



کانتکت‌دان

بعد از چند ترم تحصیل در رشته‌ی مهندسی کامپیوتر، شما و دوستان خود تصمیم می‌گیرید که یک سیستم عامل اختصاصی برای تلفن‌های همراه بنویسید. در مرحله‌ی تقسیم کار، تیم تصمیم می‌گیرد وظیفه‌ی نوشتن یک اپلیکیشن مدیریت مخاطبین را به شما محول کند. شما به عنوان یک عضو وظیفه‌شناس این تیم مسئولیت تولید این نرم افزار را بر عهده دارید.

مخاطبین

برنامه از تعدادی مخاطب¹ تشکیل شده است که هر یک اطلاعات یک نفر را در بر دارد. برنامه‌ی شما باید بتواند با استفاده از دستورات خاصی اطلاعات این افراد را ویرایش یا حذف کند. همچنین باید امکان جستجو روی اطلاعات این افراد را فراهم آوریم. در این برنامه برای هر مخاطب، چهار ویژگی نام، نام خانوادگی، آدرس ایمیل و شماره‌ی تلفن نگه‌داری می‌شود. همچنین به ازای هر مخاطب یک شناسه² یکتا نگه‌داری می‌شود که در بعضی از دسترسی‌ها از این شناسه استفاده می‌شود. دقت کنید که ترکیب نام و نام خانوادگی منحصر به فرد است یعنی ما نمی‌توانیم در برنامه خود دو مخاطب با نام و نام خانوادگی یکسان داشته باشیم. همچنین ایمیل و یا شماره‌ی تلفن نیز ویژگی منحصر به فردی است؛ یعنی نمی‌توانیم دو مخاطب با شماره‌ی تلفن یکسان یا ایمیل یکسان داشته باشیم.

دستورات

برنامه‌ی شما باید دستوراتی را که در ادامه آمده‌اند از طریق واسط خط فرمان³ دریافت و آن‌ها را پردازش کند تا بتوان مخاطبین موجود در برنامه را مدیریت کرد. پس از اجرا شدن برنامه‌ی شما، کاربر هر دستور را در یک خط و با فرمتی که در ادامه توضیح داده می‌شود به برنامه‌ی شما می‌دهد. برنامه پس از دریافت هر دستور با توجه به موفقیت‌آمیز بودن یا نبودن آن باید خروجی متناسب با آن دستور را چاپ کند. بسته به نوع دستور، از چهار گزینه⁴ زیر برای معرفی ویژگی‌های مخاطب استفاده می‌کنیم:

از گزینه‌ی f - برای معرفی نام مخاطب استفاده می‌کنیم؛ به طوری که کلمه‌ی بعد از این گزینه اسم مخاطب خواهد بود. همین طور l - برای مشخص کردن نام خانوادگی مخاطب، p - برای شماره‌ی تلفن و e - برای آدرس ایمیل مخاطب به کار می‌روند.

اضافه کردن مخاطب

برای استفاده از این دستور ابتدا باید از کلمه‌ی `add` استفاده شود و در ادامه‌ی این دستور، با استفاده از چهار گزینه‌ی معرفی‌شده، باید چهار ویژگی نام، نام خانوادگی، شماره‌ی تلفن و آدرس ایمیل مخاطب مشخص شود. (ترتیب این چهار گزینه اهمیتی ندارد ولی برای این دستور باید تمام آن‌ها باید موجود باشند). دقت داشته باشید که تولید شناسه برای هر مخاطب از وظایف برنامه‌ی شما است و این فیلد از ورودی گرفته نمی‌شود.

¹ Contact

² Id

³ Command Line Interface

⁴ Option

توجه کنید که در برنامه‌ی شما، شماره‌ی تلفن و آدرس ایمیل و همین‌طور ترکیب نام و نام خانوادگی باید یکتا باشد؛ یعنی نمی‌توانیم در لیست مخاطبین دو مخاطب با نام و نام خانوادگی، ایمیل یا شماره‌ی تلفن یکسان داشته باشیم. اگر اطلاعات ورودی برای دستور اضافه‌کردن این شروط را نقض کند، اضافه‌کردن این مخاطب ممکن نیست و باید پیغامی برای نشان‌دادن عدم موفقیت در اضافه‌کردن فرد به لیست مخاطبین به کاربر نمایش داده شود.

