

به نام خدا

تمرین عملی درس ساختار و زبان کامپیوتر

استاد: دکتر سربازی آزاد

مهلت تحویل: ۲ بهمن ۱۳۹۸

هر کدام از برنامه‌های خواسته شده را می‌توانید با زبان اسمبلی 8086 و یا MIPS بنویسید. در صورتی که یک برنامه را با هر دو زبان اسمبلی پیاده‌سازی کنید حداکثر ۳۰ درصد نمره‌ی اضافی بابت آن برنامه دریافت خواهید کرد. تمرین باید در قالب گروه‌های دو نفره انجام گیرد. کدهای اسمبلی خود را در قالب یک فایل فشرده در صفحه درس حداکثر تا ۲ بهمن ساعت ۲۳:۵۹ بارگذاری کنید. روز پنجشنبه ۳ بهمن، تحویل حضوری تمرین در آزمایشگاه پژوهشی آقای دکتر سربازی آزاد (اتاق ۷۰۳) خواهد بود.

1- در این سوال از شما خواسته شده است تا یک سری عملیات سطری را بر روی یک ماتریس مربعی پیاده‌سازی کنید. در ابتدا عدد n را ورودی گرفته سپس حافظه‌ای به اندازه یک ماتریس $n \times n$ بگیرید. سپس مقدار درایه را از ورودی بخوانید. به عنوان مثال :

3
1 3 2 1 5 6 4 2 0

این ورودی مساوی ماتریس $\begin{pmatrix} 1 & 3 & 2 \\ 1 & 5 & 6 \\ 4 & 2 & 0 \end{pmatrix}$ است.

الف) تابعی بنویسید که آدرس شروع ماتریس و دو عدد r, s و c به عنوان ورودی گرفته و سپس c برابر سطر s ام را به سطر r ام اضافه کرده و آن را به جای سطر r ام قرار دهد و سپس ماتریس را به شکل دو بعدی نمایش دهد.

ب) تابعی بنویسید که آدرس شروع ماتریس و دو عدد r و c را گرفته و سپس تمام مقادیر سطر r ام را در عدد ثابت c ضرب کند، سپس ماتریس را به شکل دو بعدی نمایش دهد.

ج) با استفاده از دو تابع بالا و روش گاوس-جردن برنامه‌ای بنویسید که یک ماتریس مربعی را ورودی گرفته و در صورت امکان وارون آن را خروجی دهد. (اگر ماتریس وارون نداشت باید با چاپ پیامی بگوید که ماتریس وارون پذیر نیست).

2 - ابتدا عدد n و سپس رشته X به طول n به شما داده می‌شود. در خط بعد، کاراکتر r به معنی $replace$ یا کاراکتر c به معنی $count$ به عنوان ورودی داده می‌شود. اگر r داده شد ابتدا اعداد a و b و سپس دو رشته به طول a و b به شما داده می‌شود. اگر رشته اول در X موجود بود، آن را با رشته دوم جایگزین کنید و رشته جدید را در خروجی چاپ کنید. اگر c به شما داده شد، ابتدا عدد a و سپس یک رشته به طول a به شما داده می‌شود. تعداد تکرارهای این رشته را در X بیابید.

3 - فرض کنید اعداد 1، 2، ... و n را روی یک دایره در جهت عقربه‌های ساعت چیده‌ایم. از عدد 1 شروع و اعداد را یکی در میان حذف می‌کنیم تا سرانجام یک عدد باقی بماند. این عدد را $f(n)$ می‌نامیم. مثلاً $f(5) = 3$. زیرا اگر اعداد 1 تا 5 را در دایره قرار دهیم، اعداد 1، 2، 4، 5 به ترتیب حذف می‌شوند و عدد 3 باقی می‌ماند. حال برنامه‌ای به زبان اسمبلی بنویسید که عدد n را به عنوان ورودی دریافت کرده و عدد $f(n)$ را در خروجی نمایش دهد.

4 - لیست پیوندی ساختاری خطی است که در آن هر عنصر شامل داده‌ی خود و اشاره‌گری به عنصر بعد، قبل یا هر دو است. انجام عملیات درج و حذف در این ساختار بسیار ساده است.

برنامه‌ای به زبان اسمبلی بنویسید که لیست پیوندی یک‌طرفه‌ای از اعداد را پیاده‌سازی کند. این کد باید قابلیت‌های زیر را داشته باشد:

1. عدد a را به ابتدای لیست اضافه کند.
2. عدد a را به انتهای لیست اضافه کند.
3. عدد a را بعد از عدد b درج کند (اول a و بعد b داده می‌شود).
4. عدد a را از لیست حذف کند.
5. جایگاه عدد a در لیست را پیدا کند (اندیس‌ها از صفر شروع می‌شوند).
6. لیست پیوندی را از ابتدا تا انتها چاپ کند.

