



سوال ۶. کمی هم گوگل کنیم!

شما باید تابع `findDivisorSum` را به صورتی پیاده‌سازی کنید که تعدادی عدد از ورودی گرفته و مجموع شمارنده‌های آن‌ها را محاسبه کند. سپس به صورت زیر خروجی را برگرداند. (متغیر `s` خارج از برنامه شما به صورت گلوبال تعریف شده است و شما می‌توانید در هر جای برنامه از آن استفاده کنید):

- اگر مقدار `s` برابر ۰ بود، تابع شما ۴ ورودی گرفته که ورودی اول آن یک کاراکتر و ورودی‌های بعدی اعداد شما هستند. اگر مقدار کاراکتر `M` بود، بیشترین و اگر `m` بود کمترین مقدار را از بین مجموع عامل‌های اول ورودی‌ها برگردانید.
- اگر مقدار `s` برابر ۰ نبود، ورودی اول تابع `n` بوده و پس از آن `n` ورودی می‌آید. این بار نیز مجموع عامل‌های اول را محاسبه کرده و کمترین آن‌ها را برگردانید.

اعداد ورودی تابع شما به صورت `calc(num1, num2, +)` یا هر عملیات ریاضی که جایگزین `+` شده است، می‌باشند. عملیات ریاضی به صورت کاراکتر نبوده و شما باید کدی بنویسید که دقیقاً همین فرمت را پردازش کند. (در صورتی که عملیات ریاضی تقسیم باشد، عدد اول حتماً بر عدد دوم بخش‌پذیر است).

مثال

حالت اول (متغیر `s` با مقدار ۰ در برنامه وجود دارد):

```
findDivisorSum('M', calc(5, 10, +), calc(17, 13, *), calc(45, 5, /))
```

حالت دوم (متغیر `s` با مقدار ۱ در برنامه وجود دارد):

```
findDivisorSum(2, calc(12, 2, *), calc(50, 17, %))
```

شما تنها باید تابع‌های خواسته شده را پیاده‌سازی کنید و کد شما نباید تابع `main` داشته باشد. همچنین باید کد زیر در ابتدای برنامه‌ی شما قرار بگیرد.



```
1 #include "grader.h"
```

به علاوه تابع زیر باید در برنامه ی شما وجود داشته باشد:

```
1 long long run(char type, int num1, int num2, int num3, int num4) {  
2     return findDivisorSum(type,  
3     calc(num1, num2, +),  
4     calc(num1, num2, -),  
5     calc(num3, num4, *));  
6 }  
7
```

نکات ارسال پاسخ:

۱. پسوند فایل نهایی خود را به `.cpp` تغییر داده و به عنوان کد `C++` ارسال کنید. (کد شما باید به زبان `C` باشد ولی ارسال به عنوان `C++` اشکالی ندارد).
۲. برای قسمت اول و دوم سوال کد یکسان ارسال کنید. توجه کنید که در صورت مغایرت کد های ارسالی نمره ای از سوال دریافت نخواهید کرد.