اگر این عمل موفقیت‌آمیز بود و مخاطب به لیست مخاطبین اضافه شد، باید پیام Command Ok و در غیر این صورت، پیام Command Failed را به کاربر نمایش دهید.

فرمت دستور

```
add <-f first_name> <-l last_name> <-p phone_number> <-e email_address>
```

فرمت خروجی

```
<Command Ok | Command Failed>
```

ورودی و خروجی نمونه

ورودی	خروجی
add -l Hajiloo -e m.taghi@gmail.com -f Mohammad -p 09191991919	Command Ok
add -l Hajiloo -e m.taghi@yahoo.com -f Mohammad -p 09208888888	Command Failed
add -l Hajiloo -e r@gmail.com -f Reza -p 09191991919	Command Failed

جستجوی مخاطب

با این دستور امکان جستجو میان مخاطبین را خواهیم داشت. برنامه‌ی شما باید در میان هر چهار ویژگی مخاطبین، کلیدواژه‌⁵ داده‌شده را جستجو کند و در خروجی لیست مخاطبینی را که این کلیدواژه زیررشته‌ی یکی از چهار ویژگی آن‌ها است چاپ کند. برای مثال، Re با za زیررشته‌ی Reza هستند و فردی با این نام باید در نتیجه‌ی جستجوی این کلیدواژه‌ها ظاهر شود. اگر در نتیجه‌ی جستجو مخاطبی وجود نداشت، نباید چیزی در خروجی چاپ شود.

فرمت دستور

```
search <keyword>
```

فرمت هر خط خروجی

```
<id> <first_name> <last_name> <email_address> <phone_number>
```

⁵ Keyword

ورودی و خروجی نمونه

ورودی	خروجی
search gmail	0 Mohammad Hajiloo m.taghi@gmail.com 09191991919

حذف مخاطب

برنامه‌ی شما باید بتواند با گرفتن شناسه‌ی یک مخاطب آن را از برنامه حذف کند. توجه کنید که برای حذف یک مخاطب که ویژگی‌های آن را – مثل ایمیل یا نام خانوادگی – می‌دانیم، ابتدا باید با استفاده از دستور جستجو، شناسه‌ی مخاطب مربوط را پیدا و سپس آن مخاطب را حذف کنیم. همانند دستور add، اگر این عمل موفقیت‌آمیز بود و مخاطب مشخص شده حذف شد، باید پیام Command Ok و در غیر این صورت، باید پیام Command Failed چاپ شود.

فرمت دستور

delete <id>

فرمت خروجی

<Command Ok | Command Failed>

ورودی و خروجی نمونه

ورودی	خروجی
delete 3 delete 0	Command Failed Command Ok

ویرایش مخاطب

در این برنامه باید بتوانیم با استفاده از دستور update اطلاعات مخاطبین موجود در برنامه را ویرایش کنیم. برای استفاده از این دستور، پس از کلمه‌ی update در ابتدای دستور، ابتدا شناسه‌ی مخاطبی که قصد ویرایش اطلاعات آن را داریم وارد می‌کنیم و سپس با استفاده از گزینه‌های 1-، f-، p- و e- مقدار جدید ویژگی‌هایی را که می‌خواهیم آن‌ها را ویرایش کنیم وارد می‌کنیم. دقت کنید که می‌توانیم با هر ترتیبی گزینه‌ها را وارد کنیم، اما برخلاف دستور add، می‌توانیم به تعداد دلخواه از آن‌ها استفاده کنیم. پس از اجرا شدن این دستور، باید شرط منحصربه‌فرد بودن ویژگی‌های مخاطبین – همان‌گونه که در قسمت‌های قبل بیان شده است – با توجه به ویژگی‌های جدید برقرار باشد تا اجرای دستور موفقیت‌آمیز باشد. همانند دستور add، اگر این عمل موفقیت‌آمیز بود و مخاطب مشخص شده حذف شد، باید پیام Command Ok و در غیر این صورت، باید پیام Command Failed چاپ شود.

