

学科・類：

学籍番号：

名前：

授業中に配布した用紙でない場合は、「コピー」と右上に大きく書くこと。

1. 下に示したプログラムは、 $N$  進数の  $N$  桁の数を順番に表示するものである。(例えば、 $N = 2$  ならば、00, 01, 10, 11 と、 $N = 3$  ならば、000, 001, 002, 010, 011, 012, 020, 021, 022, 100, 101, 102, 110, 111, 112, 120, 121, 122, 200, 201, 202, 210, 211, 212, 220, 221, 222 と表示する。) このプログラムを改造して、各数で同じ数字が 2 つ以上現れないものだけを表示するプログラムを eclipse で作成し、プログラムを実行させ、プリントスクリーンを使ってその実行結果が見えるような画像にして、講義が始まる前に IdeTokyoTech@yahoo.co.jp まで送ること。メールのサブジェクトはレポートの回数、学籍番号、名前にすること。例：「第 8 回, 09\_12345, 林彪」。

さらに、可能な学生はデータ `int data[] = {2, -1, 4, 7, -5, 3};` に対して昇順にバカソートを実行するプログラムを作成せよ。

説明：改造したプログラムは、 $N = 3$  ならば、012, 021, 102, 120, 201, 210 と表示する。もっとも簡単な実装法は、Rep0801 の出力に対して、桁に同じ数が存在するかどうかチェックして、ない場合だけ表示するように改造すればよい。また、バカソートは改造したプログラムで得られた順列をインデックスとして使用し、データを並び替え、データが昇順になっている場合だけ表示すれば良い。

```
public class Rep0801 {
    public static void main(String[] args) {
        int N = 5, k;
        int val[] = new int[N];
        // 初期化
        for (k = 0 ; k < N ; ++k) val[k] = 0;

        k = N - 1;
        while(true) {
            if (val[k] == N) {
                if (k == N - 1) break; // 解はすべて見つかった。
                val[k] = 0;
                ++k;
                val[k] += 1;
            } else {
                // 表示
                for (int l = N - 1 ; l >= 0 ; --l) {
                    System.out.printf("%2d ", val[l]);
                }
                System.out.println(); //改行
                k = 0;
                val[0] += 1;
            }
        }
    }
}
```

```

public class Rep0801Perm {
    public static void main(String[] args) {
        int N = 5, k;
        int val[] = new int[N];

        // 初期化
        for (k = 0 ; k < N ; ++k) val[k] = 0;

        k = 0;
        while(true) {
            if (val[k] == N) {
                if (k == N - 1) break;
                val[k] = 0;
                ++k;
                val[k] += 1;
            } else {
                // 同じ数字が2以上の桁で現れるかチェックする。
                boolean isSame = false;
                for (int i = 0 ; i < N - 1 ; ++i) {
                    for (int j = i + 1 ; j < N ; ++j) {
                        if (val[i] == val[j]) {
                            isSame = true;
                            break;
                        }
                    }
                }
                if (isSame) break;
            }
            // 同じ数字が2つ以上の桁で現れていなければ表示する
            if (! isSame) {
                for (int l = N - 1 ; l >= 0 ; --l) {
                    System.out.printf("%2d ", val[l]);
                }
                System.out.println(); //改行
            }
            k = 0;
            val[0] += 1;
        }
    }
}

```

- 0 から  $N - 1$  までの相異なる 2 つの整数  $i$  と  $j$  を作成する。
- そのために,  $i$  を 0 から  $N - 2$  まで順番に増やし, 1 つの  $i$  に対して,  $j$  を  $i + 1$  から  $N - 1$  まで順番に増やす。
- `isSame` の初期値を `false` にする。 `val[i]` と `val[j]` に 1 つでも同じ値があれば, `isSame` を `true` にする。
- `isSame` が `false` のときだけ, 数を表示する。

```

public class Rep0801Baka {
    public static void main(String[] args) {
        int N = 6, k;
        int val[] = new int[N];
        int data[] = {2, -1, 4, 7, -5, 3};
        // 初期化
        for (k = 0 ; k < N ; ++k) val[k] = 0;

        k = N - 1;
        while(true) {
            if (val[k] == N) {
                if (k == N - 1) break; // 解はすべて見つかった。
                val[k++] = 0;
                val[k] += 1;
            } else {
                // 同じ数字が2以上の桁で現れるかチェックする。
                boolean isSame = false;
                for (int i = 0 ; i < N - 1 ; ++i) {
                    for (int j = i + 1 ; j < N ; ++j) {
                        if (val[i] == val[j]) {
                            isSame = true;
                            break;
                        }
                    }
                }
                if (isSame) break;
            }
            if (! isSame) {
                // 求めた順番でソートされているかどうかのチェック
                boolean sortOK = true;
                for (int l = 0 ; l < N - 1 ; ++l) {
                    if (data[val[l]] > data[val[l + 1]]) {
                        sortOK = false;
                        break;
                    }
                }
                // ソートされていたらデータを表示する。
                if (sortOK) {
                    for (int l = 0 ; l < N ; ++l) {
                        System.out.println((l + 1) + "番目のデータは , data[" + val[l]
                            + "] = " + data[val[l]] + "です。");
                    }
                }
            }
            k = 0;
            val[0] += 1;
        }
    }
}

```