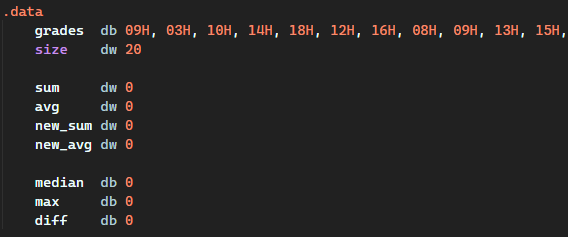
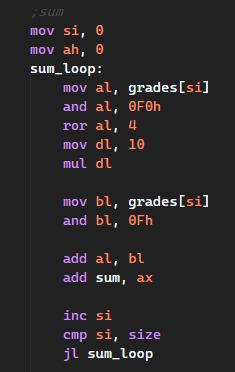
**امیرحسین احمدی 97522292 – تمرین سوم**

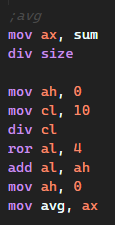
ابتدا آرایه grades شامل نمرات که به صورت bcd packed ذخیره شده اند را تعریف میکنیم. سپس متغیرهای مورد نیاز در طول الگوریتم را تعریف میکنیم.



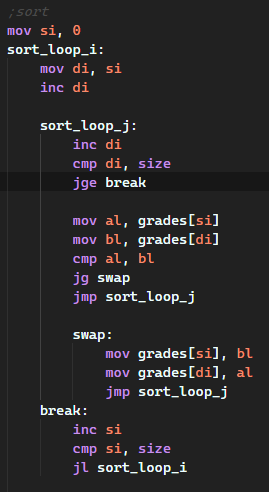
ابتدا بایست مجموع اعداد آرایه را برای محاسبه ی میانگین اولیه حساب کنیم. برای این کار عدد را از حالت bcd packed به hex تبدیل میکنیم و با متغیر sum جمع میکنیم.



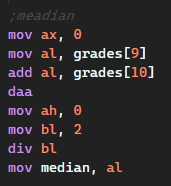
سپس برای محاسبه میانگین، sum را گرفته و بر سایز آرایه که 20 است تقسیم میکنیم. از آنجایی که sum به صورت hex است، آن را به صورت bcd packed در میاوریم.



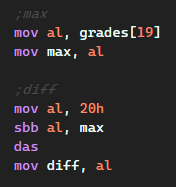
حال میخواهیم آرایه را سورت کنیم تا بتوانیم میانه را حساب کنیم. برای این کار دو بار روی آرایه فور میزنیم. برای این کار از روش selection sort استفاده میکنیم.



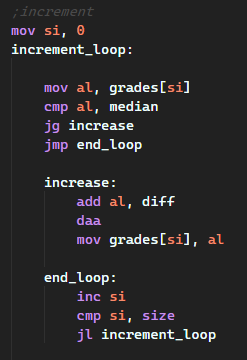
از آنجا که طول آرایه ما برابر با 20 است، میانه برابر با میانگین دو عضو وسط آن است. پس دو عدد را گرفته و جمع میکنیم (برای جمع bcd packed از دستور daa استفاده میکنیم) و آن را بر 2 تقسیم میکنیم.



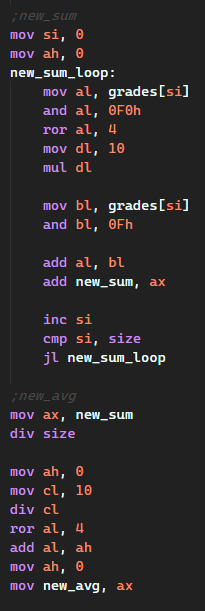
حال که آرایه سورت شده است، عدد آخر آرایه همان ماکسیموم است. آن را از 20 کم کرده تا میزانی که باید به اعضای آرایه اضافه کنیم بدست بیاید.