فرمت دستور

```
update <id> [-f first_name] [-l last_name] [-p phone_number] [-e email_address]
```

فرمت خروجی

<Command Ok | Command Failed>

ورودی و خروجی نمونه

ورودی	خروجی
Update 0 -f Sadegh	Command Ok

نکات تکمیلی

(۱) هنگام اضافه کردن و ویرایش یک مخاطب لازم است تا معتبر^۶ بودن آدرس ایمیل و شماره‌ی تلفن آن مخاطب بررسی شود. شماره‌ی تلفن در صورتی معتبر است که:

- تماماً عدد باشد.
- با 09 شروع شود.
- ۱۱ رقمی باشد.

آدرس ایمیل در صورتی معتبر است که:

- یک رشته شامل کاراکترهای @ و (dot) باشد که @ قبل از dot آمده باشد.

در صورتی که در هر یک از دستورهای add و update، آدرس ایمیل یا شماره‌ی تلفن معتبر نباشند، باید عبارت Command Failed در خروجی چاپ شود.

(۲) پس از وارد شدن هر دستور و اجرای آن توسط برنامه‌ی شما، اگر به هر دلیلی اجرای دستور موفقیت‌آمیز نبود باید پیام Command Failed چاپ شود؛ مثلاً در صورتی که در دستور add تعداد گزینه‌های وارد شده کمتر از چهار بود.

(۳) برنامه‌ی شما نباید به وجود فاصله در میان کلمات هر دستور حساس باشد؛ به عنوان مثال، دو دستور زیر باید عملکرد یکسانی داشته باشند:

```
update 1 -f amir
update 1 -f amir
```

ذخیره‌سازی (بخش امتیازی)

برنامه‌ی شما باید این قابلیت را داشته باشد که اگر برنامه پایان یابد و مجدداً اجرا شود، اطلاعات مخاطبینی که قبل از پایان یافتن برنامه در آن موجود بود، بازیابی شود.

^۶ Valid

برای پیاده‌سازی این قسمت می‌توانید قبل از پایان یافتن برنامه، اطلاعات مخاطبان را با فرمت CSV⁷ در یک فایل با نام contacts.csv ذخیره کنید، و هر بار که برنامه آغاز می‌شود، اطلاعات را از این فایل خوانده و اطلاعات مخاطبین را به کمک آن پر کنید.

CSV یک فرمت ذخیره‌سازی داده⁸ است که برای ذخیره‌سازی داده‌هایی که به صورت یک جدول هستند استفاده می‌شود. اطلاعات بیشتر درباره‌ی فرمت CSV را می‌توانید در لینک زیر پیدا کنید:

https://en.wikipedia.org/wiki/Comma-separated_values

نکته‌ی پایانی

سعی کنید پروژه را با پیاده‌سازی مرحله‌به‌مرحله‌ی دستورات گفته‌شده و با ترتیبی که در متن پروژه آمده پیاده‌سازی کنید. همچنین، پس از پیاده‌سازی هر مرحله برنامه‌ی خود را تست کنید تا اگر قسمتی از برنامه‌ی شما به درستی کار نکرد، نمره‌ی بقیه بخش‌های آن را از دست ندهید.

نحوه‌ی تحویل

فایل برنامه‌ی خود را با نام A1-SID.cpp در صفحه‌ی CECM درس بارگذاری کنید؛ برای مثال، اگر شماره‌ی دانشجویی شما ۸۱۰۱۱۲۳۴۵ است، نام فایل شما باید A1-810112345.cpp باشد.

لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.

● برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

● به فرمت و نام فایل‌های خود دقت کنید.

● برنامه‌ی شما به صورت خودکار تست خواهد شد برای همین از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.

● هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

⁷ Comma-Separated Values

⁸ Data