**Bài 1: cài đặt thuật toán**

* Quick-sort
* Bucket Sort
* Counting sort
* Merge Sort

**Bài 2: Cặp điểm gần nhất (Closest pair)**

Tìm cặp điểm gần nhất trong mặt phẳng.

Dữ liệu vào từ file “CLOSEST.INP”

* Dòng đầu: giá trị n số điểm thuộc mặt phẳng.
* Các dòng tiếp: là các cặp giá trị thể hiện tọa độ các điểm thuộc mặt phẳng ().

Kết quả ghi ra file “CLOSEST.OUT” khoảng cách gần nhất của 1 cặp điểm thuộc mặt phẳng.

Ràng buộc: là các số nguyên.

|  |  |
| --- | --- |
| **CLOSEST.INP** | **CLOSEST.OUT** |
| 11  4 4  -2 2  -3 -4  -1 3  2 3  -4 0  1 1  -1 -1  3 -1  -4 2  -2 4 | 1.414213 |

**Bài 3: Mã nhân viên**

Tổng Giám đốc công ty X nổi tiếng là một người kỹ lưỡng. Ông ta thực hiện việc quản lý nhân viên của mình bằng cách gán cho mỗi nhân viên một mã số khác nhau. Công ty có N nhân viên, nhân viên i (i=1,2,…,N) có mã số . Do bận đi công tác nước ngoài một thời gian dài nên ông trao quyền quản lý công ty cho một người khác. Khi ông trở về, công ty đã có sự thay đổi về số lượng nhân viên. Khi tiếp nhận thêm nhân viên mới, ông yêu cầu muốn biết mã số nhỏ nhất có thể gán cho nhân viên mới.

**Yêu cầu:** Cho N mã số của nhân viên trong công ty. Hãy tìm mã số nhỏ nhất chưa xuất hiện trong N mã số đã cho.

Dữ liệu: vào cho trong tệp văn bản “MASO.IN”. Dòng đầu là số N (1<N<=30000). N dòng tiếp theo, dòng thứ i ghi số .

Kết quả ghi vào tệp văn bản “MASO.OUT” một số duy nhất là mã số tìm được.

|  |  |
| --- | --- |
| MASO.INP | MASO.OUT |
| 6  7  5  6  1  3  4 | 2 |

**Bài 4: Xếp trứng**

Cho N quả trứng được đưa vào dây chuyền theo thứ tự (quả trứng thứ i có thể tích là ai). Ở cuối dây chuyền đã có sẵn **M** thùng chứa trứng. Các thùng này nhận trứng theo quy tắc: chứa trứng cho đến khi đầy thì chuyển sang thùng khác. Hãy tính sức chứa K tối thiểu của mỗi thùng để M thùng này có thể chứa hết trứng theo quy trình trên.



**Dữ liệu**

- Dòng đầu: Ghi 2 số nguyên n, m (0 < n < 106, 0 < m ≤ 109)

- Các dòng tiếp theo: dãy ai (0 < ai ≤ 106).

**Kết quả**

- Một số duy nhất là số **k** tìm được.

**Ví dụ**

**input**

5 3  
6  
5  
4  
8  
9

**output**

12