Fundamental Computer Programming - C++ Lab(I&II)

LAB 11: Basics of Collegiate Programming Exam (CPE)

International Program in Informatics for Bachelor
College of Informatics
Yuan Ze University

Rung-Bin Lin

May 13, 2021

Next CPE

https://cpe.cse.nsysu.edu.tw/

Date of Exam: 2021/5/25

Start Date & Time for Registration: 14:25, 2021/5/11 (Tue.)

End Date & Time for Registration: 18:00, 2021/5/21 (Fri.) 18:00

Registration: http://cpe.cse.nsysu.edu.tw/cpe/

Programming Skill Requirements

- **Students of the International Bachelor Program in Informatics must satisfy the requirements of programming skills before they can take the Special Project Course or Practical Training Course.**
- **Way of fulfilling such a requirement: Taking CPE and having one of the following achievements:**
  - **Answer at least two problems correctly in an CPE exam.**
  - **Answer at least three problems correctly accumulated in all the CPE exams taken.**
- **No fee for taking CPE**
- **Held four times per year: March, May/June, September, December**
- **CPE web site**
  https://cpe.cse.nsysu.edu.tw/
- **CPE Manual**
  https://cpe.cse.nsysu.edu.tw/doc/CPE_manual_en.pdf
- **Notice of CPE**
  https://cpe.cse.nsysu.edu.tw/doc/CPE_examinee_notes_EN.pdf
- **Book for CPE (in Chinese)**
  https://www.books.com.tw/products/0010575343

Key to Solving the Problems in CPE

- **The difficulty of the CPE problems is ranked from one * to five *'s. One * problem is simplest. Every CPE exam typically has about two one * problems.**
- **Key to solving the problem**
  - **Understand the problem first. If you wish, you may skip something written in the beginning (prologue).**
  - **Know the input format.**
  - **Know the output format.**
  - **Use functions given in some libraries.**

Dealing with Inputs

- **Typical form of inputs**
  - **First line: giving the number of test cases**
  - **Second line: Starting to provide input data for test case 1.**
    - **A test case takes more than one line**
      - Telling us the number of lines needed to be read
      - Not telling us the number of lines needed to be read, i.e., providing a way to know when come to the end of reading the input data for a test case, typically some special characters being read.
    - **A test case takes only one line (also applied to reading a line in multi-line test case)**
      - Telling us the number of tokens (integers or strings or some other things) explicitly needed to be read from the line
      - Not telling us the number of tokens needed to be read from the line
        - Need to detect the end of a line. This is troublesome.
- **Not a typical form of inputs**
  - **Not giving the number of test cases**
    - **Giving an explicit way to determine the end of input.**
    - **Not giving an explicit way to determine the end of input**

Examples of Handling Inputs

Vito's Family

The world-known gangster Vito Deadstone is moving to New York. He has a very big family there, all of them living in Lamafia Avenue. Since he will visit all his relatives very often, he is trying to find a house close to them. Vito wants to minimize the total distance to all of them and has blackmailed you to write a program that solves his problem.

Input

The input consists of several test cases. The first line gives the number of test cases. After that, the input data of a test case are contained in a line. The first number in a line gives the number of houses. It is then followed by the street numbers (also integers) s_1, s_2, . . . , s_i , . . . , s_r of all the houses in the test case, where they live ($0 < s_i < 30000$ and $0 < r < 500$).

Output

For each test case, your program must write the minimal sum of distances from the optimal Vito's house to each one of his relatives. The distance between two street numbers s_i and s_j is $d_{ij} = |s_i - s_j|$.

Sample Input

```
2
2 2 4
3 2 4 6
```

Sample Output

```
2
4
```

Codes for Known Number of Test Cases and Number of Houses in Each Test Case

```
#include <iostream>
#include <algorithm>    // std::sort
using namespace std;
int main()
{
   int testcase;    //Number of test cases
   cin>>testcase; // Read the number of test cases
   int dist[500]={0}; // Storing distance
   for(int i=0;i<testcase;i++) {
      int h;    //Number of houses
      cin>>h;  //reading the number of houses
      int addr[30000]; //Storing addresses
      int bestLoc=0; // Best location
   // Read the input data of each test cases
      for(int j=0; j<h; j++)  // h is known explicitly
      cin>>addr[j];  // read an address
sort(addr,addr+h);  // Sort all the addresses
bestLoc=addr[(int)h/2];    // Take the median
for(int j=0; j<h; j++)
      dist[i]+=abs(addr[j]-bestLoc);

//cout<<"Optimal : "<<best<<endl;
//cout<<"Distance : "<<d[i]<<endl;
 }
  for (int i=0;i<testcase;i++)
     cout<<dist[i]<<endl;

  return 0;
}
```

Not Knowing the Number of Test Cases Explicitly

The input consists of several test cases. ~~The first line gives the number of test cases.~~ The input data of a test case are contained in a line. The first number in a line gives the number of houses. It is then followed by the street numbers (also integers) $s_1, s_2, \ldots, s_i, \ldots, s_r$ of all the houses in the test case, where they live ($0 < s_i < 30000$ and $0 < r < 500$).

