

mission impossible

فانتا همانطور که در گوشه ای از محوطه ی مهدکودک برای خودش بازی می کرد و از دور به جمع بقیه ی بچه ها نگاه می کرد تصمیم گرفت برای اینکه به بقیه ی بچه ها درسی بدهد، کم کم پایه های یک نقشه ی بی عیب و نقص و خبیثانه را طرح ریزی کند. برای پلان اول نقشه، فانتا باید قد بچه هایی که در مهدکودک هستند را از دفتر مربی کش برود، وارد برنامه کند و برنامه اش آنها را به ترتیب مرتب کند(چون دانش ریاضی خودش هنوز به این جاها نرسیده!)

فانتا قبلا کلاسی برای یک لینک لیست نوشته بود ولی الان نیاز داشت اعضای این لینک لیست مرتب نبود و به دردش نمی خورد، پس تصمیم گرفت کلاس جدیدی را بنویسد که از کلاس قبلی ارث میبرد ولی خاصیتش این است که باید مرتب باشد. اما فانتا تا ابد وقت برای نقشه ی خبیثانه اش، و همچنین دسترسی نامحدود به ابزار کدزنی اش نداشت. دو راه پیش پای فانتا قرار داشت:

راه اول این بود که متد pushFront را بازنویسی کنند به گونه ای که هنگام اضافه شدن مقدار جدید، این مقدار در جای اصلی خود اضافه شود به طوری که مرتب بودن لیست حفظ شود. مثلا در صورتی که لیست قبلا به صورت 1 5 8 بوده و اکنون عدد 3 اضافه میشود، مکان درست برای اضافه شدن عدد 3 مابین 1 و 5 است.

راه دوم این بود که متدهای pushFront و popFront را تغییر ندهد و مانند قبل کار کند ولی متد print را بازنویسی کنند به طوری که ابتدا لیست را sort کرده و سپس لیست مرتب شده را چاپ کند.

در نتیجه باید دو کلاس بنویسد که هرکدام از یکی از این روش ها استفاده کنند.(چرا هر دو روش؟ برای محکم کاری) کلاس SortedList1 از روش اول استفاده کرده و نتیجه اش به صورت صعودی و کلاس SortedList2 از روش دوم و به صورت نزولی است.

```
1 class Node
2 {
3     public:
4     int data;
5     Node * next;
6     Node(int data, Node* next);
7     Node();
8 };
9 class LinkedList
```

```

10 {
11     protected:
12         Node* head;
13         Node* tail;
14         int size;
15     public:
16         LinkrdList();
17         void pushFront(int data);
18         bool popFront();
19         void print();
20 };
21 class SortedList1 : public LinkedList
22 {
23     public:
24         void pushFront(int data);
25 };
26 class SortedList2 : public LinkedList
27 {
28     public:
29         void print();
30 };

```

ورودی

در ابتدا اعضای لیست اول وارد میشوند، سپس عدد 0 و بعد از آن اعضای لیست دوم و در انتها عدد 0 وارد میشود.

خروجی

خروجی برنامه باید شامل پرینت شدن لیست اول و لیست دوم به صورت مرتب شده باشد. لیست اول باید به صورت صعودی و لیست دوم به صورت نزولی مرتب شوند.

ورودی نمونه 1

```

1
10
4
-1
0

```

1
9
1
0

خروجی نمونه 1

List 1 : -1 1 4 10

List 2 : 9 1 1

توجه: در صورتی که توابع خواسته شده در صورت سوال را پیاده سازی نکنید فانتای فسقلی در نقشه شکست خورده و نمره ای به شما تعلق نمیگیرد.