

ข้อสอบปฏิบัติครั้งที่ 1

ค่ายคัดเลือกคอมพิวเตอร์โอลิมปิก ครั้งที่ 1 ประจำปี 2554

วันที่สอบ 19 ตุลาคม 2553

แปลงดอกไม้แสนสวยของเด็กหญิงพอเพียง (oct19_flower)

เด็กหญิงพอเพียงปลูกดอกไม้ในสวนขนาดกว้าง m เมตรและสูง n เมตร โดยแบ่งที่ดินออกเป็นแปลง แต่ละแปลงมีพื้นที่ 1 ตารางเมตร และที่สำคัญเพื่อให้สามารถเข้าไปชื่นชมความงามของดอกไม้ได้อย่างใกล้ชิด จึงปลูกดอกไม้ไว้แปลงละ 1 ดอกเท่านั้น

เด็กหญิงพอเพียงหารายได้มาเลี้ยงครอบครัว โดยการเก็บดอกไม้ในสวนของตนเองไปขาย ซึ่งดอกไม้ในแต่ละแปลงจะมีราคาของตัวมันเองอยู่ และราคานี้จะซ้ำกันกับดอกไม้ในแปลงอื่นๆ ก็ได้ ถ้าหากวันไหนโชคดีมีฝนตกลงในแปลงดอกไม้ ดอกไม้ในแปลงนั้นจะมีราคาเพิ่มขึ้นเป็นสองเท่าของราคาก่อนฝนตก แต่ราคานี้จะไม่มากไปกว่าราคาดอกไม้สูงสุดที่จะเป็นไปได้ แต่ถ้าหากวันไหนโชคร้ายมีหนอนในไร่ยาмаกินดอกไม้ของเด็กหญิงพอเพียง ดอกไม้จะมีมูลค่าลดลงเหลือเพียงหนึ่งในสามของราคา ดอกไม้ก่อนจะถูกหนอนกิน และมูลค่าดอกไม้นี้จะไม่ลดลงน้อยกว่า 1 บาท

เมื่อใดก็ตามที่เด็กหญิงพอเพียงเก็บดอกไม้ในสวนไปขาย เด็กหญิงพอเพียงเก็บดอกไม้ที่มีราคาสูงสุดจำนวน k ดอกเพื่อนำไปขาย และจะซื้อดอกไม้มาจากตลาดกลับมาปลูกในสวนดอกไม้ของตนเอง k ดอกเช่นกัน โดยดอกไม้ที่เด็กหญิงพอเพียงซื้อกลับมาปลูกนี้ จะมีราคาเท่ากันหมดทั้ง k ดอก

งานของคุณ

ให้เขียนโปรแกรมเพื่อรับอินพุตจาก standard input แล้วให้แสดงผลพร้อมออกทางหน้าจอเป็นราคารวมของดอกไม้ที่เด็กหญิงพอเพียงนำไปขายทั้งหมด โดยโปรแกรมจะต้องทำงานรวมทั้งหมดไม่เกิน $\frac{1}{2}$ วินาที

