考试时间 0:00:00:00

一. 函数题

1 2

二. 编程题

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
16 17 18 19 20
21 22 23 24 25
26

中国大学MOOC-陈越、何钦铭-数据结构-2016秋

715分

一. 函数题 2小题，共计45分

二. 编程题 26小题，共计670分

函数题 编程题

- 编译器：
gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
作者：陈越
单位：浙江大学

05-树8

将一系列给定数字插入一个初始为空的小顶堆`H[]`。随后对任意给定的下标`i`，打印从`H[i]`到根结点的路径。

输入格式:

每组测试第1行包含2个正整数N和$M(\leq 1000)$，分别是插入元素的个数、以及需要打印的路径条数。下一行给出区间[-10000, 10000]内的N个要被插入一个初始为空的小顶堆的整数。最后一行给出M个下标。

输出格式:

对输入中给出的每个下标`i`，在一行中输出从`H[i]`到根结点的路径上的数据。数字间以1个空格分隔，行末不得有多余空格。

输入样例:

```
5 3
46 23 26 24 10
5 4 3
```

输出样例:

```
24 23 10
46 23 10
26 10
```

- 编译器：
gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
作者：陈越
单位：浙江大学

05-树9

We have a network of computers and a list of bi-directional connections. Each of these connections allows a file transfer from one computer to another. Is it possible to send a file from any computer on the network to any other?

Input Specification:

Each input file contains one test case. For each test case, the first line contains N ($2 \leq N \leq 10^4$), the total number of computers in a network. Each computer in the network is then represented by a positive integer between 1 and N. Then in the following lines, the input is given in the format:

```
I c1 c2
```

where `I` stands for inputting a connection between `c1` and `c2`; or

```
C c1 c2
```

where `C` stands for checking if it is possible to transfer files between `c1` and `c2`; or

```
S
```

where `S` stands for stopping this case.

Output Specification:

For each `C` case, print in one line the word "yes" or "no" if it is possible or impossible to transfer files between `c1` and `c2`, respectively. At the end of each case, print in one line "The network is connected." if there is a path between any pair of computers; or "There are `k` components." where `k` is the number

of connected components in this network.

Sample Input 1:

```
5
C 3 2
I 3 2
C 1 5
I 4 5
I 2 4
C 3 5
S
```

Sample Output 1:

```
no
no
yes
There are 2 components.
```

Sample Input 2:

```
5
C 3 2
I 3 2
C 1 5
I 4 5
I 2 4
C 3 5
I 1 3
C 1 5
S
```

Sample Output 2:

```
no
no
yes
yes
The network is connected.
```

-

编译器：

gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc

作者：陈越

单位：浙江大学

05-树10

In 1953, David A. Huffman published his paper "A Method for the Construction of Minimum-Redundancy Codes", and hence printed his name in the history of computer science. As a professor who gives the final exam problem on Huffman codes, I am encountering a big problem: the Huffman codes are NOT unique. For example, given a string "aaaxuaxz", we can observe that the frequencies of the characters 'a', 'x', 'u' and 'z' are 4, 2, 1 and 1, respectively. We may either encode the symbols as {'a'=0, 'x'=10, 'u'=110, 'z'=111}, or in another way as {'a'=1, 'x'=01, 'u'=001, 'z'=000}, both compress the string into 14 bits. Another set of code can be given as {'a'=0, 'x'=11, 'u'=100, 'z'=101}, but {'a'=0, 'x'=01, 'u'=011, 'z'=001} is NOT correct since "aaaxuaxz" and "aazuaxax" can both be decoded from the code 00001011001001. The students are submitting all kinds of codes, and I need a computer program to help me determine which ones are correct and which ones are not.

Input Specification:

Each input file contains one test case. For each case, the first line gives an integer N ($2 \le N \le 63$), then followed by a line that contains all the N distinct characters and their frequencies in the following format:

```
c[1] f[1] c[2] f[2] ... c[N] f[N]
```

where `c[i]` is a character chosen from {'0' - '9', 'a' - 'z', 'A' - 'Z', '_'}, and `f[i]` is the frequency of `c[i]` and is an integer no more than 1000. The next line gives a positive integer M (≤ 1000), then followed by M student submissions. Each student submission consists of N lines, each in the format:

```
c[i] code[i]
```

where `c[i]` is the `i`-th character and `code[i]` is an non-empty string of no more than 63 '0's and '1's.

Output Specification:

For each test case, print in each line either "Yes" if the student's submission is correct, or "No" if not.

