▼Practice04

0 から 1000 までのフィボナッチ数列を出力するプログラムを作成して下さい。

フィボナッチ数列とは 0,1 から始まり、前の 2 項の合計が次の項になる数列です。

実行結果

```
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987
```

▼Practice05

0 から 1000 までのトリボナッチ数列を出力するプログラムを作成して下さい。

トリボナッチ数列とは 0,0,1 から始まり、前の 3 項の合計が次の項になる数列です。

実行結果

```
0, 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149, 274, 504, 927
```

▼Practice06

各桁の値が重複しない 4 桁のランダムな整数を出力するプログラムを作成してください。（0 から始まる値も不可です）

実行例 1

```
4293
```

実行例 2

```
3021
```

▼Practice07

4 桁のランダムな整数を生成し、数当てをするゲームを作成してください。

尚、桁と数字が一致した場合は"hit"、桁は異なるが数字が同じものがある場合は"blow"としてカウントします。

実行例（正解が 7058 の場合）

```
1 回目：　4 桁の整数を入力してください ＞ 9042
1hit / 0blow

2 回目：　4 桁の整数を入力してください ＞ 9048
2hit / 0blow

3 回目：　4 桁の整数を入力してください ＞ 5078
2hit / 2blow

4 回目：　4 桁の整数を入力してください ＞ 7058
4hit / 0blow
正解です！
```

▼Practice08

String クラスの charAt( )メソッドは、( )内に数値を設定することでその位置にある文字を参照することができます。

例： String color = “blue”;

System.out.println(color.charAt(2));

※ 添字は 0 からはじまるので u が出力されます

また、文字列の長さは、変数名に .length() を付けることで知ることができます。

例： System.out.println(color.length());

※ blue の文字数 4 が出力されます。配列と違って .length の後に()がつくので気を付けましょう。

以上を踏まえ、キーボードから入力された文字列を逆から出力するプログラムを作成してください。

実行例

```
文字列を入力してください > abcdefg
gfedcba
```

▼Practice09

メールアドレスをアカウントとドメインに分割するプログラムを作成してください。

アルゴリズムを考える練習でもあるので、Practice08 と同様のメソッドのみを使い、

for 文を使用して文字列を操作すること。

実行例 1

```
メールアドレスを入力してください > shirahata@abc.co.jp
アカウント名： shirahata
ドメイン名 ： abc.co.cjp
```

実行例 2

```
メールアドレスを入力してください > shirahata@abc@co.jp
メールアドレスではありません
```

実行例 3

```
メールアドレスを入力してください > shirahata.co.jp
メールアドレスではありません
```

↑ メールアドレスかどうかは@の数（1 個かどうか）で判断している