第一回　演習問題解答例（基本文法の復習）

諸注意

- 課題は全てメソッドを作成するものである.
- 各メソッドは「Kadai.java」内に作成し，main メソッドは除去した上で，Kadai.java のみを Web から提出する（もちろん，動作確認時に main メソッドを利用することは構わない）.
- コピペ発覚時は見せた側も見せてもらった側も両方０点とする.
- 必ずコンパイルエラーのない状態で提出すること（自動採点したいのでコンパイルエラーがあると，全て０点になってしまう）.
- 次に示すようなしょうもないミスは絶対にしないようにすること.
  - 課題１の sum()は合計値を返すメソッドを作成するものである．それにもかかわらず，「sum=10」のようにデバッグで用いたのであろうコンソール出力が残っていることがある．sum()はコンソール出力なし＋合計値を返すメソッドとして提出してほしい.

課題 1

1-1	問題設定	引数で受け取った配列の合計値を返すメソッド sum() を作成してほしい.
	引数	int 型配列
	戻り値	int 型
	諸注意	メソッドは public static で定義せよ.
	テスト例	int[] iarr = {1, 2, 3, 4, 5}; System.out.println(sum(iarr));
	例の出力	15

1−1：解答例

```
public static void main(String[] args){
	// テスト例
	int[] iarr = {1, 2, 3, 4, 5};
	System.out.println(sum(iarr));
	System.out.println(exsum(iarr));
}

// 一般的なfor文を用いた解法
public static int sum(int[] arr){
	int sum = 0;
	for(int i=0; i<arr.length; i++){
		sum += arr[i];
	}
	return sum;
}

//拡張forを用いた解法
public static int exsum(int[] arr){
	int sum = 0;
	// for([型名] [一時変数名] : [配列名]){}
	for(int num : arr){
		sum += num;
	}
	return sum;
}
```

1-2	問題設定	配列の合計を返すメソッド sum を double 型配列，boolean 型配列でも実行できるようにオーバーロードせよ．
	引数	(1) double 型配列　(2) boolean 型配列
	戻り値	(1) double 型　　(2) String 型
	諸注意	メソッドは public static で定義せよ．
	テスト例	double[] darr = {1.1d, 2.2d, 3.3d, 4.4d, 5.5d}; boolean[] barr = {true, false, false, true}; System.out.println(sum(darr)); System.out.println(sum(barr));
	例の出力	16.5 truefalsefalsetrue

1−2：解答例

```
public static void main(String[] args){
	double[] darr = {1d, 2d, 3d, 4.0, 5.0};
	System.out.println(sum(darr));
	boolean[] barr = {true, true, false, false};
	System.out.println(sum(barr));
}
//拡張for文を使える方がかっこいいので
public static double sum(double[] arr){
	double sum = 0;
	for(double num : arr){
		sum += num;
	}
	return sum;
}
//拡張for文を使える方がかっこいいので
public static String sum(boolean[] arr){
	String sum = "";
	for(boolean num : arr){
		sum += num;
	}
	return sum;
}
```

課題２

2-1	問題設定	引数で受け取った配列の各要素が特定の値と一致しているかを boolean 配列で返すメソッド where() を作成してほしい.
	引数	調査したい配列(int 型配列), 特定の値(int 型)
	戻り値	boolean 型配列
	諸注意	メソッドは public static で定義せよ.
	テスト例	int[] iarr = {1, 2, 3, 4, 5, 3}; System.out.println(sum(where(iarr, 1))); System.out.println(sum(where(iarr, 3)));
	例の出力	truefalsefalsefalsefalsefalse falsefalsetruefalsefalsetrue

2-1：解答例

```
public static void main(String[] args){
	int[] iarr = {1, 2, 3, 4, 5, 3};
	System.out.println(sum(where(iarr, 3)));
}
public static boolean[] where(int[] arr, int num){
	// 入力と同じ長さのboolean配列を定義する
	boolean[] res = new boolean[arr.length];
	for(int i=0; i<arr.length; i++){
		// 条件式はtrueかfalseになるので
		// (arr[i] == num)はboolean型である
		// =と==の区別がややこしいので括弧をつける
		// if文で条件分岐して{ture,false}を代入してもよい
		res[i] = (arr[i] == num);
	}
	return res;
}
```

課題３

3-1	問題設定	２つの条件式に基づいて一つの判定を行うために&&や‖といった論理演算子が存在する．しかしこれらの演算子では XOR の様に条件が一つだけ true の時 true となるような結果を返すことができない（正確には&&と‖と!を組み合わせることで実現できるが手間である）．そこで２つの条件式を引数とし，XOR の結果を返すメソッド xor()を作成してほしい．
	引数	条件式，条件式　（何型かは自分で考える）
	戻り値	boolean 型
	諸注意	メソッドは public static で定義せよ．
	テスト例	int a=100, b=90, c=100; System.out.println(xor(a==100, b==100)); System.out.println(xor(a==100, c==100));
	例の出力	true false

3-1：解答例

```
public static void main(String[] args){
	int a = 100, b=90, c=100;
	System.out.println(xor(a==100, b==100));
	System.out.println(xor(a==100, c==100));
}

// 条件式はboolean型になるので引数はどちらもboolean型である
// 戻り値は{ture, false}なのでboolean型である
public static boolean xor(boolean a, boolean b){
	// この式を見てもxorが思い出せないのならば
	// 福間先生に怒られてしまうぞ
	return (a && !b) || (!a && b);
}
```

3-2	問題設定	複数の条件式に基づいて一つの判定を行う例を応用し，２つ以上の条件を総合的に見て true が３の倍数回出現した場合に true を返すメソッド three() を作成してほしい．
	引数	条件式　（型は自分で考える．可変長とする．）
	戻り値	boolean 型
	諸注意	メソッドは public static で定義せよ． true が０回のときも true を返すものとする．
	テスト例	int a=100, b=90, c=100; System.out.println(three(a==100, b==100, c==100)); System.out.println(three(a==100,b!=100,c==100,a%9==0));
	例の出力	false true

3-2：解答例

```
public static void main(String[] args){
	int a=100, b=90, c=100;
	System.out.println(three(a==100, b==100, c==100));
	System.out.println(
			three(a==100, b==100, c==100, a%10==0));
}

// 条件式が可変長ということはboolean型の可変長が引数である
public static boolean three(boolean... barr){
	int count = 0;
	for(boolean f : barr){
		// barrの各条件がtrueの時のみcount++する
		if(f){
			count ++;
		}
	}
	return (count % 3 == 0);
}
```