ข้อกำหนด

- m และ n มีค่าไม่เกิน 1,000 เมตร
- ดอกไม้มีราคาไม่เกิน 5,000,000 บาท (ห้าล้านบาทถ้วน) ($\text{max_price} = 5000000$)
- หากฝนตกลงมาในแปลงดอกไม้ ดอกไม้ในแปลงนั้นจะมีราคาเท่ากับ $2 \times p$ เมื่อ p คือ ราคาของดอกไม้ก่อนฝนตก และถ้า $2 \times p$ มีค่ามากกว่า max_price ให้กำหนดราคาดอกไม้หลังฝนตกเท่ากับ max_price
- หากมีหนอนมากินดอกไม้ในแปลงไหน ให้ลดราคาของดอกไม้ในแปลงนั้นเป็น $\text{floor}(p/3)$ เมื่อ p คือ ราคาดอกไม้ก่อนหนอนมากิน ถ้า $\text{floor}(p/3)$ มีค่าน้อยกว่า 1 ให้กำหนดราคาดอกไม้หลังหนอนมากินเป็น 1
- ในตอนเริ่มต้น กำหนดให้ มีดอกไม้เต็มทุกแปลงในสวน และดอกไม้แต่ละดอกมีราคาเท่ากับ $r * m + c + 1$ เมื่อ r และ c คือ row และ column ของแปลงดอกไม้ตามลำดับ และ m คือ ความกว้างของสวนดอกไม้
- r จะมีค่าอยู่ในช่วง $[0, n-1]$ และ c จะมีค่าอยู่ในช่วง $[0, m-1]$
- operation ทั้งสามอย่าง (มีฝนตก หนอนกินดอกไม้ และตัดดอกไม้ไปขาย) รวมกันไม่เกิน 101 ครั้ง และจำนวนครั้งของการตัดไปขายจะมีค่าโดยประมาณ 20 ครั้ง ในแต่ละครั้งของการตัดดอกไม้ไปขาย จะตัดดอกไม้ไม่เกิน 10 ดอก และดอกไม้ที่นำมาปลูกทดแทนจะมีราคาไม่เกิน $m \times n$ บาท
- หากดอกไม้มีราคาเท่ากัน ให้ตัดดอกไม้ที่อยู่ใน row ที่มีค่าน้อยกว่า ถ้า row เท่ากัน ให้ตัดดอกไม้ที่อยู่ใน column น้อยกว่าก่อน

ข้อมูลเข้า

- บรรทัดแรกมีค่าสองค่า คือ n และ m
- บรรทัดต่อไป จะแบ่งออกเป็น 3 กรณี ตามตัวอักษรหน้าสุดของบรรทัดนั้น
 - ถ้าอักษรตัวแรกเป็น 'R' จะมีตัวเลขจำนวนเต็มตามมามากี่จำนวนคือ ตำแหน่ง row เริ่มต้น ตำแหน่ง column เริ่มต้น ตำแหน่ง row สุดท้าย และ ตำแหน่ง column สุดท้าย ตามลำดับ โดยมีความหมายว่า มีฝนตกลงในแปลงดอกไม้ที่มีตำแหน่งตั้งแต่ row เริ่มต้น ไปจนถึง row สุดท้าย และ column เริ่มต้นไปจนถึง column สุดท้าย
 - ถ้าอักษรตัวแรกเป็น 'W' จะมีตัวเลขจำนวนเต็มตามมามากี่จำนวนคือ ตำแหน่ง row เริ่มต้น ตำแหน่ง column เริ่มต้น ตำแหน่ง row สุดท้าย และ ตำแหน่ง column สุดท้าย ตามลำดับ โดยมีความหมายว่า มีหนอนมากินดอกไม้ในแปลงดอกไม้ที่มีตำแหน่งตั้งแต่ row เริ่มต้น ไปจนถึง row สุดท้าย และ column เริ่มต้นไปจนถึง column สุดท้าย
 - ถ้าตัวอักษรตัวแรกเป็น 'S' จะมีตัวเลขตามมามากี่สองจำนวน คือ จำนวนดอกไม้ที่เด็กหญิงพอเพียงนำไปขาย และราคาของดอกไม้ที่เด็กหญิงพอเพียงนำมาปลูกใหม่

ตัวอย่างข้อมูลนำเข้า

```
10 10
W 8 8 9 9
R 2 4 2 4
S 3 25
W 8 6 9 9
S 5 35
```

ตัวอย่างข้อมูลส่งออก

756

ตัวอย่างของแต่ละ Operation

10 10

สร้างแปลงดอกไม้ขนาด 10 x 10 โดยแต่ละแปลงมีดอกไม้ตามราคาดังต่อไปนี้

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	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

W 8 8 9 9

มีหนอนมากินดอกไม้ ทำให้ราคาของดอกไม้เปลี่ยนเป็น

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	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	29	30
91	92	93	94	95	96	97	98	33	33

