



مقدمه

هدف از این تمرین آشنایی شما با ورودی و خروجی و کار با رشته‌ها و بردارها در زبان C++ به عنوان مهارت‌های پایه‌ای برنامه‌نویسی در این زبان است. این مهارت‌ها پیاده‌سازی پروژه‌های بزرگتر را امکان‌پذیر می‌کنند. در این تمرین شما یک فهرست مخاطبین ساده را پیاده‌سازی می‌کنید.

علاوه بر درستی برنامه، تمیزی کد و جدا کردن برنامه به توابع مختلف اهمیت زیادی دارند. سعی کنید قبل از انجام تمرین ویدیوهای بخش تمیزی کد موجود در صفحه‌ی درس را مشاهده کنید.

شرح مسئله

در این تمرین قرار است برنامه شما فهرستی از مخاطبین را نگهداری کند و اعمال اضافه، حذف و جستجو را روی آن ارائه کند. هر مخاطب نام، نام خانوادگی، شماره تلفن و آدرس‌ای میل دارد. ترکیب نام و نام خانوادگی مخاطبین یکتا است، یعنی نمی‌توانیم دو مخاطب با نام و نام خانوادگی یکسان داشته باشیم. به همین ترتیب، شماره تلفن و آدرس‌ای میل مخاطبین نیز یکتا است.

فرمان‌های برنامه

ورودی برنامه در قالب تعدادی سطر در ورودی استاندارد به برنامه داده می‌شود که هر سطر حاوی یکی از سه فرمان اضافه، جستجو و حذف است. قالب این فرمان‌ها در ادامه ذکر شده است. دقت نمایید مابین کلمه‌های یک خط فرمان می‌تواند تعداد دلخواهی فاصله خالی باشد.

اضافه کردن مخاطب

قالب ورودی این فرمان به این ترتیب است:

```
add -f firstName -l lastName -p phone -e email
```

چهار بخش مشخص شده بعد از کلمه add از گزینه‌های مشخص شده با -f و -l و -p و -e برای مشخص کردن نام، نام خانوادگی، شماره تلفن و آدرس‌ای میل استفاده می‌کند. فرض کنید هر دستور add حتماً این چهار جزء را دارد اما ترتیب آنها می‌تواند به شکل دلخواهی داده شود. همچنین می‌توانید فرض کنید هیچ‌یک از چهار جزء داده شده رشته تهی نیستند.

هنگام اضافه کردن شماره تلفن و آدرس‌ای میل باید قواعد زیر را کنترل کنید:

- شماره تلفن باید حتما ۱۱ رقم داشته باشید و حتما با 09 شروع شود.
- آدرس ای میل به شکل name@domain است که در آن
 - بخش name فقط از حروف و ارقام تشکیل شده و طول آن صفر نیست
 - بخش domain از حروف، ارقام و نقطه تشکیل شده، با این شرط که اول و آخر آن نقطه نیست و بین هر دو نقطه حداقل یک حرف یا رقم وجود دارد.

در صورتی که اضافه کردن مخاطب موفقیت آمیز باشد عبارت Command OK و در غیر این صورت (تکراری بودن ارقام یا نامعتبر بودن تلفن یا ای میل) عبارت Command Failed در یک سطر مجزا در خروجی نوشته می شود.

جستجوی مخاطب

قالب ورودی این فرمان به این ترتیب است:

search keyword

با این فرمان، کاربر می تواند در بین مخاطبینی که تا آن زمان اضافه شده اند جستجو کند. نتیجه این جستجو شامل مخاطبینی می شود که دست کم یکی از اجزای آنها (نام، نام خانوادگی، شماره تلفن، آدرس ای میل) شامل زیر رشته keyword می شود. خروجی این فرمان هر مخاطب را در یک سطر خروجی با قالب زیر می نویسد:

id firstName lastName phone email

که در آن id شناسه مخاطب است. این شناسه یک عدد یکتا است که هنگام اضافه کردن مخاطب توسط برنامه شما به آن مخاطب اختصاص می یابد. این شناسه یکی از بزرگترین شناسه موجود در فهرست مخاطبین در آن لحظه بیشتر است. اولین مخاطب شناسه صفر دریافت می کند. خروجی فرمان جستجو باید بر اساس شناسه به طور صعودی مرتب باشد. در صورتی که مخاطبی پیدا نشود که شامل کلید واژه جستجو باشد چیزی در خروجی نخواهیم داشت.

