احمدرضا یزدانی(۹۹۳۲۰۹۵)

قسمت اول:

من برای رسیدن به جواب در این مرحله ابتدا کد ورودی را به دو بخش بدست آوردن opcode و عدد معادل رجیستر ها تقسیم کردم:

- برای بدست آوردن opcode همه های حالت ها با توجه به ۴ دستور مورد نیاز و اندازه رجستر، ان ها را یکی یکی با شرط برسی کردم و بدست آوردم.

- برای بدست اوردن عدد معادل رجیستر ها، من با توجه به خاصیت های dictionary در پایتون عدد معادل رجستر ها را به توجه به source یا destination بودن در ۲ دیکشنری مختلف قرار دادم سپس ان ها را با mod که در این مرحله ثابت است جمع کردم و در انتها عدد بدست آمده را به شکل hexadecimal در آوردم.

- در مرحله اخر هم opcode و عدد معادل رجستر ها را به هم چسباندم که معادل کد اسمبل شده است.

قسمت دوم:

من در ادامه مرحله قبل و برای اضافه کردن توانایی کار با حافظه، همانند مرحله قبل با استفاده از خاصیت های dictionary آن ها را نیز به کدم اضافه کردم، که البته این کد از مموری های ۳۲ و ۸ بیت و فقط بعضی از ۱۶ بیتی ها پشتیبانی می کند.

قسمت سوم:

در این مرحله من برای پشتیبانی کردن کدم از پرش های تک بایتی کارهای زیر را انجام دادم:

- ابتدا قابلیت خواندن فایل را به کدم اضافه کردم که بتواند چند خط دستور اسمبلی را باهم به زبان ماشین تبدیل کند.

- تابعی ساختم که با گرفتن یک لیست(همان فایل کد اسمبلی مرتب شده) و label پرشی که می خواهیم آنرا تبدیل کنیم، به ما کد ماشین آن پرش را بدهد. به این گونه که تفاوت مکان حافظه label که باید به آن بپرید را با مکان حافظه دستور jmp به آن label را حساب می کند وسپس آنرا به hexadecimal تبدیل میکند و به تابع اصلی بر می گرداند.