


Printing all Combinations

Print all combinations of the number 12345. If you have done it correctly, there will be 120 numbers.



First, let's find
some patterns
in the 12345.

12345	23451	34512	45123	51234
12354	23415	34521	45132	51243
12453	23514	34125	45231	51342
12435	23541	34152	45213	51324
12534	23145	34251	45312	51423
12543	23154	34215	45321	51432
13452	24513	35124	41235	52341
13425	24531	35142	41253	52314
13524	24135	35241	41352	52413
13542	24153	35214	41325	52431
13245	24351	35412	41523	52134
13254	24315	35421	41532	52143
14523	25134	31245	42351	53412
14532	25143	31254	42315	53421
14235	25341	31452	42513	53124
14253	25314	31425	42531	53142
14352	25413	31524	42135	53241
14325	25431	31542	42153	53214
15234	21345	32451	43512	54123
15243	21354	32415	43521	54132
15342	21453	32514	43125	54231
15324	21435	32541	43152	54213
15423	21534	32145	43251	54312
15432	21543	32154	43215	54321

12345

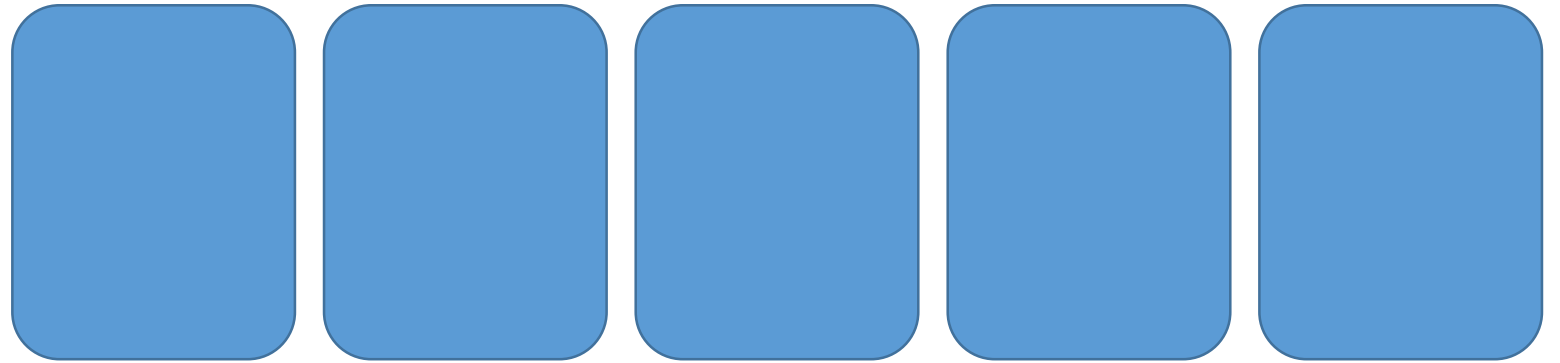
12354

12453

12435

12534

12543



5 choices

4 remaining choices

3 remaining choices

2 remaining choices

1 remaining choice

```
public void fourth(int p, int q, int r, int s, int t) {  
    System.out.println(" "+p+" "+q+" "+r+" "+s+" "+t);  
    System.out.println(" "+p+" "+q+" "+r+" "+t+" "+s);  
}
```

12345
12354
12453
12435
12534
12543

It leaves the first three digits alone

This prints out the pair, with the last two digits swapped.

```
public void third(int k, int l, int m, int n, int o) {  
    fourth (k, l, m, n, o);  
    fourth (k, l, n, o, m);  
    fourth (k, l, o, m, n);  
}
```

It leaves the first two digits alone

12	345
12	354
12	453
12	435
12	534
12	543

This sends rotates the last three digits, creating these three sets.