R 2 4 2 4

มีฝนตกทำให้ราคาของดอกไม้เปลี่ยนเป็น

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	29	30
91	92	93	94	95	96	97	98	33	33

S 3 25

เด็กหญิงพอเพียงนำดอกไม้จำนวน 3 ดอก ที่มีราคาสูงสุดไปขาย

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	29	30
91	92	93	94	95	96	97	98	33	33

เด็กหญิงพอเพียงปลูกดอกไม้ราคา 25 บาท ทดแทนดอกที่นำไปขาย

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	29	30
91	92	93	94	95	25	25	25	33	33

พ 8 6 9 9

มีहनอนมากินดอกไม้ ทำให้ราคาดอกไม้เปลี่ยนเป็น

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	29	29	9	10
91	92	93	94	95	25	8	8	11	11

S 5 35

เด็กหญิงพอเพียงนำดอกไม้จำนวน 5 ดอก ที่มีราคาสูงสุดไปขาย

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	29	29	9	10
91	92	93	94	95	25	8	8	11	11

เด็กหญิงพอเพียงปลูกดอกไม้ราคา 35 บาท ทดแทนดอกไม้ที่นำไปขาย

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	50	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	29	29	9	10
35	35	35	35	35	25	8	8	11	11

เด็กหญิงพอเพียงนำดอกไม้ไปขายสองครั้ง ครั้งละ 3 ดอก และ 5 ดอก ตามลำดับ รวมรายได้ที่ได้จากการขายดอกไม้เท่ากับ

$$98 + 97 + 96 + 95 + 94 + 93 + 92 + 91 = 756$$

ข้อสอบปฏิบัติครั้งที่ 1

ค่ายคัดเลือกคอมพิวเตอร์โอลิมปิก ครั้งที่ 1 ประจำปี 2554

วันที่สอบ 19 ตุลาคม 2553

สองอะเรย์ (oct19_twoarrays)

กำหนดอะเรย์สองอะเรย์ a และ b ซึ่งแต่ละตัวมี n ช่อง แต่ละช่องบรรจุตัวแปรประเภท int หนึ่งตัว และข้อมูลในอะเรย์ทั้งสองเรียงจากน้อยไปหามากแล้ว

งานของคุณ

จงเขียนฟังก์ชัน $\text{int select}(\text{int } k, \text{int } n)$ ซึ่งคืนตัวเลขที่มีขนาดเล็กที่สุดเป็นอันดับที่ k ($1 \leq k \leq 2n$) ในทั้งสองอะเรย์มา โดยฟังก์ชันที่คุณเขียนสามารถเรียกใช้งานฟังก์ชันได้สองฟังก์ชัน

- $\text{int read_a}(\text{int } i)$ เมื่อ i มีค่าตั้งแต่ 0 ถึง $n-1$ โดยฟังก์ชันนี้จะคืนค่า $a[i]$ กลับมาให้
- $\text{int read_b}(\text{int } i)$ เมื่อ i มีค่าตั้งแต่ 0 ถึง $n-1$ โดยฟังก์ชันนี้จะคืนค่า $b[i]$ กลับมาให้

ข้อกำหนด

- $2 \leq n \leq 1,000,000$
- ฟังก์ชัน select ที่คุณเขียนจะต้องเรียกฟังก์ชัน read_a และ read_b รวมแล้วไม่เกิน 100 ครั้ง
- คุณจะต้องเขียนโปรแกรมเป็นภาษา C++ เท่านั้น
- ไฟล์ที่คุณส่งต้องมีฟังก์ชัน select ตามที่กำหนดข้างบนและไม่มีฟังก์ชัน main
- ไฟล์ที่คุณส่งจะต้องมีข้อความ `#include "twoarrays.h"` อยู่ข้างใน
- ห้ามเขียนและอ่านข้อมูลจากไฟล์หรือ standard input
- ในแต่ละชุดข้อมูลทดสอบ โปรแกรมตัวตรวจจะเรียกฟังก์ชัน select ของคุณเพียงครั้งเดียวเท่านั้น
- ฟังก์ชัน select ของคุณรวมกับโค้ดของตัวตรวจ จะต้องทำงานภายในเวลา 10 วินาที และใช้หน่วยความจำไม่เกิน 32 MB (ส่วนของตัวตรวจจะใช้เวลาทำงานไม่เกิน 5 วินาที และใช้หน่วยความจำไม่เกิน 16MB)

