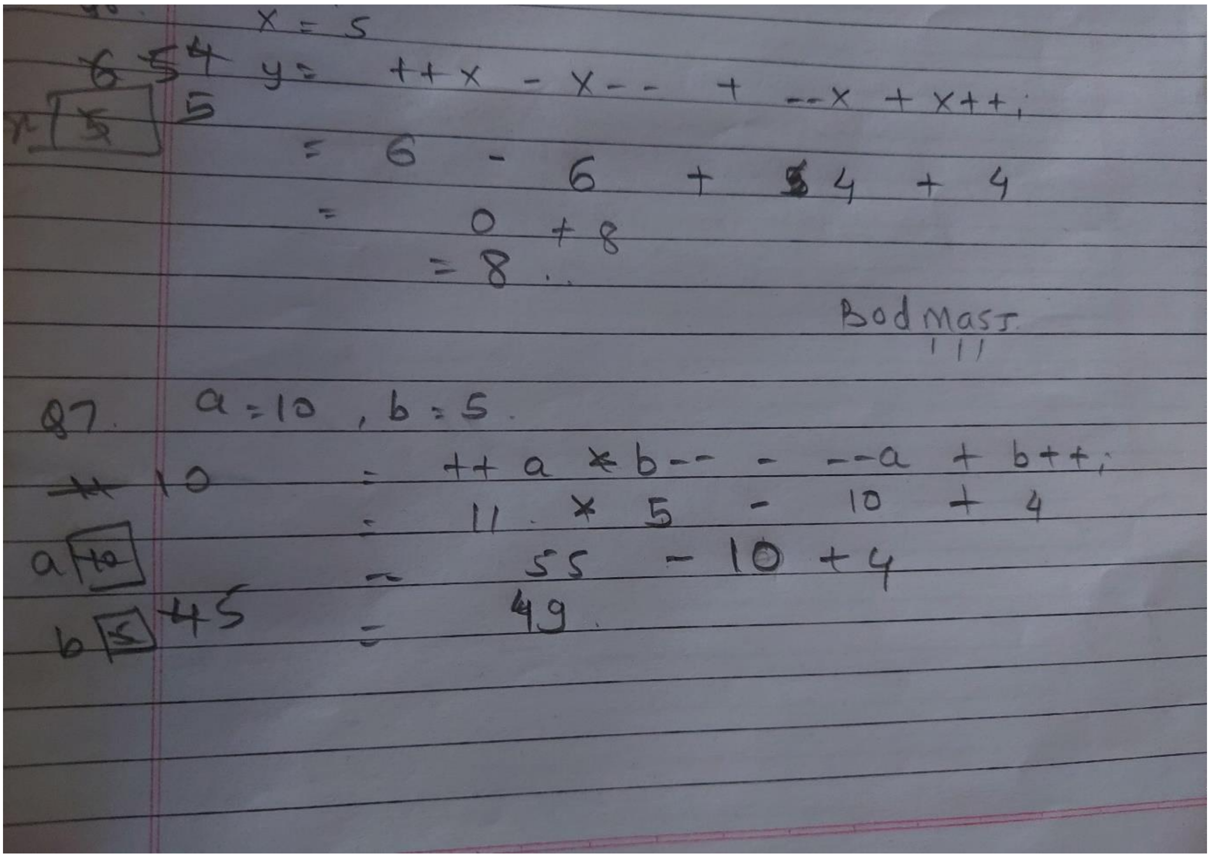
= 6 - 6 + ~~5~~ 4 + 4

= 0 + 8

= 8.

Snippet 7

```
public class NestedIncrement {
 public static void main(String[] args) {
int a = 10;
 int b = 5;
 int result = ++a * b-- - --a + b++;
System.out.println(result);
 }
 }
// Guess the output of this code snippet
```



x = 5

~~6~~ ~~5~~ 4 y = ++x - x-- + --x + x++;

x [~~5~~] 5

= 6 - 6 + ~~5~~ 4 + 4

= 0 + 8

= 8.

Bodmas

Q7. a = 10, b = 5.

~~11~~ 10 = ++a * b-- - --a + b++;

= 11 * 5 - 10 + 4

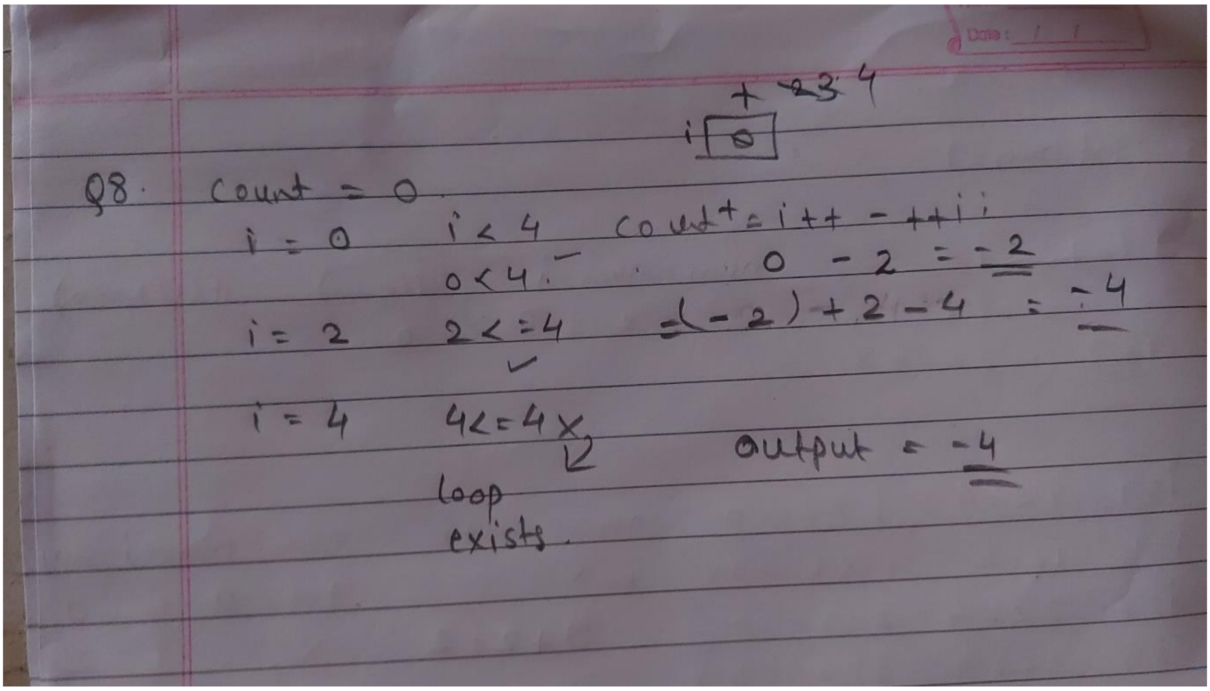
a [~~10~~] = 55 - 10 + 4

b [5] ~~4~~ 5 = 49.

Snippet 8

```
public class LoopIncrement {
public static void main(String[] args) {
int count = 0;
 for (int i = 0; i < 4; i++) {
count += i++ - ++i;
 }
System.out.println(count);
}
 }
```

// Guess the output of this code snippet



Date : / /

~~1~~ ~~2~~ ~~3~~ 4

i $\boxed{\cancel{0}}$

Q8. count = 0

i = 0   i < 4   count += i++ - ++i;

0 < 4.   0 - 2 = -2

i = 2   2 <= 4 ✓   = (-2) + 2 - 4 = -4

i = 4   4 <= 4 ✗

loop exists.

output = -4