Note: The optimal solution is not necessarily generated by Huffman algorithm. Any prefix code with code length being optimal is considered correct.

Sample Input:

```
7
A 1 B 1 C 1 D 3 E 3 F 6 G 6
4
A 00000
B 00001
C 0001
D 001
E 01
```

```
F 10
G 11
A 01010
B 01011
C 0100
D 011
E 10
F 11
G 00
A 000
B 001
C 010
D 011
E 100
F 101
G 110
A 00000
B 00001
C 0001
D 001
E 00
F 10
G 11
```

Sample Output:

```
Yes
No
No
```

-

编译器：

gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc

作者：陈越

单位：浙江大学

06-图1

给定一个有N个顶点和E条边的无向图，请用DFS和BFS分别列出其所有的连通集。假设顶点从0到$N-1$编号。进行搜索时，假设我们总是从编号最小的顶点出发，按编号递增的顺序访问邻接点。

输入格式:

输入第1行给出2个整数$N(0<N\le 10)$和E，分别是图的顶点数和边数。随后E行，每行给出一条边的两个端点。每行中的数字之间用1空格分隔。

输出格式:

按照"{ v_1 v_2 ... v_k }"的格式，每行输出一个连通集。先输出DFS的结果，再输出BFS的结果。

输入样例:

```
8 6
0 7
0 1
2 0
4 1
2 4
3 5
```

输出样例:

```
{ 0 1 4 2 7 }
{ 3 5 }
{ 6 }
{ 0 1 2 7 4 }
{ 3 5 }
{ 6 }
```

-

编译器：

gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc

作者：陈越

单位：浙江大学

06-图2

This time let us consider the situation in the movie "Live and Let Die" in which James Bond, the world's most famous spy, was captured by a group of drug dealers. He was sent to a small piece of land at the center of a lake filled with crocodiles. There he performed the most daring action to escape -- he jumped onto the head of the nearest crocodile! Before the animal realized what was happening, James jumped again onto the next big head... Finally he reached the bank before the last crocodile could bite him (actually the stunt man was caught by the big mouth and barely escaped with his extra thick boot).

Assume that the lake is a 100 by 100 square one. Assume that the center of the lake is at (0,0) and the northeast corner at (50,50). The central island is a disk centered at (0,0) with the diameter of 15. A number of crocodiles are in the lake at various positions. Given the coordinates of each crocodile and the distance that James could jump, you must tell him whether or not he can escape.

Input Specification:

Each input file contains one test case. Each case starts with a line containing two positive integers N (

≤ 100), the number of crocodiles, and D, the maximum distance that James could jump. Then N lines follow, each containing the (x, y) location of a crocodile. Note that no two crocodiles are staying at the same position.

Output Specification:

For each test case, print in a line "Yes" if James can escape, or "No" if not.

Sample Input 1:

```
14 20
25 -15
-25 28
8 49
29 15
-35 -2
5 28
27 -29
-8 -28
-20 -35
-25 -20
-13 29
-30 15
-35 40
12 12
```

Sample Output 1:

```
Yes
```

Sample Input 2:

```
4 13
-12 12
12 12
-12 -12
12 -12
```

Sample Output 2:

```
No
```

- 编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：DS课程组
  单位：浙江大学

06-图3

“六度空间”理论又称作“六度分隔（Six Degrees of Separation）”理论。这个理论可以通俗地阐述为：“你和任何一个陌生人之间所间隔的人不会超过六个，也就是说，最多通过五个人你就能够认识任何一个陌生人。”如图1所示。