โค้ดสำหรับทดลอง

คุณสามารถดาวน์โหลดโค้ดสำหรับทดลองได้จากเว็บตัวตรวจ ซึ่งมันอยู่ในไฟล์ `twoarrays.zip` ซึ่งเมื่อขยายออกแล้วจะมีไฟล์ต่อไปนี้ข้างใน

- `twoarrays.h` ซึ่งมี prototype ของ read_a และ read_b
- `twoarrays.cpp` ซึ่งมีตัวอย่างฟังก์ชัน select ที่ทำงานได้ถูกต้อง แต่เรียก read_a กับ read_b มากไปอยู่ ไฟล์นี้เป็นไฟล์ที่คุณควรแก้ไขแล้วส่งเข้า grader
- `main.cpp` มีฟังก์ชัน main ซึ่งทำหน้าที่อ่าน input, เรียก select , และเขียน output นอกจากนี้ยังมีตัวอย่างฟังก์ชัน read_a และ read_b ด้วย

ฟังก์ชัน `main` จะอ่านข้อมูลเข้าจากทาง `standard input` โดยรูปแบบที่อ่านข้อมูลเป็นดังนี้

- บรรทัดแรกมีจำนวนเต็ม n
- บรรทัดที่สองมีจำนวนเต็ม n จำนวน แทนสมาชิกของ a เรียงจากหน้าไปหลัง
- บรรทัดที่สามมีจำนวนเต็ม n จำนวน แทนสมาชิกของ b เรียงจากหน้าไปหลัง
- บรรทัดสุดท้ายมีจำนวนเต็ม k โดยที่ $1 \leq k \leq n$

หลังจากนั้นมันจะเรียก `select(k, n)` แล้วพิมพ์ผลลัพธ์ที่ได้ออกทางหน้าจอ

อนึ่ง คุณสามารถคอมไพล์โปรแกรมโดยการสั่ง

```
g++ main.cpp twoarrays.cpp -O2
```

ตัวอย่างข้อมูลนำเข้าของโค้ดตัวอย่าง

```
3
1 5 9
2 4 10
5
```

(หมายเหตุ: สำหรับ input นี้ `select` ควรจะคืนค่า 9)

ข้อควรระวัง

- โปรแกรมตัวอย่างจะไม่เช็คว่าคุณเรียก `read_a` และ `read_b` รวมกันเกิน 100 ครั้งหรือไม่ ถ้าอยากให้เช็คให้ไปเขียนโค้ดเอง
- คุณสามารถตรวจสอบว่าไฟล์ที่ส่งไปคอมไพล์กับโค้ดของกรรมการได้จากการดู `compiler message` เมื่อส่งไฟล์ แต่กรรมการจะไม่ติดตั้ง `test interface` ให้คุณ
- เนื่องจากปัญหานี้เน้นจำนวนครั้งการเรียก `read_a` และ `read_b` และได้ให้เวลาในการทำงานไว้มาก จึงไม่มีประโยชน์ที่คุณจะใช้ `test interface` ในการทดสอบโปรแกรมแต่อย่างใด

ข้อสอบปฏิบัติครั้งที่ 1

ค่ายคัดเลือกคอมพิวเตอร์โอลิมปิก ครั้งที่ 1 ประจำปี 2554

วันที่สอบ 19 ตุลาคม 2553

ใหญ่กับฟ้า (oct19_majority)