هر یک از عملیات بالا به ترتیب با کد 1 تا 6 و پس از آن داده‌ی مربوطه داده خواهند شد (البته به جز عملیات 6). فرض کنید در هر لحظه، اعضای لیست متمایز هستند و اعداد خواسته‌شده در لیست موجودند.

5 - برنامه‌ای به زبان اسمبلی بنویسید که یک ماشین حساب ساده را پیاده‌سازی کند. این ماشین حساب یک رشته از عملیات ریاضی شامل پرانتز می‌گیرد؛ در صورت صحت رشته، نتیجه و در غیر این صورت، پیغام خطای مناسب چاپ می‌کند.

عملیاتی که این ماشین حساب انجام می‌دهد عبارتند از جمع، تفریق، ضرب و تقسیم. خطاهایی که باید این ماشین حساب تشخیص دهد سرریز، تقسیم بر صفر و پرانتز گذاری اشتباه است.

6 - تعدادی نقطه در فضای دوبعدی با اعداد صفر و یک برچسب خورده‌اند. می‌خواهیم با عبور خطی در این فضا، نقاط با برچسب مختلف را از هم جدا کنیم؛ بدین صورت که در حالت ایده‌آل، همه‌ی نقاط بالای خط برچسب یک، و همه‌ی نقاط

پایین خط برچسب صفر خورده باشند. بهترین خط، خطی است که کمترین تعداد نقطه را اشتباه برچسب بزند (در مثال ذکر شده، یعنی کمترین تعداد صفر بالای خط و یک پایین خط داشته باشیم).

سه نوع خط برای جداسازی این نقاط در نظر می‌گیریم:

۱. خطی که تنها مختصات x نقاط را جدا کند، به عبارتی بر محور x عمود باشد.
۲. خطی که تنها مختصات y نقاط را جدا کند، به عبارتی بر محور y عمود باشد.
۳. خطی که نسبت مختصات y به x را جدا کند، به عبارتی از مبدأ مختصات رد شود.

برنامه‌ای به زبان اسمبلی بنویسید که n نقطه در فضای دو بعدی بگیرد، برای سه مدل ذکر شده بهترین خط را پیدا کند و در نهایت معادله‌ی هر خط به همراه خطای آن را گزارش کند. شیب و عرض از مبدأ خطوط را اعداد صحیح در نظر بگیرید.

7 - برنامه‌ای به زبان اسمبلی بنویسید که با کاربر هپ پیشرفته بازی کند! بازی از عددی که کاربر وارد کند شروع می‌شود و در هر مرحله اعداد سه تا سه تا زیاد می‌شوند. هر طرف در نوبت خود، یا عدد بعدی و یا در صورتی که عدد اول باشد، عبارت hop چاپ (از طرف کد شما) یا وارد (از طرف کاربر) می‌شود. دقت کنید که کد شما همیشه باید خروجی درست را چاپ کند، ولی کاربر ممکن است اشتباه کند (مثلاً عدد بعدی را ننویسد یا عدد بعدی اول باشد و hop وارد نکند). در این صورت کد شما باید این اشتباه را گزارش کرده بازی را متوقف کند.

نمونه ورودی:

```
4
10
16
22
29
```

خروجی مربوطه:

```
hop      // 7
hop      // 13
hop      // 19
25
You lose :P
```

8 - برنامه‌ای به زبان اسمبلی بنویسید که با گرفتن ورودی اعداد a و b و x در مبنای a ، عدد x را در مبنای b بدست آورد.

9 – برنامه‌ای به زبان اسمبلی بنویسید که مکان بزرگترین زیررشته‌ی آینه‌ای در رشته ورودی را پیدا کند. رشته‌ی آینه‌ای رشته‌ای است که از دو طرف به یک شکل دیده می‌شود: مثال: kalak

10 – در یک مختصات مشخصی در صفحه‌ی دوبعدی، گنج وجود دارد و کاربر می‌خواهد با انجام حرکاتی به سمت بالا، پایین، چپ یا راست به این گنج برسد. ولی نابرده رنج، گنج میسر نمی‌شود و در مسیر موانعی وجود دارد. برنامه‌ای بنویسید که ابتدا مختصات دوبعدی نقطه‌ی گنج به همراه تعداد موانع و مختصات آنها را بگیرد. پس از آن کاربر با شروع از مبدأ، با وارد کردن نویسه‌های u, d, l و r در صورت نبود مانع، به ترتیب یک خانه به بالا، پایین، چپ یا راست منتقل شود و مختصات کاربر چاپ شود. در صورت وجود مانع، پیغام can't move, obstacle ahead چاپ شود. در صورتی که کاربر به گنج برسد، پیغام reached treasure! چاپ شده و برنامه پایان یابد.

نمونه ورودی:

```
3 3      // treasure coordinates
2        // number of obstacles
1 1      // obstacle 1 coordinates
1 2      // obstacle 2 coordinates
u
u
r        // hit obstacle 2
u
r
r
r
```

خروجی مربوطه:

```
0 1
0 2
can't move, obstacle ahead
0 3
1 3
2 3
3 3
reached treasure!
```