

Add One

(Imp)

$$n = 5$$

$$\text{arr} = \begin{array}{|c|c|c|c|c|} \hline 5 & 2 & 3 & 4 & 1 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$$

$$\text{ans} = \begin{array}{|c|c|c|c|c|} \hline 5 & 2 & 3 & 4 & 2 \\ \hline 0 & 1 & 2 & 3 & 4 \\ \hline \end{array}$$

$$\begin{array}{r} 52,34\boxed{9} \\ + 1 \\ \hline 523\boxed{5}0 \\ \hline \end{array}$$

res = 10
ans = res % 10 = 0
carry = res / 10 = 1

Carry = \cancel{x} 1

$0 \rightarrow 8$ $\xrightarrow{\text{rightmost digit}}$ $+1$

$\underline{\underline{9}}$

Ex :- $n=5$
 $\text{arr} = [5, 1, 2, 3, 3]$
 $+ 1$
 $\boxed{\begin{array}{c} \text{arr} \\ | \\ \boxed{5 \ 1 \ 2 \ 3 \ 4} \\ \hline 0 \ 1 \ 2 \ 3 \ 4 \end{array}}$

$i=4$, $\text{ans} = \text{current ele} + 1$
 $= 4$

$i=3$, $\text{ans} = \text{current ele.}$

$i=2$,

$i=1$

$i=0$

Ex 2
 $\text{arr} = [\underline{\underline{2}} \ \underline{\underline{3}} \ \underline{\underline{\frac{9}{2}}} \ \underline{\underline{\frac{9}{3}}} \ \underline{\underline{\frac{9}{4}}}]$
 \downarrow
 $\text{arr} = [\underline{\underline{2}} \ \underline{\underline{4}} \ \underline{\underline{0}} \ \underline{\underline{0}} \ \underline{\underline{0}}]$

$i=4$ if($\text{arr}[i] == 9$)
 $\text{arr}[i] = 0;$

y

$i=3$ if($\text{arr}[i] == 9$)
 $\text{arr}[i] = 0;$

y

$i=2$, $\text{current ele.} == 9$

$i=1$, $\text{Current ele.} == 9$

Ex:-

2	3	4	9	9	9
2	3	5	0	0	0
0	1	2	3	4	5

$i = 5$, current ele == 9 ✓
update current ele with 0

$i = 4$, current ele == 9 ✓
update current ele with 0

$i = 3$, current ele == 9 ✓
update current ele with 0

$i = 2$, current ele == 9 X
update current ele with $\underline{\underline{arr[i]++}}$
and return

gmp

pseudo code

1) traverse (n-1) to 0
1.1) if current ele == 9
 $arr[i] = 0;$
1.2) else
 $arr[i] = arr[i] + 1$
return;

gmp ex:-

$$arr = [\underset{0}{g}, \underset{1}{g}, \underset{2}{g}, \underset{3}{g}] \quad (4)$$

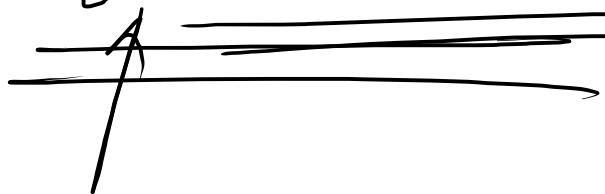
Pseudo code

1) traverse (n-1) to 0

$$arr = [\underset{0}{0}, \underset{1}{0}, \underset{2}{0}, \underset{3}{0}]$$

1.1) if current ele == g
arr[i] = 0;

$$ans = [\underset{1}{1}, \underset{0}{0}, \underset{0}{0}, \underset{0}{0}, \underset{0}{0}] \quad (5)$$



1.2) else

arr[i] = arr[i] + 1
return;

```

public static int[] addOne(int[] arr, int n) {
    for (int i = n - 1; i >= 0; i--) {
        if (arr[i] == 9) {
            arr[i] = 0;
        } else {
            arr[i]++;
            return arr;
        }
    }
    int[] ans = new int[n + 1];
    ans[0] = 1;
    return ans;
}

```

$$\text{arr} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \quad \text{gggg}$$