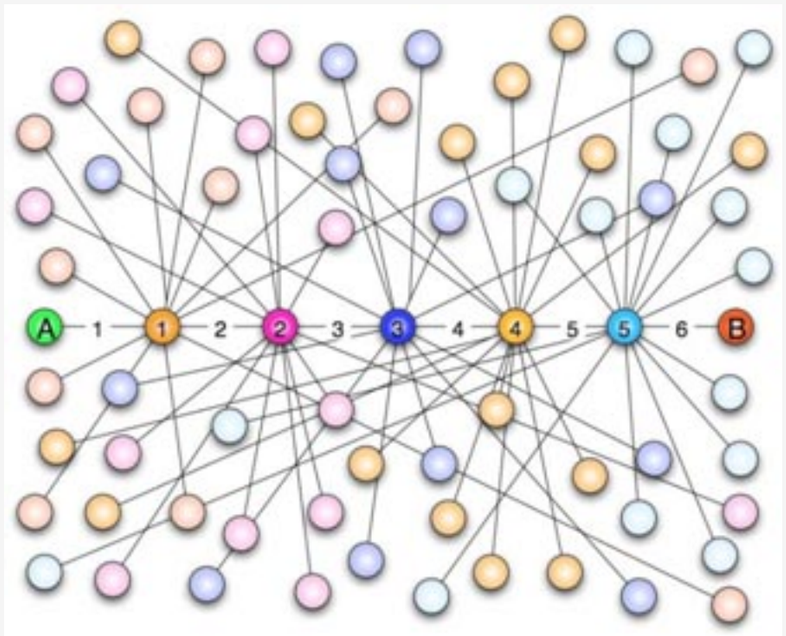

图1 六度空间示意图

“六度空间”理论虽然得到广泛的认同，并且正在得到越来越多的应用。但是数十年来，试图验证这个理论始终是许多社会学家努力追求的目标。然而由于历史的原因，这样的研究具有太大的局限性和困难。随着当代人的联络主要依赖于电话、短信、微信以及因特网上即时通信等工具，能够体现社交网络关系的一手数据已经逐渐使得“六度空间”理论的验证成为可能。

假如给你一个社交网络图，请你对每个节点计算符合“六度空间”理论的结点占结点总数的百分比。

输入格式:

输入第1行给出两个正整数，分别表示社交网络图的结点数N（$1 < N \leq 10^4$，表示人数）、边数M（$\leq 33 \times N$，表示社交关系数）。随后的M行对应M条边，每行给出一对正整数，分别是该条边直接连通的两个结点的编号（节点从1到N编号）。

输出格式:

对每个结点输出与该结点距离不超过6的结点数占结点总数的百分比，精确到小数点后2位。每个结节点输出一行，格式为“结点编号:（空格）百分比%”。

输入样例:

```
10 9
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
9 10
```

输出样例:

```
1: 70.00%
2: 80.00%
3: 90.00%
4: 100.00%
5: 100.00%
6: 100.00%
7: 100.00%
8: 90.00%
9: 80.00%
10: 70.00%
```

- 🎖
  编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：陈越
  单位：浙江大学

07-图4

哈利·波特要考试了，他需要你的帮助。这门课学的是用魔咒将一种动物变成另一种动物的本事。例如将猫变成老鼠的魔咒是haha，将老鼠变成鱼的魔咒是hehe等等。反方向变化的魔咒就是简单地将原来的魔咒倒过来念，例如ahah可以将老鼠变成猫。另外，如果想把猫变成鱼，可以通过念一个直接魔咒lalala，也可以将猫变老鼠、老鼠变鱼的魔咒连起来念：hahahehe。

现在哈利·波特的手里有一本教材，里面列出了所有的变形魔咒和能变的动物。老师允许他自己带一只动物去考场，要考察他把这只动物变成任意一只指定动物的本事。于是他来问你：带什么动物去可以让最难变的那种动物（即该动物变为哈利·波特自己带去的动物所需要的魔咒最长）需要的魔咒最短？例如：如果只有猫、鼠、鱼，则显然哈利·波特应该带鼠去，因为鼠变成另外两种动物都只需要念4个字符；而如果带猫去，则至少需要念6个字符才能把猫变成鱼；同理，带鱼去也不是最好的选择。

输入格式:

输入说明：输入第1行给出两个正整数N ($\leq$ 100)和M，其中N是考试涉及的动物总数，M是用于直接变形的魔咒条数。为简单起见，我们将动物按1~N编号。随后M行，每行给出了3个正整数，分别是两种动物的编号、以及它们之间变形需要的魔咒的长度($\leq$ 100)，数字之间用空格分隔。

输出格式:

输出哈利·波特应该带去考场的动物的编号、以及最长的变形魔咒的长度，中间以空格分隔。如果只带1只动物是不可能完成所有变形要求的，则输出0。如果有若干只动物都可以备选，则输出编号最小的那只。

输入样例:

```
6 11
3 4 70
1 2 1
5 4 50
2 6 50
5 6 60
1 3 70
4 6 60
3 6 80
5 1 100
2 4 60
5 2 80
```

输出样例:

```
4 70
```

- 🎖
  编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：陈越
  单位：浙江大学

07-图5

This time let us consider the situation in the movie "Live and Let Die" in which James Bond, the world's most famous spy, was captured by a group of drug dealers. He was sent to a small piece of land at the center of a lake filled with crocodiles. There he performed the most daring action to escape -- he jumped onto the head of the nearest crocodile! Before the animal realized what was happening, James jumped again onto the next big head... Finally he reached the bank before the last crocodile could bite him (actually the stunt man was caught by the big mouth and barely escaped with his extra thick boot).

