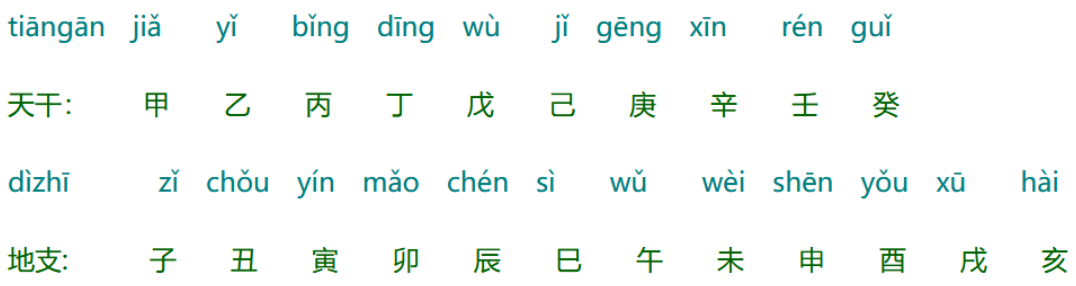
2019S CS102A Assignment 2

(prepared by Chunfeng Hu)

A2Q1: Chinese Lunar Year [20 points]

Please write a program to show the Chinese Lunar year of Stems-and-Branches (天干地支) according to the year number, and the animal symbol (属相) of the year which a person was born in.

The Stems-and-Branches uses 10 Stems called: jia, yi, bing, ding, wu, ji, geng, xin, ren, and gui(甲乙丙丁戊己庚辛壬癸), and 12 Branches called zi, chou, yin, mao, chen, si, wu, wei, shen, you, xu and hai (子丑寅卯辰巳午未申酉戌亥).Combining each of the10 Stems with one of the 12 Branches in sequence creates 60 chronological symbols. For example jiazi, yichou, bingyin, etc. These 60 symbols are used in circles and thus each year has a chronological symbol. And the Branches also represents the animal symbol of Rat, Ox, Tiger, Rabbit, Dragon, Snake, Horse, Sheep, Monkey, Rooster, Dog, Pig. For example, it is the year of JiHai (己亥年) in 2019. And it is the year of Pig.



tiāngān	jiǎ	yǐ	bǐng	dīng	wù	jǐ	gēng	xīn	rén	guǐ		
天干:	甲	乙	丙	丁	戊	己	庚	辛	壬	癸		
dìzhī	zǐ	chǒu	yín	mǎo	chén	sì	wǔ	wèi	shēn	yǒu	xū	hài
地支:	子	丑	寅	卯	辰	巳	午	未	申	酉	戌	亥

鼠
子

牛
丑

虎
寅

兔
卯

龙
辰

蛇
巳

马
午

羊
未

猴
申

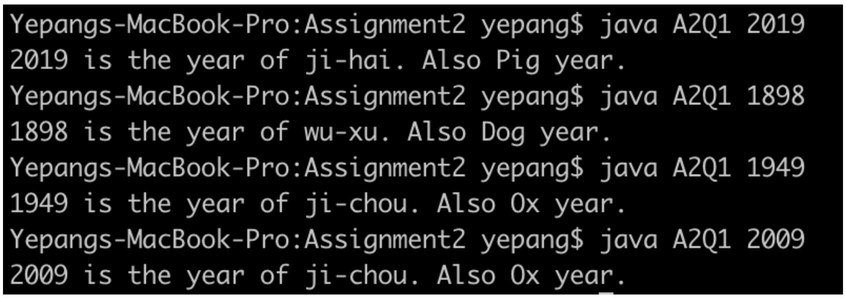
鸡
酉

狗
戌

猪
亥

Tip: Search on web how to calculate it. And write a program using ***switch*** statement.
ref: http://baijiahao.baidu.com/s?id=1543800345816500

Sample runs:



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q1 2019
2019 is the year of ji-hai. Also Pig year.
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q1 1898
1898 is the year of wu-xu. Also Dog year.
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q1 1949
1949 is the year of ji-chou. Also Ox year.
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q1 2009
2009 is the year of ji-chou. Also Ox year.
```

Sample code:

```
public class A2Q1 {
  public static void main(String[] args){
    int year = Integer.parseInt(args[0]);
                //define the array of Stems-and-Branches, animal symbols
    String stem[] = ;
    String branch[] = ;
    String animal[] = ;
                // to calculate

