Souvenirs

عمار يريد أن يشتري قطع تذكارية من محل أجنبي

يوجد N نوع من القطع التذكارية كما أنه يوجد مالانهاية من القطع التذكارية لكل نوع متوفر في المحل

كل نوع من القطع التذكارية لديه سعر ثابت أي أن التذكار من النوع i $(0 \leq i < N)$ سعره $P[i]$ دينار, حيث $P[i]$ هو عدد صحيح موجب تماما.

يعرف عمار أن أنواع القطع التذكارية مرتبة ترتيبًا تنازليًا حسب السعر, وأن أسعار القطع التذكارية مختلفة مثنى مثنى. على وجه التحديد، $P[0] > P[1] > \cdots > P[N-1] > 0$. كذلك، تمكن من معرفة قيمة $P[0]$. لسوء الحظ، ليس لدى عمار أي معلومات أخرى عن أسعار القطع التذكارية

لشراء بعض القطع التذكارية، سيجري عمار عدداً من المعاملات مع البائع.

تتكون كل معاملة من الخطوات التالية:

1. يسلم عمار بعض الدنانير (الموجبة) للبائع.
2. يضع البائع هذه الدنانير في كومة على طاولة في الغرفة الخلفية, حيث لا يستطيع عمار رؤيتها.
3. يعتبر البائع كل نوع من القطع التذكارية $0, 1, \ldots, N-1$ بهذا الترتيب، واحدًا تلو الآخر. يتم النظر في كل نوع **مرة واحدة بالضبط**.
   - عند إعتبار نوع القطع التذكارية i, إذا كان العدد الحالي للدنانير في الكومة هو $P[i]$ على الأقل، فإن
     - البائع يزيل $P[i]$ دنانير من الكومة، و
     - يضع البائع قطعة تذكارية واحدة من النوع i على الطاولة.
4. يأخذ البائع جميع الدنانير المتبقية في الكومة وجميع القطع التذكارية على الطاولة ويقدمها إلى عمار.

لاحظ أنه لا توجد دنانير أو قطع تذكارية على الطاولة قبل بدء المعاملة. مهمتك هي أن توجه عمار بإجراء عدد من المعاملات، بحيث:

- في كل معاملة يشتري **قطعة تذكارية واحدة على الأقل**، و
- بشكل عام يشتري **بالضبط** i قطعة تذكارية من النوع i, لكل i بحيث $0 \leq i < N$. لاحظ أن هذا يعني أنه لا ينبغي أن يشتري عمار أي قطعة تذكارية من النوع 0.

لا يتعين على عمار تقليل عدد المعاملات ولديه مخزون غير محدود من الدنانير.

Implementation Details

You should implement the following procedure.

```
void buy_souvenirs(int N, long long P0)
```

- N: عدد أنواع القطع التذكارية.

- $P0$: قيمة $P[0]$.
- يتم استدعاء هذا الإجراء مرة واحدة فقط لكل حالة اختبار.

يمكن للإجراء أعلاه إجراء مكالمات إلى الإجراء التالي لتوجيه عمار لتنفيذ معاملة:

```
std::pair<std::vector<int>, long long> transaction(long long M)
```

- M:عدد الدنانير التي سلمها عمار للبائع.

- يُرجع الإجراء ثنائية العنصر الأول من الثنائية هو مصفوفة L, تحتوي على أنواع القطع التذكارية التي تم شراؤها (بترتيب تصاعدي). العنصر الثاني هو عدد صحيح R, عدد العملات التي تم إرجاعها إلى عمار بعد المعاملة.

- يشترط أن $P[0] > M \geq P[N-1]$. يضمن الشرط $P[0] > M$ ألا يشتري عمار أي قطعة تذكارية من النوع 0 ، و $M \geq P[N-1]$ يضمن أن يشتري عمار قطعة تذكارية واحدة على الأقل. إذا لم تتحقق هذه الشروط، فإن الحل الخاص بك سيحصل على الحكم `Output isn't correct: Invalid argument`. لاحظ أنه على عكس $P[0]$ ، لم يتم توفير قيمة $P[N-1]$ في المدخلات.

- يمكن استدعاء الإجراء 5000 مرة على الأكثر في كل حالة اختبار.

إن سلوك مصنف الدرجات **غير قابل للتكيف**. وهذا يعني أن تسلسل الأسعار P ثابت قبل استدعاء `buy_souvenirs`.

Constraints

- $2 \leq N \leq 100$
- $0 \leq i < N$ for each i such that $1 \leq P[i] \leq 10^{15}$.
- $0 \leq i < N-1$ for each i such that $P[i] > P[i+1]$.

Subtasks

Subtask	Score	Additional Constraints
1	4	$N = 2$
2	3	$P[i] = N - i$ for each i such that $0 \leq i < N$.
3	14	$P[i] \leq P[i+1] + 2$ for each i such that $0 \leq i < N-1$.
4	18	$N = 3$
5	28	$P[i+1] + P[i+2] \leq P[i]$ for each i such that $0 \leq i < N-2$. $P[i] \leq 2 \cdot P[i+1]$ for each i such that $0 \leq i < N-1$.
6	33	No additional constraints.

Example

Consider the following call.

```
buy_souvenirs(3, 4)
```

هناك $N = 3$ أنواع من القطع التذكارية و $P[0] = 4$. لاحظ أنه لا يوجد سوى ثلاثة متتاليات ممكنة للأسعار P: $[4, 3, 2]$, $[4, 3, 1]$, و $[4, 2, 1]$.

افترض أن `buy_souvenirs` يستدعي `transaction(2)`. لنفترض أن المكالمة تقوم بإرجاع $([2], 1)$, وهذا يعني أن عمار اشترى قطعة تذكارية واحدة من نوع 2 وأعاد له البائع 1 دينار. لاحظ أن هذا يسمح لنا باستنتاج أن $P = [4, 3, 1]$ ، بما أن:

- بالنسبة إلى $P = [4, 3, 2]$ ، `transaction(2)` كان من الممكن أن يقوم بإرجاع $([2], 0)$.
- بالنسبة إلى $P = [4, 2, 1]$ ، `transaction(2)` كان من الممكن أن يقوم بإرجاع $([1], 0)$.

ثم `buy_souvenirs` يمكن الاتصال بـ `transaction(3)`, والتي تقوم بإرجاع $([1], 0)$, وهذا يعني أن عمار اشترى قطعة تذكارية واحدة من نوع 1 وأعاد له البائع 0 دينار. حتى الآن، في المجموع اشترى قطعة تذكارية واحدة من النوع 1 قطعة تذكارية واحدة من النوع 2.

وأخيرًا، `buy_souvenirs` يمكنه الاتصال بـ `transaction(1)`, والتي تقوم بإرجاع $([2], 0)$, مما يعني أن أمارو اشترى قطعة تذكارية واحدة من النوع 2. لاحظ أنه كان بإمكاننا أيضًا استخدام `transaction(2)` هنا. في هذه المرحلة، في المجموع لدى عمار قطعة تذكارية واحدة من النوع 1 وقطعتين من النوع 2 ، كما هو مطلوب.

Sample Grader

Input format:

```
N
P[0]  P[1] ... P[N-1]
```

Output format:

```
Q[0]  Q[1] ... Q[N-1]
```

هنا $Q[i]$ هو عدد القطع التذكارية من النوع i التي تم شراؤها إجمالاً لكل i بحيث $0 \le i < N$.