กำหนดให้มีอาร์เรย์ A ซึ่งมี n ช่อง ถ้าข้อมูลในช่องใดในอาร์เรย์ดังกล่าว ปรากฏซ้ำใน A เกินครึ่ง ข้อมูลช่องนั้นจะถูกเรียกว่า “ตัวหุ้มมาก” ตัวอย่างเช่น $\{1, 1, 2, 1, 3, 1, 4\}$ มี 1 เป็นตัวหุ้มมาก แต่ $\{1, 2, 2, 1, 0, 1, 1, 0\}$ ไม่มีตัวหุ้มมาก

งานของคุณ

จงเขียนฟังก์ชัน `int major(int n)` ซึ่งคืนค่า -1 ถ้าใน A ไม่มีตัวหุ้มมาก และคืนค่า “ตำแหน่งของตัวหุ้มมาก” ใน A ในกรณีที่ A มีตัวหุ้มมาก โดยฟังก์ชันที่คุณเขียนนั้น ไม่สามารถอ่านค่าของตัวแปรในอาร์เรย์ได้ แต่จะสามารถถามว่า ข้อมูลในอาร์เรย์สองช่องใด ๆ มีค่าเหมือนกันหรือไม่ ได้ โดยคุณสามารถเรียกใช้งานฟังก์ชันต่อไปนี้ได้

- `bool same(int x, int y)` เมื่อ x, y มีค่าตั้งแต่ 0 ถึง $n-1$ โดยฟังก์ชันนี้จะคืนค่า `true` ถ้า $A[x] == A[y]$ แต่จะคืนค่า `false` ในกรณีอื่น ๆ

ข้อกำหนด

- $1 \leq n \leq 1,000$
- ถ้าคุณตอบถูก คะแนนที่ได้จะแปรตามจำนวนครั้งที่คุณเรียกฟังก์ชัน `same`
 - ถ้าฟังก์ชัน `major` เรียกใช้ `same` รวมแล้วไม่เกิน $4*n$ ครั้งจะได้คะแนนเต็ม
 - ถ้าฟังก์ชัน `major` เรียกใช้ `same` รวมแล้วไม่เกิน $4*n \log_2 n$ ครั้งจะได้คะแนน 60%
 - ถ้าฟังก์ชัน `major` เรียกใช้ `same` รวมแล้วไม่เกิน $n * (n - 1) / 2$ ครั้งจะได้คะแนน 20%
 - ในกรณีอื่น ๆ จะไม่ได้
- คุณจะต้องเขียนโปรแกรมเป็นภาษา C++ เท่านั้น
- ไฟล์ที่คุณส่งต้องมีฟังก์ชัน `select` ตามที่กำหนดข้างบนและไม่มีฟังก์ชัน `main`
- ไฟล์ที่คุณส่งจะต้องมีข้อความ `#include "major.h"` อยู่ข้างใน
- ห้ามเขียนและอ่านข้อมูลจากไฟล์หรือ `standard input`
- ในแต่ละชุดข้อมูลทดสอบ โปรแกรมตัวตรวจจะเรียกฟังก์ชัน `select` ของคุณเพียงครั้งเดียวเท่านั้น
- ฟังก์ชัน `select` ของคุณรวมกับโค้ดของตัวตรวจ จะต้องทำงานภายในเวลา 10 วินาที และใช้หน่วยความจำไม่เกิน 32 MB (ส่วนของตัวตรวจจะใช้เวลาทำงานไม่เกิน 5 วินาที และใช้หน่วยความจำไม่เกิน 16MB)

โค้ดสำหรับทดลอง

คุณสามารถดาวน์โหลดโค้ดสำหรับทดลองได้จากเว็บตัวตรวจ ซึ่งมันอยู่ในไฟล์ major.zip ซึ่งเมื่อขยายออกแล้วจะมีไฟล์ต่อไปนี้อยู่ข้างใน