                // output
    System.out.printf("%s is the year of %s-%s. Also %s year.\n",...);
  }
}
```

A2Q2: Simple statistics of a number sequence [30 points]

Write a program to calculate the average (平均数), mode (众数) and median (中位数) of a sequence of numbers.

The average of the numbers is the sum divided by count. The mode is the number appears most frequently. There may be more than 1 mode in a number sequence. In that case, your program should output the multiple modes according to the order they first appear in the input sequence of numbers. The median is the number at "Middle" position of the sequence after sorting. The median is at the "Middle" position of (n-1)/2+1 when n is odd, or is the average value of numbers at position of n/2 and n/2+1.

Using Array and for statement.

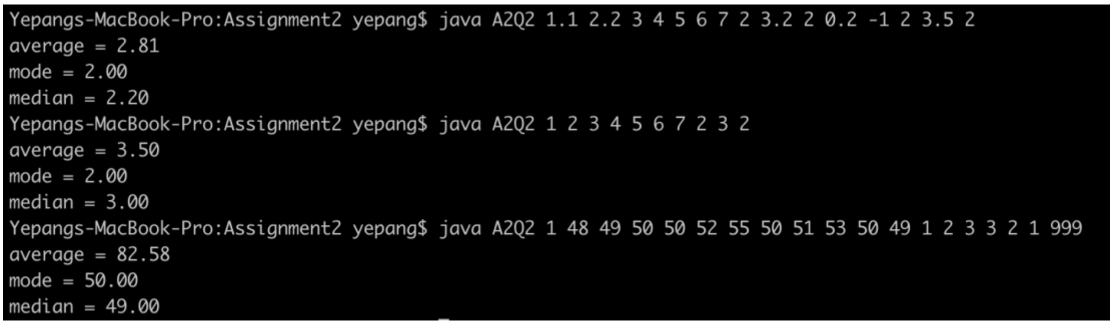
(1) Calculate the average;

(2) Count the numbers and find the number appears max times;

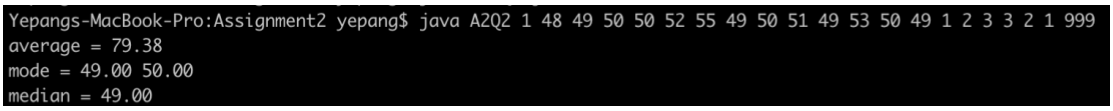
(3) Sort all numbers from small to big, and find the "Middle";

Tips: ***args.length*** shows how many arguments inputted.

Sample runs:



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q2 1.1 2.2 3 4 5 6 7 2 3.2 2 0.2 -1 2 3.5 2
average = 2.81
mode = 2.00
median = 2.20
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q2 1 2 3 4 5 6 7 2 3 2
average = 3.50
mode = 2.00
median = 3.00
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q2 1 48 49 50 50 52 55 50 51 53 50 49 1 2 3 3 2 1 999
average = 82.58
mode = 50.00
median = 49.00
```



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q2 1 48 49 50 50 52 55 49 50 51 49 53 50 49 1 2 3 3 2 1 999
average = 79.38
mode = 49.00 50.00
median = 49.00
```

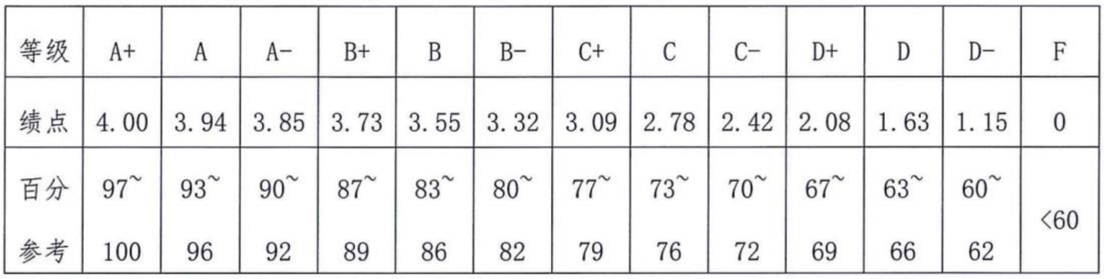
Sample code:

```
public class A2Q2 {
   public static void main(String[] args){
      double number[];
      double average = 0.0;
      double mode = 0.0;
      double median = 0.0;

      number = new double[args.length];
      //1. to get the input numbers and store in array

      for (int i=0; i<args.length; i++){

      }
      //2. average
      double sum=0.0;
```

```
        System.out.printf("average = %.2f\n",average);
        //3. count numbers
        int count[] = new int[args.length];
        for (int i=0; i<args.length; i++){
            count[i]=1;
        }
        for (int i=0; i<args.length; i++){

        }
        int maxCount=0;
                                //3.1 find the max count
        for (int i=0; i<args.length; i++){

        }
                                //3.2 find the mode with max count
        System.out.print("mode = ");

        //4.1 sorting
        for (int i=0; i<args.length; i++){
            for (int j=0; j<args.length; j++){

            }
        }

       // 4.2 median

        System.out.printf("\nmedian = %.2f\n",median);
    }
}
```

A2Q3: Grading system [20 points]

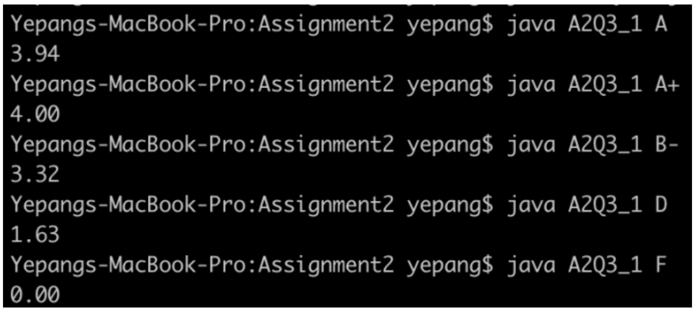
Please write 2 programs to calculate the GPA at SUSTech:



等级	A+	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
绩点	4. 00	3. 94	3. 85	3. 73	3. 55	3. 32	3. 09	2. 78	2. 42	2. 08	1. 63	1. 15	0
百分 参考	97~ 100	93~ 96	90~ 92	87~ 89	83~ 86	80~ 82	77~ 79	73~ 76	70~ 72	67~ 69	63~ 66	60~ 62	<60

(1) A2Q3_1: Input a score of letter grading, and to calculate the Grade Point.
Tips: ***string.charAt(int index)*** returns the value of the character at the specified *index* of the string

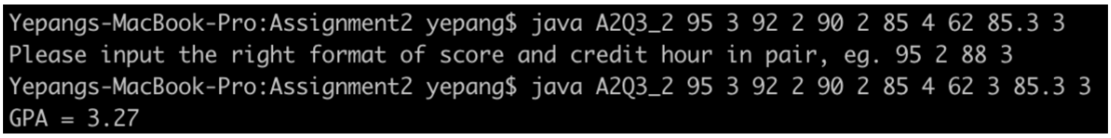
Sample runs:



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_1 A
3.94
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_1 A+
4.00
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_1 B-
3.32
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_1 D
1.63
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_1 F
0.00
```

(2) A2Q3_2: Input a serials of grades and credit hours, and to calculate the GPA.

Sample runs：



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_2 95 3 92 2 90 2 85 4 62 85.3 3
Please input the right format of score and credit hour in pair, eg. 95 2 88 3
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q3_2 95 3 92 2 90 2 85 4 62 3 85.3 3
GPA = 3.27
```

Sample code：

```
public class A2Q3_2 {
  public static void main(String[] args){
    double score[];
    int credit[];

    if (0==args.length || 0 != args.length%2){
      System.out.println("Please input the right format of score and credit hour in pair, eg.
95 2 88 3");
      return;
    }

    score = new double[args.length/2];
    credit = new int[args.length/2];
    // to get the input numbers and store in array

    for (int i=0; i<args.length; i+=2){
    }

    // calculate GP
    double GP[] = new double[args.length/2];
    for (int i=0; i<args.length/2; i++){

    }
    double GPA=0.0;

