توضیحات روش حل سوالات تمرین کامپیوتری دوم

سوال اول)
تابعی می‌نویسیم که برای هر نفر دو نوع خوشحالی را محاسبه کند:
a- ماکزیمم خوشحالی در صورتی که خود فرد حضور داشته باشد : حالت خوشحالی فرد + مجموع خوشحالی نوع b برای تمامی زیر دست‌هایش
b- ماکزیمم خوشحالی در صورتی که خود فرد حضور نداشته باشد : مجموع ماکزیمم خوشحالی افراد زیر دستش(ماکزیمم خوشحالی هر فرد زیردست = ماکزیمم بین خوشحالی نوع a و b او)
برای پیاده‌سازی هم تابعی می‌نویسیم که در خروجی‌اش هر دوی این مقادیر را برگرداند چون اگر دو تابع جدا بزنیم، بدلیل اینکه برای بدست آوردن حالت b هم از حالت a استفاده می‌شود و هم b این امکان بوجود می‌آید که یک تابع با ورودی‌های یکسان چندین‌بار صدا شود.

سوال دوم)
برای هر حرف در رشته نامرئی اگر طول کلمه متناظرش را k در نظر بگیریم(که این عدد k می‌تواند از یک شروع شود و به ترتیب اضافه شود) کلمه‌ای به طول k از ابتدای متن برداشته می‌شود و با حرف داخل رشته نامرئی متناظر می‌شود سپس سعی می‌کنیم ادامه باقی مانده متن را به همین روش با ادامه رشته تناظر بدهیم و در صورتی که توانستیم به طور کامل متن را با حروف نامرئی تناظر دهیم یعنی رشته نامرئی با متن هم‌خوانی دارد .
* چون ممکن است یک حرف چند بار در رشته نامرئی بیاید هر حرف در رشته نامرئی و کلمه متناظرش را نگه‌می‌داریم تا در صورت تکرار حرف به همان کلمه قبلی تناظر بدهیم.

سوال سوم)
ابتدا یک لیست مرتب از ۱ تا n را به وسیله ی رشته‌ی debug مرتب (با استفاده از merge sort و تابع debug داده شده) می‌کنیم و در نهایت می‌دانیم لیستی داریم شامل اعداد ۱ تا n به ترتیب مرتب است پس می‌توانیم بفهمیم هر عدد در ابتدا کجای لیست بوده است.
برای مثال فرض کنید ورودی ما به صورت زیر باشد:

```
4
LRRLR
```

ما در ابتدا لیست [۱,۲,۳,۴] را با استفاده از رشته‌ی debug مرتب می‌کنیم و به لیست [۴,۱,۳,۲] می‌رسیم (توجه کنید که این دنباله سورت شده است) پس به این نتیجه می‌رسیم که عددی که در ابتدا در جایگاه ۴ بوده است مقدار ۱ داشته، عددی که در جایگاه ۱ بوده است مقدار ۲، عددی که در جایگاه ۳ بوده عدد ۳ و عددی که در جایگاه ۲ بوده مقدار ۴. پس لیست اولیه برابر [۲,۴,۳,۱]

سوال چهارم)
به ترتیب هر بار از یکی از نقاط شروع می‌کنیم و با توجه به جهت داده شده در دنباله یک حرکت به آن سمت و به طول یک و یا دو برمی‌داریم و چک می‌کنیم نقطه ای که به آن رسیدیم نقطه validای باشد(داخل صفحه باشد و قبلا visit نشده باشد) و در صورتی که valid بود سعی می‌کنیم بقیه دنباله جهت‌ها را روی صفحه به همین شکل انجام دهیم و در نهایت به دو حالت می‌رسیم:

- تمامی دنباله جهت‌ها انجام شده باشند و جهت دیگری برای اعمال نمانده باشد -> یک جواب درست یافته ایم.

- زمانی که هنوز در دنباله جهت‌ها، جهت ای برای اعمال باقی ماند است ولی نمی‌توانیم با رعایت شروط گفته شده در بالا حرکتی انجام دهیم -> مجموعه حرکاتی که تا به الان انجام دادیم ما را به جوابی نمی‌رساند پس آخرین حرکت انجام شده را undoمی کنیم و سعی می کنیم راه حل دیگری پیدا کنیم.

* دقت کنید تنها در صورتی که حرکتی به طول ۲ انجام دهید ممکن است خطی که قبلا کشیده اید را قطع کنید.

* در map ای وضعیت هر نقطه را نگه می‌داریم(visit شده یا خیر) و در انجام یک حرکت اگر به طول یک باشد نقطه‌ای که در آن قرار گرفته ایم را visit می کنیم و در صورتی که طول حرکت دو باشد هم نقطه‌ای که در آن قرار گرفته ایم و هم نقطه میانی‌ای که در طول مسیر دیده ایم را visitمی‌کنیم.