12345
12354
12453
12435
12534
12543
13452
13425
13524
13542
13245
13254
14523
14532
14235
14253
14352
14325
15234
15243
15342
15324
15423
15432

```
public void second(int f, int g, int h, int i, int j) {  
    third (f, g, h, i, j);  
    third (f, h, i, j, g);  
    third (f, i, j, g, h);  
    third (f, j, g, h, i);  
}
```

Fixes the first digit

Rotates the rest to choose the second digit

```

public void first(int a, int b, int c, int d, int e) {
    second (a, b, c, d, e);
    second (b, c, d, e, a);
    second (c, d, e, a, b);
    second (d, e, a, b, c);
    second (e, a, b, c, d);
}

```

12345	23451	34512	45123	51234
12354	23415	34521	45132	51243
12453	23514	34125	45231	51342
12435	23541	34152	45213	51324
12534	23145	34251	45312	51423
12543	23154	34215	45321	51432
13452	24513	35124	41235	52341
13425	24531	35142	41253	52314
13524	24135	35241	41352	52413
13542	24153	35214	41325	52431
13245	24351	35412	41523	52134
13254	24315	35421	41532	52143
14523	25134	31245	42351	53412
14532	25143	31254	42315	53421
14235	25341	31452	42513	53124
14253	25314	31425	42531	53142
14352	25413	31524	42135	53241
14325	25431	31542	42153	53214
15234	21345	32451	43512	54123
15243	21354	32415	43521	54132
15342	21453	32514	43125	54231
15324	21435	32541	43152	54213
15423	21534	32145	43251	54312
15432	21543	32154	43215	54321

Rotates to
choose the
first digit

```
public class AllCombinations {  
  
    public static void main(String args[]) {  
        new AllCombinations() ;  
    }  
  
    public AllCombinations() {  
  
    }  
  
}
```



Starter
Code

```

public class AllCombinations2 {

    public static void main(String args[]) {
        new AllCombinations2() ;
    }

    public AllCombinations2() {
        first(1, 2, 3, 4, 5);
    }

    public void first(int a, int b, int c,
int d, int e) {
        second (a, b, c, d, e);
        second (b, c, d, e, a);
        second (c, d, e, a, b);
        second (d, e, a, b, c);
        second (e, a, b, c, d);
    }
}

```

The
complete
code

```

public void second(int f, int g, int h, int
i, int j) {
    third (f, g, h, i, j);
    third (f, h, i, j, g);
    third (f, i, j, g, h);
    third (f, j, g, h, i);
}

public void third(int k, int l, int m, int n,
int o) {
    fourth (k, l, m, n, o);
    fourth (k, l, n, o, m);
    fourth (k, l, o, m, n);
}

public void fourth(int p, int q, int r, int
s, int t) {
    System.out.println(""+p+""+q+""+r+""+s+""+t);
    System.out.println(""+p+""+q+""+r+""+t+""+s);
}
}

```

How the method calls work

```
first (1, 2, 3, 4, 5);
```

```
second (a, b, c, d, e);  
second (b, c, d, e, a);  
second (c, d, e, a, b);  
second (d, e, a, b, c);  
second (e, a, b, c, d);
```

```
third (f, g, h, i, j);  
third (f, h, i, j, g);  
third (f, i, j, g, h);  
third (f, j, g, h, i);
```

```
third (f, g, h, i, j);  
third (f, h, i, j, g);  
third (f, i, j, g, h);  
third (f, j, g, h, i);
```

```
third (f, g, h, i, j);  
third (f, h, i, j, g);  
third (f, i, j, g, h);  
third (f, j, g, h, i);
```

```
third (f, g, h, i, j);  
third (f, h, i, j, g);  
third (f, i, j, g, h);  
third (f, j, g, h, i);
```

```
third (f, g, h, i, j);  
third (f, h, i, j, g);  
third (f, i, j, g, h);  
third (f, j, g, h, i);
```

```
fourth (k, l, m, n, o);  
fourth (k, l, n, o, m);  
fourth (k, l, o, m, n);
```

```
fourth (k, l, m, n, o);  
fourth (k, l, n, o, m);  
fourth (k, l, o, m, n);
```

```
fourth (k, l, m, n, o);  
fourth (k, l, n, o, m);  
fourth (k, l, o, m, n);
```

```
fourth (k, l, m, n, o);  
fourth (k, l, n, o, m);  
fourth (k, l, o, m, n);
```

```
System.out.println(count+" "+p+" "+q+" "+r+" "+s+" "+t);  
System.out.println(count+" "+p+" "+q+" "+r+" "+t+" "+s);
```

```
System.out.println(count+" "+p+" "+q+" "+r+" "+s+" "+t);  
System.out.println(count+" "+p+" "+q+" "+r+" "+t+" "+s);
```

```
System.out.println(count+" "+p+" "+q+" "+r+" "+s+" "+t);  
System.out.println(count+" "+p+" "+q+" "+r+" "+t+" "+s);
```

$$5 \times 4 \times 3 \times 2 = 120$$