    System.out.printf("GPA = %.2f\n",GPA);
  }
}
```

A2Q4: Calculator [30 points]

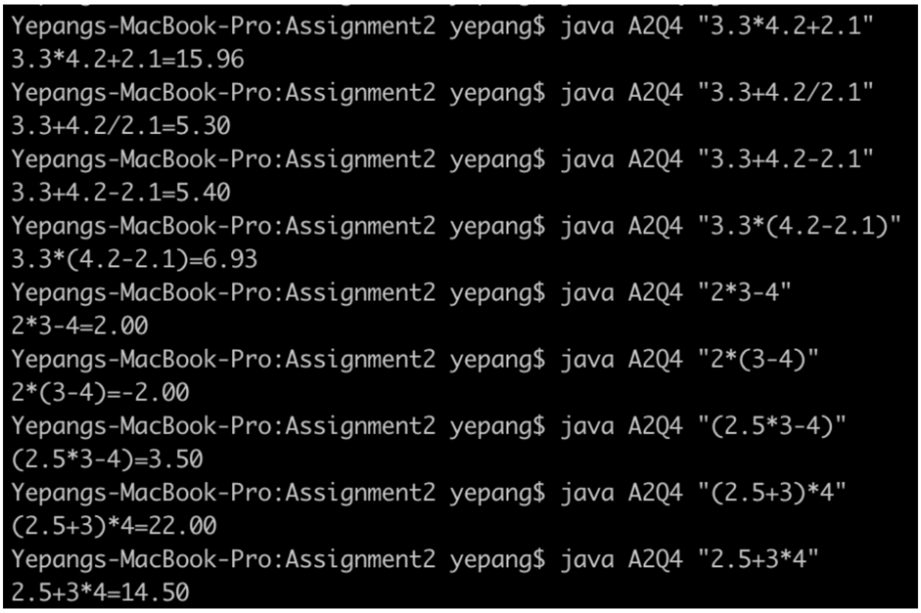
Write a program using basic arrays to calculate the value of an arithmetic expression consisting of numbers (could be floating-point) and operators +, -, * and /. The arithmetic expression is passed to the program from the command line as a string.

Tips:

(1) We can assume that the input string always represents a valid arithmetic expression;

(2) The arithmetic expression contains only 3 numbers and 2 operators. There may be a pair of brackets.

Sample runs:



```
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "3.3*4.2+2.1"
3.3*4.2+2.1=15.96
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "3.3+4.2/2.1"
3.3+4.2/2.1=5.30
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "3.3+4.2-2.1"
3.3+4.2-2.1=5.40
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "3.3*(4.2-2.1)"
3.3*(4.2-2.1)=6.93
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "2*3-4"
2*3-4=2.00
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "2*(3-4)"
2*(3-4)=-2.00
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "(2.5*3-4)"
(2.5*3-4)=3.50
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "(2.5+3)*4"
(2.5+3)*4=22.00
Yepangs-MacBook-Pro:Assignment2 yepang$ java A2Q4 "2.5+3*4"
2.5+3*4=14.50
```

Note that the quotation marks in the above sample runs (in bash on a Macbook Pro) will NOT be provided to the program as part of the command line argument. Only the string inside the quotation marks will be provided to the program as input. You may not need the quotation marks if you run your programs in command line on Windows.

Sample code:

```
public class A2Q4 {
  public static void main(String[] args) {
    //if (0 == args.length) return;
    int n = args[0].length();
    double num[] = new double[3];//
    char cal[] = new char[2];//
    int nums = 0;
    int cals = 0;
    String strNumber = "";
    //to parse the string
    for (int i = 0; i < n; i++) {
      //System.out.print(args[0].charAt(i));
      switch (args[0].charAt(i)) {
        case '+':
          // to do youself
          break;
        case '-':
          // to do youself
```

```
                    break;
                case '*':
                    // to do youself
                    break;
                case '/':
                    // to do youself
                    break;
                default:
                    // to do youself
                    break;
            }
            //System.out.print(nums+","+cals);
        }


        // to calculate
        double result = 0.0;
        if (('+' == cal[0] || '-' == cal[0]) && ('*' == cal[1] || '/' == cal[1])){
                                                // to finish
                                }
                                else
                                {
                                                // to finish
                                }
                                // to print result
        System.out.printf("%s=%.2f\n", args[0], result);
    }
}
```

Submission requirements

1. Please submit ".java" file of these five questions.
2. The class name of each ".java" file should be A2Q1, A2Q2, A2Q3_1, A2Q3_2, A2Q4 respectively to represent these five questions.
3. No Chinese characters are allowed to appear in your code.
4. No package included.
5. The arguments and the output must strictly follow the description of each question.

6. Please submit your assignment on the SAKAI site of your lab section. Marks will be deducted if you submit later than the deadline. If you submit your assignment within 24 hours after the deadline (grace period), your score will be half of the score you could get if the submission was made before the deadline. Assignments submitted after the grace period will not be graded (meaning you will get a zero for the assignment).