Assume that the lake is a 100 by 100 square one. Assume that the center of the lake is at (0,0) and the northeast corner at (50,50). The central island is a disk centered at (0,0) with the diameter of 15. A

number of crocodiles are in the lake at various positions. Given the coordinates of each crocodile and the distance that James could jump, you must tell him a shortest path to reach one of the banks. The length of a path is the number of jumps that James has to make.

Input Specification:

Each input file contains one test case. Each case starts with a line containing two positive integers N (≤ 100), the number of crocodiles, and D, the maximum distance that James could jump. Then N lines follow, each containing the (x, y) location of a crocodile. Note that no two crocodiles are staying at the same position.

Output Specification:

For each test case, if James can escape, output in one line the minimum number of jumps he must make. Then starting from the next line, output the position (x, y) of each crocodile on the path, each pair in one line, from the island to the bank. If it is impossible for James to escape that way, simply give him 0 as the number of jumps. If there are many shortest paths, just output the one with the minimum first jump, which is guaranteed to be unique.

Sample Input 1:

```
17 15
10 -21
10 21
-40 10
30 -50
20 40
35 10
0 -10
-25 22
40 -40
-30 30
-10 22
0 11
25 21
25 10
10 10
10 35
-30 10
```

Sample Output 1:

```
4
0 11
10 21
10 35
```

Sample Input 2:

```
4 13
-12 12
12 12
-12 -12
12 -12
```

Sample Output 2:

```
0
```

- 编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：陈越
  单位：浙江大学

07-图6

有了一张自驾旅游路线图，你会知道城市间的高速公路长度、以及该公路要收取的过路费。现在需要你写一个程序，帮助前来咨询的游客找一条出发地和目的地之间的最短路径。如果有若干条路径都是最短的，那么需要输出最便宜的一条路径。

输入格式:

输入说明：输入数据的第1行给出4个正整数N、M、S、D，其中N（$2 \leq N \leq 500$）是城市的个数，顺便假设城市的编号为0~($N-1$)；M是高速公路的条数；S是出发地的城市编号；D是目的地的城市编号。随后的M行中，每行给出一条高速公路的信息，分别是：城市1、城市2、高速公路长度、收费额，中间用空格分开，数字均为整数且不超过500。输入保证解的存在。

输出格式:

在一行里输出路径的长度和收费总额，数字间以空格分隔，输出结尾不能有多余空格。

输入样例:

```
4 5 0 3
0 1 1 20
1 3 2 30
0 3 4 10
0 2 2 20
2 3 1 20
```

输出样例:

```
3 40
```

- 编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：陈越
  单位：浙江大学

08-图7

现有村落间道路的统计数据表中，列出了有可能建设成标准公路的若干条道路的成本，求使每个村落都有公路连通所需要的最低成本。

输入格式:

输入数据包括城镇数目正整数N（≤ 1000）和候选道路数目M（$\leq 3N$）；随后的M行对应M条道路，每行给出3个正整数，分别是该条道路直接连通的两个城镇的编号以及该道路改建的预算成本。为简单起见，城镇从1到N编号。

输出格式:

输出村村通需要的最低成本。如果输入数据不足以保证畅通，则输出-1，表示需要建设更多公路。

输入样例:

```
6 15
1 2 5
1 3 3
1 4 7
1 5 4
1 6 2
2 3 4
2 4 6
2 5 2
2 6 6
3 4 6
3 5 1
3 6 1
4 5 10
4 6 8
5 6 3
```

输出样例:

```
12
```

- 编译器：
  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc
  作者：陈越
  单位：浙江大学

08-图8

Given the relations of all the activities of a project, you are supposed to find the earliest completion time of the project.

Input Specification:

Each input file contains one test case. Each case starts with a line containing two positive integers N (≤ 100), the number of activity check points (hence it is assumed that the check points are numbered from 0 to $N-1$), and M, the number of activities. Then M lines follow, each gives the description of an activity. For the `i`-th activity, three non-negative numbers are given: `S[i]`, `E[i]`, and `L[i]`, where `S[i]` is the index of the starting check point, `E[i]` of the ending check point, and `L[i]` the lasting time of the activity. The numbers in a line are separated by a space.