حذف مخاطب

قالب ورودی این فرمان به این ترتیب است:

delete id

با این فرمان کاربر درخواست حذف مخاطب با شناسه id را صادر می کند. در صورتی که مخاطبی با شناسه id موجود باشد آن مخاطب حذف می شود و عبارت Command OK در خروجی نوشته می شود. در غیر این صورت عبارت Command Failed در خروجی نوشته می شود.

ورودی و خروجی نمونه

ورودی	خروجی
add -f ramtin -l khosravi -p 09123456789 -e ramtung@gmail.com	Command OK
add -f ramtan -e ramtung@gmail.biz -p 09123456780 -l khosravi	Command OK
add -f ramton -l khosravi -p 09123456785 -e ramtung@gmail.com	Command Failed
search ram	0 ramtin khosravi 09123456789 ramtung@gmail.com

```
delete 1
search 123
delete 1
```

```
1 ramtan khosravi 09123456780 ramtung@gmail.biz
Command OK
0 ramtin khosravi 09123456789 ramtung@gmail.com
Command Failed
```

نحوه تحویل

- کد خود را در قالب یک فایل با نام A1-SID.cpp در صفحه‌ی eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۰۰۰۰ باشد، نام پرونده کد شما باید A1-810100000.cpp باشد که شامل کد شما است.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- تمیزی کد، شکستن مرحله‌به‌مرحله مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.
- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین از درستی کامل قالب خروجی برنامه خود اطمینان حاصل کنید و از دادن خروجی‌هایی که در صورت پروژه ذکر نشده‌است اجتناب کنید. برای مقایسه‌ی خروجی خود حتماً از توضیحات **پیوست ۱** استفاده کنید تا مشکلی در زمان تحویل در اجرای تست‌های شما پیش نیاید.
- در طول این تمرین ممکن است با مشکلاتی روبه‌رو شوید که راه حل آن‌ها را نمی‌دانید؛ در این صورت، جست‌وجوگرهایی مانند google و سایت‌هایی مانند **stackoverflow** و **cplusplus.com** ممکن است به شما کمک کنند.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.

پیوست ۱

مقایسه خروجی برنامه با خروجی مورد انتظار با چشم شاید برای برنامه‌های کوچک که خروجی کمی تولید می‌کنند و روند اجرای کوتاهی دارند میسر باشد، اما این کار برای برنامه‌های بزرگ‌تر با مسیر اجرای پیچیده کاری دشوار است. برای این کار می‌توان از ابزارهایی که در سیستم عامل لینوکس در دسترس است استفاده کرد. برای این کار با استفاده از مجموعه دستورات زیر خروجی را در فایل دلخواه ذخیره می‌کنیم.

```
g++ -std=c++11 source.cpp
./a.out < in.txt > out.txt
```

حال فرض کنیم که خروجی درست در فایل true_out.txt قرار داشته باشد، برای مقایسه‌ی این خروجی با خروجی out.txt که توسط برنامه‌ی ما تولید شد از ابزار **diff** استفاده می‌کنیم.

```
diff out.txt true_out.txt
```

اگر دو فایل یکسان باشند دستور diff هیچ خروجی ای تولید نمی کند. وگرنه، تفاوت های دو فایل را نمایش می دهد.
برای مطالعه ی بیشتر درباره ی diff می توانید از [این لینک](#) استفاده کنید.