

1. 配列はランダムアクセスが可能なデータ構造であり、整数の添字（インデックス）で要素を参照することができる。この特徴を用いると、整数からデータへの変換表を作ることができる。以下は、配列を用いて1桁の整数を英単語に“翻訳”するプログラムである。空欄を埋めて動作を確認せよ。

```
import java.util.Scanner;

public class Program {

    // final は変更されないことを示す
    final static String[] ewords = {
        "zero", "one", "two", "three",
        "four", "five", "six", "seven",
        "eight", "nine" };

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("number? ");
        int n = sc.nextInt();

        if (0 <= n && n <= 9)
            System.out.println(ewords[    ]);
    }
}
```

2. ハッシュ関数はキーと呼ばれるデータを入力とし、それからなるべくバラバラな整数値（ハッシュ値）を算出する関数である。以下は文字列のハッシュ関数の例であり、入力された文字列に対してなるべく重複しない整数を生成する。このハッシュ関数によって、次の文字列はどのような数値に変換されたか。

- (a) zero _____ (b) one _____ (c) two _____ (d) three _____ (e) four _____
 (f) five _____ (g) six _____ (h) seven _____ (i) eight _____ (j) nine _____

```
import java.util.Scanner;

public class Program {
    // 素数がよい
    final static int HASHSIZE = 31;

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        for (int i = 0; i < 10; i++) {
            System.out.print("string? ");
            String str = sc.next();
            // ハッシュ関数によって整数に変換
            int h = hash(str);
            System.out.println("code: " + h);
        }
    }

    // 任意の文字列を 0~30 の整数に変換
    public static int hash(String key) {

        int len = key.length();
        int h = 0;
        for (int i = 0; i < len; i++) {
            // i 番目の文字の文字コード
            int code = key.charAt(i);
            // System.out.println(code);

            // これも別の素数がよい
            h = 37 * h + code;
        }
        // Java では%の結果は負になる可能性
        return Math.abs(h % HASHSIZE);
    }
}
```

3. ハッシュテーブルは、データをそのハッシュ値の位置の配列要素（バケット）に格納することによって、探索を高速化する仕組みである。下記は、ハッシュテーブルの原理を理解するための単純なプログラムである。配列の添字に2で求めた整数を埋めて動作を確認せよ。また、問題点を考察せよ。

```
import java.util.Scanner;

public class Program {
    // 2.と同じ数にする
    final static int HASHSIZE = 31;

    // ハッシュテーブル（要素は文字列）
    static String[] hashtable;

    public static int hash(String key) {
        // 2.と全く同じ内容
    }

    public static void main(String[] args) {
        hashtable = new String[HASHSIZE];

        // 各バケットに初期データを登録する
        hashtable[    ] = "zero";
        hashtable[    ] = "one";
        hashtable[    ] = "two";
        hashtable[    ] = "three";
        hashtable[    ] = "four";
        hashtable[    ] = "five";
        hashtable[    ] = "six";
        hashtable[    ] = "seven";
    }
}
```