Output Specification:

For each test case, if the scheduling is possible, print in a line its earliest completion time; or simply output "Impossible".

Sample Input 1:

```
9 12
0 1 6
0 2 4
0 3 5
1 4 1
2 4 1
3 5 2
5 4 0
4 6 9
4 7 7
5 7 4
6 8 2
7 8 4
```

Sample Output 1:

```
18
```

Sample Input 2:

```
4 5
0 1 1
0 2 2
2 1 3
1 3 4
3 2 5
```

Sample Output 2:

```
Impossible
```

-

  编译器：

  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc

  作者：DS课程组

  单位：浙江大学

08-图9

假定一个工程项目由一组子任务构成，子任务之间有的可以并行执行，有的必须在完成了其它一些子任务后才能执行。“任务调度”包括一组子任务、以及每个子任务可以执行所依赖的子任务集。

比如完成一个专业的所有课程学习和毕业设计可以看成一个本科生要完成的一项工程，各门课程可以看成是子任务。有些课程可以同时开设，比如英语和C程序设计，它们没有必须先修哪门的约束；有些课程则不可以同时开设，因为它们有先后的依赖关系，比如C程序设计和数据结构两门课，必须先学习前者。

但是需要注意的是，对一组子任务，并不是任意的任务调度都是一个可行的方案。比如方案中存在“子任务A依赖于子任务B，子任务B依赖于子任务C，子任务C又依赖于子任务A”，那么这三个任务哪个都不能先执行，这就是一个不可行的方案。

任务调度问题中，如果还给出了完成每个子任务需要的时间，则我们可以算出完成整个工程需要的最短时间。在这些子任务中，有些任务即使推迟几天完成，也不会影响全局的工期；但是有些任务必须准时完成，否则整个项目的工期就要因此延误，这种任务就叫“关键活动”。

请编写程序判定一个给定的工程项目的任务调度是否可行；如果该调度方案可行，则计算完成整个工程项目需要的最短时间，并输出所有的关键活动。

输入格式:

输入第1行给出两个正整数$N(\leq 100)$和M，其中N是任务交接点（即衔接相互依赖的两个子任务的节点，例如：若任务2要在任务1完成后才开始，则两任务之间必有一个交接点）的数量。交接点按1~N编号，M是子任务的数量，依次编号为1~M。随后M行，每行给出了3个正整数，分别是该任务开始和完成涉及的交接点编号以及该任务所需的时间，整数间用空格分隔。

输出格式:

如果任务调度不可行，则输出0；否则第1行输出完成整个工程项目需要的时间，第2行开始输出所有关键活动，每个关键活动占一行，按格式“V->W”输出，其中V和W为该任务开始和完成涉及的交接点编号。关键活动输出的顺序规则是：任务开始的交接点编号小者优先，起点编号相同时，与输入时任务的顺序相反。

输入样例:

```
7 8
1 2 4
1 3 3
2 4 5
3 4 3
4 5 1
4 6 6
5 7 5
6 7 2
```

输出样例:

```
17
1->2
2->4
4->6
6->7
```

-

  编译器：

  gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbnc

  作者：陈越

  单位：浙江大学

09-排序1

给定N个（长整型范围内的）整数，要求输出从小到大排序后的结果。

本题旨在测试各种不同的排序算法在各种数据情况下的表现。各组测试数据特点如下：

- 数据1：只有1个元素；
- 数据2：11个不相同的整数，测试基本正确性；
- 数据3：10^3个随机整数；
- 数据4：10^4个随机整数；
- 数据5：10^5个随机整数；
- 数据6：10^5个顺序整数；
- 数据7：10^5个逆序整数；

- 数据8：10^5个基本有序的整数；
- 数据9：10^5个随机正整数，每个数字不超过1000。

输入格式:
输入第一行给出正整数N（$\leq 10^5$），随后一行给出N个（长整型范围内的）整数，其间以空格分隔。
输出格式:
在一行中输出从小到大排序后的结果，数字间以1个空格分隔，行末不得有多余空格。
输入样例:
```
11
4 981 10 -17 0 -20 29 50 8 43 -5
```
输出样例:
```
-20 -17 -5 0 4 8 10 29 43 50 981
```

编译器：gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbr

作者：陈越

单位：浙江大学

09-排序2

According to Wikipedia:

Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no inpu elements remain.

Merge sort works as follows: Divide the unsorted list into N sublists, each containing 1 element (a list of 1 element is considered sorte Then repeatedly merge two adjacent sublists to produce new sorted sublists until there is only 1 sublist remaining.

