

تمرین شماره ۶

درس اسمبلی و زبان ماشین

می خواهیم الگوریتمی بنویسیم که تمامی مقسوم علیه های اول یک عدد طبیعی را چاپ کند. شما باید این الگوریتم را به صورت یک تابع به زبان C و یک تابع به زبان اسمبلی پیاده سازی کنید.

فایل hw6.zip را باز کنید. این فایل شامل فایل های زیر است:

```
asm_io.asm
asm_io.inc
prime_factors_asm.asm
prime_factors.c
testcase
test_prime_factors.c
Makefile
```

فایل های `asm_io.asm` و `asm_io.inc` شامل توابع تعریف شده در کتاب هستند و صرفا می توانند برای خطایابی در برنامه اسمبلی به کار روند. استفاده از این توابع در برنامه اسمبلی لزومی ندارد. (بلکه در برنامه نهایی نباید از توابع کتاب استفاده شود)

فایل `testcase` شامل یکسری داده برای امتحان کردن برنامه است. تست کردن توابع شما در فایل `test_prime_factors.c` صورت می گیرد. این فایل را باز کنید و محتویات آن را نگاه کنید. در این فایل دو تابع `prime_factors` و `prime_factors_asm` فراخوانی شده اند که اولی به زبان C و دومی به زبان اسمبلی باید نوشته شوند. این توابع به ترتیب باید در فایل های `prime_factors.c` و `prime_factors_asm.asm` نوشته شوند. در حال حاضر این دو فایل خالی هستند.

فایل `Makefile` برای کامپایل کردن برنامه به کار می رود. با دستور `make` برنامه کامپایل می شود و در صورت موفق بودن عملیات کامپایل یک فایل اجرایی با نام `test_prime_factors` ساخته می شود. این فایل اجرایی را با دستور زیر می توانید در لینوکس اجرا کنید:

```
$ ./test_prime_factors
```

یا برای کامپایل کردن و اجرای همزمان می توانید در ترمینال لینوکس بنویسید

```
$ make && ./test_prime_factors
```

در لینک های زیر می توانید در مورد `Makefile` بخوانید.

<http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>

<http://www.tutorialspoint.com/makefile/>

همانطور که عنوان شد وظیفه شما نوشتن توابع `prime_factors` و `prime_factors_asm` می باشد. پروتوتایپ توابع به این صورت است:

```
int prime_factors(int n, int factors[]);  
int prime_factors_asm(int n, int factors[]);
```

(فایل `test_prime_factors.c` را نگاه کنید)

تابع `prime_factors` باید حتما به زبان C و در فایلی به نام `prime_factors.c` نوشته شود. (در حال حاضر این فایل خالی است)

تابع `prime_factors_asm` باید حتما به زبان اسمبلی و در فایلی به نام `prime_factors_asm.asm` نوشته شود. (در حال حاضر این فایل خالی است).

دقت کنید که نام فایلها و نام توابع باید دقیقا رعایت شوند.

هر دو این توابع مقسوم علیه های اول عدد n را به صورت مرتب شده از کوچک به بزرگ در آرایه `factors` می ریزند و به عنوان مقدار برگشتی تعداد مقسوم علیه های اول را چاپ می کنند. در حقیقت مقدار برگشتی مقداری از آرایه `factors` که استفاده شده را نشان می دهد. توجه کنید که تکرار مقسوم علیه های اول باید محاسبه شوند. برای مثال اگر عدد n برابر با ۱۲۰ باشد. محتوای آرایه `factors` باید به صورت زیر باشد:

2, 2, 2, 3, 5

و همچنین مقدار برگشتی تابع برابر با 5 است.

فایل `test_prime_factors.c` هر دو تابع فوق را صدا می زند و خروجی این توابع را بر روی تعدادی از اعداد موجود در فایل `testcase` تست می کند. همچنین تخمینی زمان اجرای هر کدام از توابع را چاپ می کند.

توجه کنید که بجز فایل های `prime_factors.c` و `prime_factors_asm.asm`، هیچک از فایلها را نباید تغییر دهید.