Some Tips for Taking Collegiate Programming Examination (CPE) Lab 4: A Number System

Rung-Bi Lin
csrlin@saturn.yzu.edu.tw

International Program in informatics for Bachelor
Yuan Ze University

Collegiate Programming Exam (CPE)

- Programs are developed for some given problems online and judged automatically. The purpose is to evaluate the programming competence of students.
- Exam's are normally held in March, June, September, and December. They are free of charge. More than 2000 students from about 40 universities in Taiwan took the exam each time. Students can freely select the test site wherever is nearer to him/her.
- Every student must program alone on a computer which does not have networking capability. No materials except English Dictionary can be brought into the exam room.
- **Web site: https://cpe.cse.nsysu.edu.tw/**

CPE Problems

- There are seven problems with different levels of difficulty. Three of them are easy ones.
- Except the public test data sets, there are some hidden test data sets. The public data sets can be used to help debug a program.
- There are five difficulty levels denoted by the number of star symbol *. The easiest problems are denoted by one star symbol. Each exam will have at least one problem out of about 40 one-star problems being included into the problem set. Please refer to the following links for details.
  http://cpe.cse.nsysu.edu.tw/environment.php#starList

Coming Exam

- Exam date: 2020/10/20
- Starting date and time for registration: 2020/10/6, 14:25
- Ending date and time for registration: 2020/10/16,18:00
- Exam time
  - 17:30-17:40 for entering exam room，no admission after **18:00**
  - 17:40-18:30 for practice
  - 18:40-21:40 for exam
- Programming and evaluating platform
  - CodingFrenzy http://coding-frenzy.arping.me/

Some Rules

- Must obtain an account before you can register for a test.
- If you are absent for a test without permission, your right for next test will be suspended.
- If you cheat, the test score is zero and your right for next three tests will be suspended.

Example of a Problem

(CPE10461, UVA10931)

We define the parity of an integer n as the sum of the bits in binary representation computed modulo two. As an example, the number $21 = 10101_2$ has three 1s in its binary representation so it has parity $3 (\text{mod} 2)$, or 1.

In this problem you have to calculate the parity of an integer $1 \leq I \leq 2147483647$.

Input

Each line of the input has an integer I and the end of the input is indicated by a line where $I = 0$ that should not be processed.

Output

For each integer I in the inputt you should print a line '`The parity of` B `is` P `(mod 2).`', where B is the binary representation of I.

Sample Input

```
1
2
10
21
0
```

Sample Output

```
The parity of 1 is 1 (mod 2).
The parity of 10 is 1 (mod 2).
The parity of 1010 is 2 (mod 2).
The parity of 10101 is 3 (mod 2).
```

Basics of Reading Test Data

- Typically, one of the following three ways of reading test data into a program is used.
  - Clearly specifies the number of data items that will be read.
  - Read data items until end of input is met. That is, after the last item is read.
  - Read data items until a particular number or symbol is met. Typically, stop reading after a zero is read.

Reading n Data Items

Input

```
3
10
31
50
```

Corresponding code

```
int main() {
    int n;
    cin>>n;
    while (n--) {
        // 讀取每筆資料
    }
    return 0;
}
```

```
int main() {
        Int n;
        cin >> n;
        while (n > 0){
        // do something here
        n = n-1;
        }
        return 0;
}
```

Problem Example of Reading n Data Items

- Problem Title: Summing two numbers
- Calculate the sum of two numbers on an input lines
  - Input
    - The first line gives the number of data items. Starting from the second line, every line presents one data item.
    - Every data item consists of two numbers separated by a space character.
  - Output

Each line shows the result of adding the two numbers.

Input

```
3
10 20
33 25
41 64
```

Output

```
30
58
105
```

Reading Data till the End of Input

Input

```
10
31
50
```

Corresponding code

```
int main() {
    int x;
    while (cin>>x) {
        // 處理目前這筆資料
    }
    return 0;
}
```

If cin >> x can read the input successfully, a value of greater 0 will be returned. Hence, the condition in while statement will be true. To stop reading you need to press a ^Z (i.e., cntl Z).

Problem Example of Reading Data Items till End of Input

- Problem Title: Summing two numbers
- Calculate the sum of two numbers on an input lines
  - Input
    - Start from the first line, every line presents one data item. Continue reading data items till the end of file.
    - Every data item consists of two numbers separated by a space character.
  - Output

  Each line shows the result of adding the two numbers.

Input

```
10 20
33 25
41 64
```

Output

```
30
58
105
```

Reading Data Items till 0's Are Encountered

Input

```
10
31
50
0
```

Corresponding Code

```
int main() {
    int n;
    while (cin>>n) {
        if (n==0) break;

        // ...
    }
    return 0;
}
```

```
int main() {
        Int n;
        while (cin >> n && n != 0){
        // do something here
        }
        return 0;
}
```

Problem Example of Reading Data Items till 0's Are Met

- Problem Title: Summing two numbers
- Calculate the sum of two numbers on an input lines
  - Input
    - Start from the first line, every line presents one data item. Continue reading data items till two 0's are read.
    - Every data item consists of two numbers separated by a space character.
  - Output

Each line shows the result of adding the two numbers.

Input

```
10 20
33 25
41 64
0 1
0 0
```

Output

```
30
58
105
1
```

CPE Web Site

- Collegiate Programming Exam
  - https://cpe.cse.nsysu.edu.tw/
- It uses「瘋狂程設」評判系統 (codingFrenzy System online judge system)
  - You have to get an account by http://coding-frenzy.arping.me/

Lab 4: A Number System with Digits and Text

Problem description

A number system uses ’kuti’ (10000000), ’lakh’ (100000), ’hajar’ (1000), ’shata’ (100) as units while expanding and converting a number to text. You are going to write a program to convert a given number to text with these units.

Input

The input file may contain several test cases. Each case will contain a non-negative number ≤999999999999.

Output

For each case of input, you have to output a line containing the converted text.

Sample Input & Output

Sample Input

23764
45897458973958
0
1000000001
900010000400100
999999999999999
8931000290
10000000

Sample Output

23 hajar 7 shata 64
45 lakh 89 hajar 7 shata 45 kuti 89 lakh 73 hajar 9 shata 58
0
1 shata kuti 1
9 kuti 1 hajar kuti 4 lakh 1 shata
9 kuti 99 lakh 99 hajar 9 shata 99 kuti 99 lakh 99 hajar 9 shata 99
8 shata 93 kuti 10 lakh 2 shata 90
1 kuti