Now given the initial sequence of integers, together with a sequence which is a result of several iterations of some sorting method, can tell which sorting method we are using?

Input Specification:

Each input file contains one test case. For each case, the first line gives a positive integer N (≤ 100). Then in the next line, N integers ar given as the initial sequence. The last line contains the partially sorted sequence of the N numbers. It is assumed that the target sequen is always ascending. All the numbers in a line are separated by a space.

Output Specification:

For each test case, print in the first line either "Insertion Sort" or "Merge Sort" to indicate the method used to obtain the partial result. T run this method for one more iteration and output in the second line the resuling sequence. It is guaranteed that the answer is unique f each test case. All the numbers in a line must be separated by a space, and there must be no extra space at the end of the line.

Sample Input 1:

```
10
3 1 2 8 7 5 9 4 6 0
1 2 3 7 8 5 9 4 6 0
```

Sample Output 1:

```
Insertion Sort
1 2 3 5 7 8 9 4 6 0
```

Sample Input 2:

```
10
3 1 2 8 7 5 9 4 0 6
1 3 2 8 5 7 4 9 0 6
```

Sample Output 2:

```
Merge Sort
1 2 3 8 4 5 7 9 0 6
```

-

编译器：gcc,g++,javac,python3,python2,awk,clisp,clang,clang++,mcs,dmd,gccgo,gcj,go,ghc,luajit,node,lua,ocamlc,fpc,php,perl,ruby,racket,bash,cat,valac,vbr

作者：陈越

单位：浙江大学

09-排序3

According to Wikipedia:

Insertion sort iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration, insertion sort removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no inpu elements remain.

Heap sort divides its input into a sorted and an unsorted region, and it iteratively shrinks the unsorted region by extracting the largest element and moving that to the sorted region. it involves the use of a heap data structure rather than a linear-time search to find the maximum.

Now given the initial sequence of integers, together with a sequence which is a result of several iterations of some sorting method, can tell which sorting method we are using?

Input Specification:

Each input file contains one test case. For each case, the first line gives a positive integer N (≤ 100). Then in the next line, N integers ar given as the initial sequence. The last line contains the partially sorted sequence of the N numbers. It is assumed that the target sequen is always ascending. All the numbers in a line are separated by a space.

Output Specification:

For each test case, print in the first line either "Insertion Sort" or "Heap Sort" to indicate the method used to obtain the partial result. Th
run this method for one more iteration and output in the second line the resuling sequence. It is guaranteed that the answer is unique f
each test case. All the numbers in a line must be separated by a space, and there must be no extra space at the end of the line.

Sample Input 1:

```
10
3 1 2 8 7 5 9 4 6 0
1 2 3 7 8 5 9 4 6 0
```

Sample Output 1:

```
Insertion Sort
1 2 3 5 7 8 9 4 6 0
```

Sample Input 2:

```
10
3 1 2 8 7 5 9 4 6 0
6 4 5 1 0 3 2 7 8 9
```

Sample Output 2:

```
Heap Sort
5 4 3 1 0 2 6 7 8 9
```

01-复杂度1 最大子列和问题　(20分)

时间限制：50000m
内存限制：64MB
代码长度限制：16k
判题程序：系统默认
作者：DS课程组
单位：浙江大学

给定K个整数组成的序列{ N_1, N_2, ..., N_K }，“连续子列”被定义为{ N_i, N_{i+1}, ..., N_j }，其中$1 \le i \le j \le K$。“最大子列和”则被定义为所有连续子列元素的和中最大者。例如给定序列{ -2, 11, -4, 13, -5, -2 }，其连续子列{ 11, -4, 13 }有最大的和20。现要求你编写程序，计算给定整数序列的最大子列和。

本题旨在测试各种不同的算法在各种数据情况下的表现。各组测试数据特点如下：

- 数据1：与样例等价，测试基本正确性；
- 数据2：10^2个随机整数；
- 数据3：10^3个随机整数；
- 数据4：10^4个随机整数；
- 数据5：10^5个随机整数；

输入格式:

输入第1行给出正整数K (≤ 100000)；第2行给出K个整数，其间以空格分隔。

输出格式:

在一行中输出最大子列和。如果序列中所有整数皆为负数，则输出0。

输入样例:

```
6
-2 11 -4 13 -5 -2
```

输出样例:

```
20
```

题目判定

解题程序

编译器： * gcc

程序代码： *

该题目集还未添加任何题目