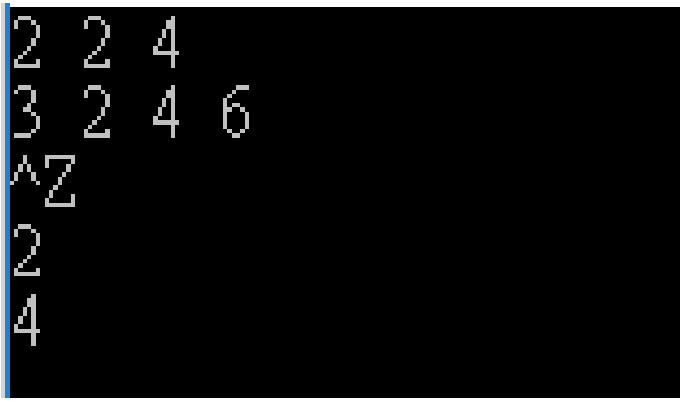
Note: Have to use control Z to end the input data.

Sample Input

2 2 4

3 2 4 6

Sample Output



Codes for Not Knowing the Number of Test Cases

```
#include <iostream>
#include <algorithm>    // std::sort
using namespace std;
int main()
{
   int testcase =0 ;    //Number of test cases
   cin>>testcase; //raeding the number of test cases
   int dist[500]={0}; // Storing distance
   int addr[30000]; //Storing addresses
   int bestLoc=0; // Best location
   for(int i=0;i<testcase;i++) {
      int h;   //Number of houses
      cin>>h; // Read the number of houses
    while (cin >> h) {     // detecting end of input
// Read the input data of each test cases
     for(int j=0; j<h; j++)  // h is known explicitly
     cin>>addr[j];  // read an address
sort(addr,addr+h);  // Sort all the addresses
 bestLoc=addr[(int)h/2];    // Take the median
 for(int j=0; j<h; j++)
       dist[testcase]+=abs(addr[j]-bestLoc);

testcase++;
 //cout<<"Optimal : "<<best<<endl;
//cout<<"Distance : "<<d[i]<<endl;
 }
   for (int i=0;i<testcase;i++)
      cout<<dist[i]<<endl;

   return 0;
}
```

Note: Have to use control Z to end the input data.

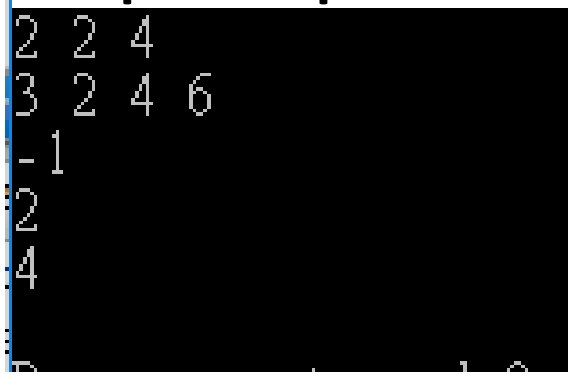
Not Knowing the Number of Test Cases Explicitly, but Input Ending with Some Special Characters Being Read

- **The input consists of several test cases. The input data of a test case are contained in a line. The first number in a line gives the number of houses. It is then followed by the street numbers (also integers) $s_1, s_2, \ldots, s_i, \ldots, s_r$ of all the houses in the test case, where they live ($0 < s_i < 30000$ and $0 < r < 500$). When the first number in a line being read is a negative number, this means end of input.**

Sample Input

```
2 2 4
3 2 4 6
-1
```

Sample Output



Code for the Input Ended with Special Characters

```
#include <iostream>
#include <algorithm>    // std::sort
using namespace std;
int main()
{
   int testcase =0 ;    //Number of test cases
   int dist[500]={0}; // Storing distance
   int addr[30000]; //Storing addresses
   int bestLoc=0; // Best location
  // stop when number of houses <0
   while (cin >> h && h >= 0) {
    // Read the input data of each test cases
      for(int j=0; j<h; j++)  // h is known explicitly
      cin>>addr[j];  // read an address
sort(addr,addr+h);  // Sort all the addresses
 bestLoc=addr[(int)h/2];    // Take the median
 for(int j=0; j<h; j++)
       dist[testcase]+=abs(addr[j]-bestLoc);

testcase++;
 //cout<<"Optimal : "<<best<<endl;
//cout<<"Distance : "<<d[i]<<endl;
  }
    for (int i=0;i<testcase;i++)
       cout<<dist[i]<<endl;

   return 0;
}
```

