

در این پروژه، یک سامانه کوچک برای یک بانک را شبیه‌سازی خواهیم کرد. این سامانه، رکوردهایی از تمامی مشتریان، حاوی نام، نام خانوادگی، شماره حساب و میزان موجودی در بانک را ذخیره می‌کند. این بانک ظرفیت سرویس‌دهی به حداکثر ۱۰۵ مشتری را دارد و لذا، شماره حساب‌ها پنج رقمی بوده و از ۰۰۰۰۰ آغاز و به ۹۹۹۹۹ ختم می‌شود. عملیاتی که بر روی سامانه تعریف شده‌اند، به قرار زیر است:

۱. جستجوی یک مشتری با شماره حساب مشخص: در صورت یافتن شماره حساب مورد نظر، نام، نام خانوادگی و میزان موجودی دارنده حساب را چاپ می‌کند. در غیر این صورت، پیام شکست عملیات را چاپ می‌کند. (بدترین حالت: $O(\log n)$)

```
INPUT:
SEARCH <account_no>

OUTPUT:
<first_name> <last_name> <assets>
```

۲. جستجوی مشتریان با سرمایه منفی: طی تراکنش‌های مشتریان، ممکن است برداشتی بیش از موجودی خود در بانک داشته باشند و موجودی آن‌ها منفی شود. این عملیات، اطلاعات تمامی مشتریان با موجودی منفی را برمی‌گرداند. در صورتیکه هیچ مشتری با موجودی منفی در بانک وجود نداشته باشد، نتیجه این عملیات پیام موفقیت خواهد بود. (بدترین حالت: $O(n)$)

```
INPUT:
SEARCH MINUS

OUTPUT:
<first_name> <last_name> <account_no> <assets>
```

۳. جستجوی پولدارترین مشتری: طی این عملیات، اطلاعات مشتری با بیشترین موجودی در بانک چاپ می‌شود. (بدترین حالت: $O(1)$)

```
INPUT:
SEARCH RICHEST

OUTPUT:
<first_name> <last_name> <account_no> <assets>
```

۴. جستجوی تمامی مشتریان با موجودی بیشتر یا مساوی یک مبلغ: طی این عملیات، اطلاعات تمامی مشتریانی که موجودی آن‌ها بیشتر یا مساوی مبلغ ذکر شده است، چاپ می‌شود. در صورتیکه هیچ مشتری با این مشخصات یافت نشود، نتیجه این عملیات پیام شکست خواهد بود. (بدترین حالت: $O(n)$)

```
INPUT:
SEARCH GT <value>

OUTPUT:
<first_name> <last_name> <account_no> <assets>
```

۵. افزودن یک مشتری جدید: طی این عملیات، یک مشتری با نام، نام خانوادگی و موجودی اولیه داده شده به مجموعه مشتریان بانک افزوده می‌شود و کوچکترین شماره حساب ممکن به او تخصیص داده می‌شود. توجه کنید که نام و نام خانوادگی مشتریان یکتا است و افزودن یک مشتری با نام و نام خانوادگی‌ای که قبلاً در سامانه ثبت شده است، مجاز نیست و باید پیام شکست عملیات چاپ شود. (بدترین حالت: $O(\log n)$)

```
INPUT:
REGISTER <first_name> <last_name> <assets>

OUTPUT:
SUCCESS
```

۶. حذف یک مشتری با شماره حساب مشخص: طی این عملیات، اطلاعات دارنده حساب به شماره ذکر شده حذف می‌شود. اگر مشتری با شماره حساب ذکر شده در سامانه موجود نباشد، پیام شکست و در صورت حذف موفقیت‌آمیز، پیام موفقیت چاپ می‌شود. (بدترین حالت: $O(\log n)$)

```
INPUT:
DELETE <account_no>
```

```
OUTPUT:
SUCCESS
```

۷. حذف مشتریان با موجودی منفی: این دستور، تمامی مشتریان با موجودی منفی را از سامانه حذف می‌کند و پس از اتمام عملیات، تعداد مشتریان حذف شده را چاپ می‌کند. (بدترین حالت: $O(n)$)

```
INPUT:
DELETE MINUS
```

```
OUTPUT:
<n_deleted>
```

۸. تغییر میزان موجودی مشتری به‌واسطه تراکنش‌های صورت گرفته توسط او: هر مشتری می‌تواند تراکنش‌هایی انجام دهد که به تغییر موجودی حساب او منجر شود. هر یک از این تراکنش‌ها به شکل زیر به سامانه اعلام می‌شوند. پس از انجام این تغییرات، اطلاعات کامل دارنده حساب به‌همراه موجودی پیشین و فعلی او چاپ می‌شود. (بدترین حالت: $O(\log n)$)

```
INPUT:
CHANGE <account_no> +/-<value>
```

```
OUTPUT:
<first_name> <last_name> <assets> <prev_assets> <new_assets>
```

۹. (امتیازی): قرعه‌کشی و افزایش موجودی برنده به میزان جایزه تعیین شده: قرعه‌کشی بدین صورت است که به هر فرد شانس متناسب با میزان موجودی‌اش تعلق می‌گیرد؛ بدین شکل که فردی که موجودی کمتری دارد، شانس بیشتری برای برنده شدن خواهد داشت. دقت کنید که افرادی که موجودی منفی دارند در قرعه‌کشی شرکت داده نمی‌شوند. میزان جایزه قرعه‌کشی نیز پیش از انجام آن به‌گونه‌ای تعیین می‌شود که با افزایش موجودی فرد برنده، سرمایه او بیشتر از اولین فردی که موجودی بیشتری از او دارد، نشود. اگر این میزان کمتر از میانگین موجودی دهک پایین مشتریان بانک باشد، قرعه‌کشی برگزار نمی‌شود و پیام شکست چاپ می‌شود. در غیر این صورت، اطلاعات فرد برنده (پیش از افزایش موجودی) به‌همراه میزان جایزه‌ای که به او تعلق گرفته است، چاپ می‌شود. (بدترین حالت: $O(\log n)$)

```
INPUT:
REWARD
```

```
OUTPUT:
<first_name> <last_name> <account_no> <prev_assets> +<value>
```

اگر عملیاتی به‌جز افزودن مشتری بر روی سامانه در زمانی که هنوز مشتری‌ای در سامانه ثبت نشده است، صورت گرفت، پیام خالی بودن پایگاه داده را چاپ کنید.

همچنین، لازم است اطلاعات بر روی فایل ذخیره شود (به‌کمک دستور SAVE و نیز در زمان خروج از برنامه) و در ابتدای اجرا نیز امکان بازیابی آن‌ها وجود داشته باشد (به‌کمک دستور LOAD).

۱. پایگاه داده مورد نیاز برای شبیه‌سازی سامانه بانک ذکر شده را به کمک یکی از ساختارهای داده‌ای درخت قرمز و سیاه یا درخت بی پیاده‌سازی کنید.

۲. به‌کمک عملگرهای تعریف شده در ساختار داده انتخابی در قسمت قبل و نیز، ساختار داده‌های مناسب دیگر عملیاتی که در بالا توصیف شده‌اند را پیاده‌سازی کنید. توجه کنید که کارایی الگوریتم‌ها از نظر حافظه و زمان اجرا مهم است و نمره متناسب با کارایی و ساختارهای انتخابی خواهد بود.

۳. (امتیازی) یک رابط گرافیکی ساده برای برنامه طراحی و پیاده‌سازی کنید.