

خرد خرد

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این سوال به شما یک عدد k داده میشود که تعداد سکه‌هاست. باید ارزش عدد N را با استفاده از کمترین تعداد ارزش جمع‌های k سکه‌ی مختلف بسازید.

نکات مهم:

- تعدادی که از هر ارزش سکه داریم، نامحدود است.
- باید سوال به صورت تابع بازگشتی حل شود. نمره‌ی راه حل‌های با روش‌های دیگر 0 تلقی می‌شود.

ورودی گرفتن یک آرایه‌ی int به صورت تک خط:

```
1 | int[] input = Array.ConvertAll(Console.ReadLine().Split(), int.Parse);
```

ورودی

در خط اول عدد k به شما ورودی داده می‌شود و در خط بعدی، k ارزش‌های مختلف سکه‌هایی که داریم می‌آیند. (که ارزش تکراری وجود ندارد) در خط آخر ورودی عدد N آمده است.

$$1 \leq k, N \leq 100$$

خروجی

تنها خروجی لازم، کمترین تعداد سکه‌ی لازم برای ساخت N است.

مثال

ورودی نمونه 1

3
1 3 4
10

خروجی نمونه 1

3

باید 10 با کمترین تعداد جمع اعداد 1 3 4 ساخته شوند که در این مثال با $4+3+3$ ساخته می‌شود.

ورودی نمونه 2

3
1 7 8
6

خروجی نمونه 2

6

$$6=1+1+1+1+1+1$$

ورودی نمونه 3

4
1 5 4 9
35

خروجی نمونه 3

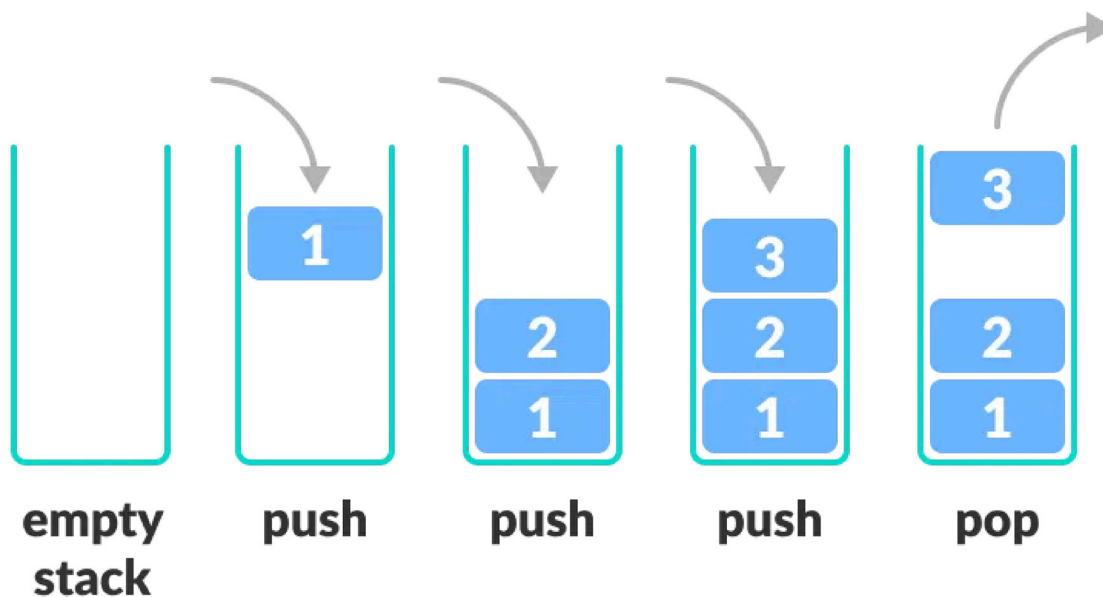
5

$$35=9+9+9+4+4$$

پیاده‌سازی استک

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این سوال شما باید یک استک (پشته) طراحی کنید که با گرفتن دستورات، اعمال push و pop را انجام دهد.



استک یا پشته از روند LIFO استفاده میکند (Last In First Out)

یعنی ورودی‌هایی که به استک داده می‌شوند مانند تصویر روی هم انبار می‌شوند و برای خارج کردن، از عمل pop استفاده می‌شود که آخرین عضو push شده را خروجی می‌دهد.