- major.h ซึ่งมี prototype ของ same และ same
- major.cpp ซึ่งมีตัวอย่างฟังก์ชัน select ที่ทำงานได้ถูกต้อง แต่เรียก same มากไปอยู่ ไฟล์นี้เป็นไฟล์ที่คุณควรแก้ไขแล้วส่งเข้า grader
- main.cpp มีฟังก์ชัน main ซึ่งทำหน้าที่อ่าน input, เรียก major, และเขียน output นอกจากนี้ยังมีตัวอย่างฟังก์ชัน same ด้วย

ฟังก์ชัน main จะอ่านข้อมูลเข้าจากทาง standard input โดยรูปแบบที่อ่านข้อมูลเป็นดังนี้

- บรรทัดแรกมีจำนวนเต็ม n
- บรรทัดถัดมามีข้อมูล n ตัวซึ่งระบุถึงข้อมูลในอาร์เรย์

หลังจากนั้นมันจะเรียก major(n) แล้วพิมพ์ผลลัพธ์ที่ได้ออกทางหน้าจอ

อนึ่ง คุณสามารถคอมไพล์โปรแกรมโดยการสั่ง

```
g++ main.cpp major.cpp -O2
```

ตัวอย่างข้อมูลนำเข้าของโค้ดตัวอย่าง

8

1 1 1 1 2 2 2 1

(หมายเหตุ: สำหรับ input นี้ same จะคืนค่า 0 หรือ 1 หรือ 2 หรือ 3 หรือ 7 ก็ได้)

ข้อควรระวัง

- โปรแกรมตัวอย่างจะไม่เชื่อกว่าคุณเรียก same ถ้าอยากให้เช็คให้ไปเขียนโค้ดเอง
- คุณสามารถตรวจสอบว่าไฟล์ที่ส่งไปคอมไพล์กับโค้ดของกรรมการได้จากการดู compiler message เมื่อส่งไฟล์ แต่กรรมการจะไม่ติดตั้ง test interface ให้คุณ
- เนื่องจากปัญหานี้เน้นจำนวนครั้งการเรียก same และได้ให้เวลาในการทำงานไว้มาก จึงไม่มีประโยชน์ที่คุณจะใช้ test interface ในการทดสอบโปรแกรมแต่อย่างใด

ข้อสอบปฏิบัติครั้งที่ 1

ค่ายคัดเลือกคอมพิวเตอร์โอลิมปิก ครั้งที่ 1 ประจำปี 2554

วันที่สอบ 19 ตุลาคม 2553

ลำดับสับสน (oct19_sds)

“ลำดับอธิบายตัวเอง” (Self-Describing Sequence) คือลำดับอนันต์ของตัวเลขจำนวนเต็มบวก a_1, a_2, a_3, \dots ซึ่งมีลักษณะพิเศษคือ มันจะมีตัวเลข i อยู่ a_i ตัวพอดี และให้สังเกตว่าลำดับนี้จะเรียงจากน้อยไปมากอยู่แล้ว สมาชิกลำดับต้น ๆ ของลำดับนี้เป็นดังนี้

a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	2	2	3	3	4	4	4	5	5

งานของคุณ

คำนวณหาค่าของสมาชิกในตำแหน่งต่าง ๆ ของ “ลำดับอธิบายตัวเอง”

ข้อมูลเข้า

บรรทัดแรกมีค่า N ซึ่งระบุถึงจำนวนสมาชิกของลำดับที่ต้องการทราบค่า ($1 \leq N \leq 1000$) อีก N บรรทัดถัดมาเก็บค่า x_i ซึ่งเป็นค่าระบุตำแหน่งของสมาชิกที่ต้องการทราบค่า โดยที่ค่าของแต่ละตำแหน่งนั้นจะอยู่ในช่วง 1 ถึง 2,000,000,000

ข้อมูลออก

ข้อมูลออกมีทั้งหมด N บรรทัด แต่ละบรรทัดจะเป็นค่าของ a_{x_i}

ตัวอย่าง

Input	Output
3	2
2	3
4	5
10	

Ex2

Input	Output
4	21
100	356
9999	1684
123456	438744
1000000000	