Not Knowing the Number of Houses Explicitly

The input consists of several test cases. The first line gives the number of test cases. The input data of a test case are contained in a line that gives the street numbers (also integers) $s_1, s_2, \ldots, s_i, \ldots, s_r$ of all the houses in the test case, where they live ($0 < s_i < 30000$ and $0 < r < 500$).

Sample Input

```
2
2 4
2 4 6
```

Because the number of houses is not explicitly specified. You have to count it when you read the street numbers from a line. There are two ways for this task:

1. **You can use getline() to read the whole line into a stringstream and then use cin to read input data from the stringstream.**
2. **Without using stringstreamm but detecting the end of a line to know whether the program come to the end of a test case.**

Learn how to use string stream

String stream can be used to extract individual tokens (words, numbers, strings) from a string when it is used along with >> (stream extraction operator). It is just like *cin*. String stream can be used to insert individual tokens (words, numbers, strings) into a string when it is used along with << (stream insertion operator). It is just like *cout*.

Codes for Not Knowing the Number of Houses

```
#include <iostream>
#include <sstream>
#include <string>
#include <algorithm>    // std::sort
using namespace std;
int main()
{
   int testcase;    //Number of test cases
   string aLine;
   stringstream ssLine;
   int dist[500]={0}; // Storing distance

   cin>>testcase;
   getline(cin, aLine);  // completing reading the first line
   for(int i=0;i<testcase;i++) {
      int h = 0;    //Number of houses
      int addr[30000]; //Storing addresses
      int bestLoc=0; // Best location
      getline(cin, aLine); // getting a test case
      ssLine << aLine; // writing aLine into ssLine;
      // Read the input data of each test cases
      while (ssLine >> addr[h])
         h++;
      sort(addr,addr+h);  // Sort all the addresses
      bestLoc=addr[(int)h/2];    // Take median
      for(int j=0; j<h; j++)
      dist[i]+=abs(addr[j]-bestLoc);

      //cout<<"Optimal : "<<best<<endl;
      //cout<<"Distance : "<<d[i]<<endl;
      ssLine.str(string()); // reset ssLine to an empty string
      ssLine.clear(); // reset flag “eof” to 0
   }
   for (int i=0;i<testcase;i++)
      cout<<dist[i]<<endl;
   return 0;
}
```

LAB-11 (PART I):Not knowing the Number of Test Cases & the Number of Houses in each Test Case (50%)

The world-known gangster Vito Deadstone is moving to New York. He has a very big family there. Since he will visit all his relatives very often, he is trying to find a house close to them. **Assume that the address in the New York is specified by a pair of numbers that represent avenue and street numbers. The streets and avenues are orthogonal to each other.** Vito wants to minimize the total distance to all of them and has blackmailed you to write a program that solves his problem.

The input consists of several test cases. It starts with the input data of the first test case. The input data of each test case are contained in a line that gives the avenue numbers a_i and street numbers s_i (also integers), such as $a_1, s_1, a_2, s_2, \ldots, a_i, s_i, \ldots, a_r, s_r$, of all the houses in the test case, where their houses are located ($0 < s_i, a_i < 30000$ and $0 < r < 500$).

The first part of your task is to develop the code for this problem with this sort of input format using stringstream to process input data.

Sample Input

2 0 4 1 6 10 8 9

12 3 6 7 100 395 400 127 83 910

1 3 5 7 9 11 13 2 4 6 8 10 12 12

1 2 3 4 5 6 7 8

Note: Have to use control Z to end the input data.

Sample Output

26

1777

46

16

LAB 11(Part II): Not knowing the Number of Test Cases & the Number of Houses in each Test Case (50%)

- **The second part of your task is to develop the code for this problem without using stringstream to process this sort of input data.**

Sample Input

2 0 4 1 6 10 8 9

12 3 6 7 100 395 400 127 83 910

1 3 5 7 9 11 13 2 4 6 8 10 12 12

1 2 3 4 5 6 7 8

Note: Have to use control Z to end the input data.

Sample Output

26

1777

46

16