- دستور 1: عددی که بعد از دستور 1 می‌آید باید به استک push شود.
- دستور 2: با آمدن دستور 2، باید از استک pop کرده و آن را خروجی دهید. (از شما خواسته نمی‌شود که از استک خالی، چیزی را pop کنید).
- دستور 0: با وارد شدن دستور 0، ورودی‌ها تمام می‌شوند.

ورودی

تعداد کل خطوط ورودی از 1000 کمتر است.

در هر خط تنها یک عدد به شما داده می‌شود

خروجی

در خروجی باید به تعداد دستورات 2، اعداد pop شده را خروجی دهید.

مثال

ورودی نمونه ۱

1
2
1
3
1
1
2
2
0

خروجی نمونه ۱

1
3

به ترتیب اعداد 2، 3، 1 به استک اضافه میشوند و 2 تا آخر pop می‌شوند.

ورودی نمونه ۲

1
1
1
1
2
1
2
2
1
0
2
0

خروجی نمونه ۲

1
2
0

اول اعداد 1، 1 به استک اضافه میشوند، بعد آخرین 1 pop میشود، بعد 2 به استک اضافه شده و بلافاصله خروجی داده میشود، بعد 0 اضافه شده و دوباره بلافاصله خروجی داده میشود.

لیست اختلافات

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در این سوال ابتدا به شما n عدد داده می‌شود. باید با پیاده‌سازی یک تابع سورت، آن‌ها را مرتب کنید. **هم به**

صورت صعودی و هم به صورت نزولی

بعد باید با مقایسه‌ی اعضای نظیر به نظیر هر آرایه‌ی سورت شده با آرایه اصلی، کمترین اندازه آرایه تغییراتی که لازم است را خروجی دهید. برای فهم بهتر به مثال‌ها رجوع کنید.

نکته مهم: حتما خودتان سورت را پیاده‌سازی کنید. با استفاده از سورت آماده، نمره شما از 20% محاسبه خواهد شد.

```
1 | int[] arr = {4,5,2};
2 | Array.Sort(arr); //inplace sort function
```

همچنین در صورت برابری اندازه‌ی تغییرات، لیست صعودی را خروجی دهید.

مثال

ورودی

در خط اول عدد n به شما داده می‌شود و در خط بعدی، n اعداد آرایه به شما داده می‌شوند.

$$2 \leq n \leq 4000$$

اعضای آرایه حداکثر 4 رقمی‌اند.

خروجی

در تنها خط خروجی، n عدد قرار دارند که اختلاف هر عدد، به نظیر نزدیک‌ترین فرم مرتب‌شده‌ی خودش قرار دارد.

مثال

ورودی نمونه ۱

5
5 4 0 1 2

خروجی نمونه ۱

0 0 2 0 -2

در این نمونه، اعداد مرتب‌شده می‌شوند: 0,1,2,4,5 و 5,4,2,1,0.

حالا باید برای هر لیست مرتب، لیست اختلافات را حساب کنیم:

• ترتیب نزولی

5, 4, 0, 1, 2

5, 4, 2, 1, 0

لیست اختلاف برای مرتب نزولی می‌شود:

0, 0, 2, 0, -2

• ترتیب صعودی

5, 4, 0, 1, 2

0, 1, 2, 4, 5

لیست اختلاف برای مرتب صعودی می‌شود:

-5, -3, 2, 3, 3

حال برای انتخاب لیست اختلاف بهتر، لیستی که مجموع اندازه‌ی اعضایش کمتر است انتخاب می‌شود. که چون 4 > 16 پس لیست اختلاف مرتب نزولی خروجی داده می‌شود.

ورودی نمونه ۲

3

4 5 3

خروجی نمونه ۲

1 -1 0

مرتب صعودی: 3 4 5

مرتب نزولی: 5 4 3

• لیست اختلافات صعودی:

-1, -1, 2

• لیست اختلافات نزولی:

1, -1, 0

در این مثال هم جمع اندازه لیست اختلافات نزولی کمتر بود ($4 > 2$) پس لیست اختلافات نزولی خروجی داده میشود.