$i=3, \text{arr}[3]=0$

$i=2, \text{arr}[2]=0$

$i=1, \text{arr}[1]=0$

$i=0, \text{arr}[0]=0$

$$\text{ans} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 & 9 \end{bmatrix}$$

$$\text{arr} = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

$i=4, \underline{\text{arr}[4]}++$

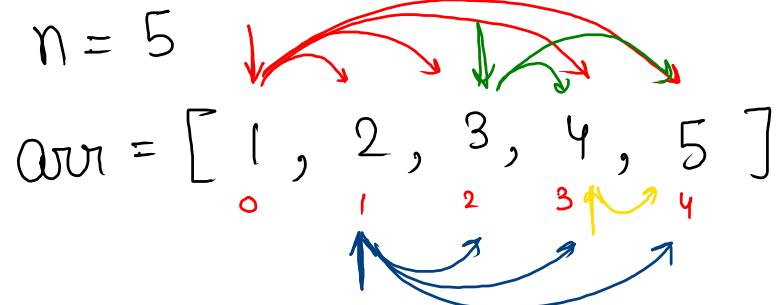
$$\text{arr} = \begin{bmatrix} 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

$i=4, \text{arr}[4]=0$

$i=3, \text{arr}[3]=0$

$i=2, \text{arr}[2]++;$
return;

Print Pair



~~pairs~~ (i, j)

1 2
1 3
1 4
1 5
2 3
2 4
2 5
3 4
3 5
4 5

```
for (int i = 0; i < n; i++) {  
    for (int j = i+1; j < n; j++) {  
        cout (arr[i] + " " + arr[j]);  
    }  
}
```

Combination: when we move only in forward direction.



with repetition [C w r]

without repetition [c wo r]

Permutation: when we move in both directions



with repetition [P w r]

without repetition [P wo r]

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }

    printPair(arr, n);
}

public static void printPair(int[] arr, int n) {

    for (int i = 0; i < n; i++) {
        for (int j = i + 1; j < n; j++) {
            System.out.println(arr[i] + " " + arr[j]);
        }
    }
}
```

logic

1, 2

Find all Combination

n = 5
arr =

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

sum = 8

Ans = 3, 5
 4, 4

observation

↳ w ✓
↳ wo ✓

pairs

1, 1	x	3, 3	x
1, 2	x	3, 4	x
1, 3	x	3, 5	✓
1, 4	x	4, 4	✓
1, 5	x	4, 5	x
2, 2	x	5, 5	x
2, 3	x		
2, 4	x		
2, 5	x		

```
for( int i=0; i<n; i++ ) {  
    for( int j=i ; j<n; j++ ) {  
        if( arr[i] + arr[j] == sum ) {  
            cout << " " ;  
        }  
    }  
}
```

Code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int sum = scn.nextInt();

    findComb(arr, n, sum);
}

public static void findComb(int[] arr, int n, int sum) {

    for (int i = 0; i < n; i++) {
        for (int j = i; j < n; j++) {
            if (arr[i] + arr[j] == sum) {
                System.out.println(arr[i] + " " + arr[j]);
            }
        }
    }
}
```

Coding part

Combination with repetition : $i = 0$
 $j = i$

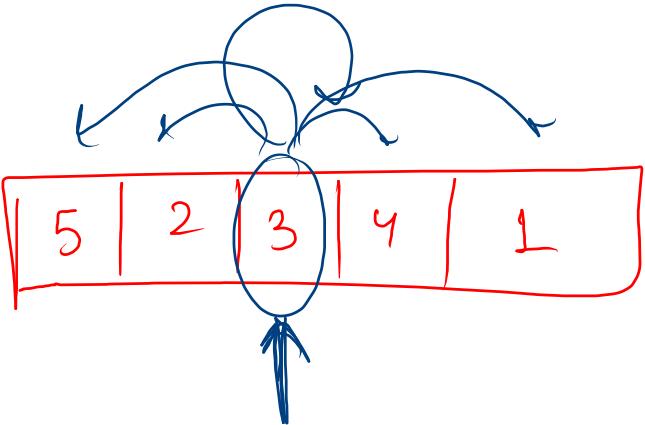
Combination without repetition : $i = 0$
 $j = i + 1$

Permutation with repetition : $i = 0$
 $j = 0$

Permutation without repetition : $i = 0$ and
 $j = 0$ check cond

Greater Than Me

L P ω R



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