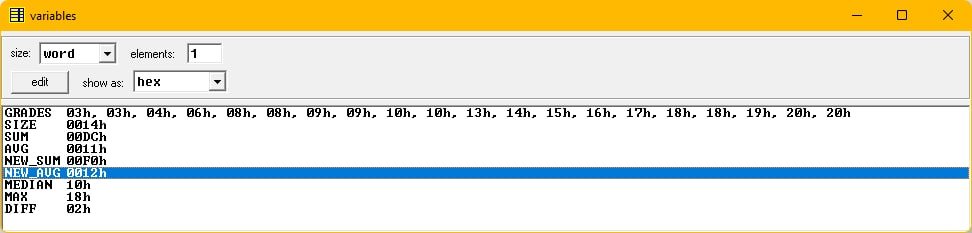
سپس روی آرایه سورت زده و اختلاف را با اعضای بای میانه جمع میکنیم.



در آخر نیز مانند همان اول کار new\_sum و new\_avg را محاسبه میکنیم.



نتیجه ران کردن کد را نیز میتوانید در زیر ببینید.



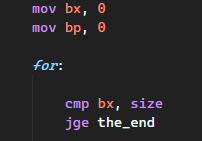


ابتدا یک آرایه شامل 20 اسم تعریف کردم، از آنجایی که لیست استرینگ نداریم، انتهای هر اسم $ گذاشته تا از هم تفکیک شوند. یک آرایه از اعداد به اسم ans نیز تعریف کردم که 20 درایه دارد و تعداد تکرار هر کلمه را در آن نگه میدارم که در ابتدا صفر است. متغیر size نیز برابر با سایز arr یعنی 20 است.

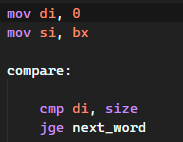


الگوریتم کلی به این شکل است که من روی آرایه اسم‌ها حرکت میکنم، به اول هر اسمی که رسیدم، از اول روی آرایه حرکت میکنم و تمام کلمات را با کلمه ی فعلی مقایسه میکنم و درصورت تشابه خانه متناظر در ans را افزایش میدهم.

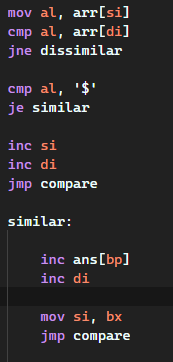
ابتدا دو متغیر bx برای حرکت رو arr و متغیر bp برای حرکت روی ans را برابر صفر(اول آرایه) میگذارم. سپس وارد for میشوم. در هر فور ابتدا چک میکنم که اگر bx به انتحای arr رسیده بود (بزرگتر از size بود) از for خارج شود و الگوریتم تمام شود.



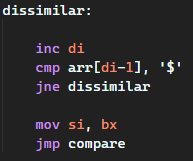
حال برای مقایسه di را روی اول arr گذاشته که بتوانم همه کلمات را به کلمه ای که bx روی آن قرار دارد مقایسه کنم و متغیر si را برابر bx گذاشته که برای هر بار مقایسه روی کلمه فعلی حرکت کرده بدون آن که bx را گم کنم. در هر بار مقایسه ابتدا چک میکنم که اگر di به انتهای arr رسیده بود مقایسه ها تمام است و باید به کلمه بعدی برویم.



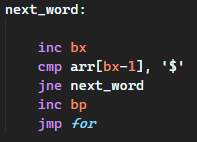
حال si و di را یکی یکی جلو برده و با هم مقایسه میکنیم. اگر شبیه به هم بودند انقدر این کار را تکرار میکنیم تا به $ برسیم و کلمه تمام شود که در این صورت به similar میرویم. در similar خانه مربوط به کلمه فعلی در ans را افزایش داده، di که روی $ است را هم یکی افزایش میدهیم تا روی حرف اول کلمه بعد برود و si را دوباره به اول کلمه فعلی میبریم تا دوباره عمل مقایسه را آغاز کنیم. اگر به کاراکتر غیر مشابه رسیدیم به dissimilar میرویم.



در dissimilar چون کلمه مشابه نبوده کاری به ans نداریم. از آنجایی که ممکن است وسط کلمه باشیم، di را آنقدر زیاد کرده تا به $ برسیم و به کلمه بعدی برویم. si را نیز دوباره به اول کلمه برده و دوباره مقایسه را آغاز میکنیم.



در انتها وقتی که مقایسه ها تمام شد، bx را که اول کلمه فعلیست آنقدر زیاد میکنیم تا به $ برسیم و به اول کلمه بعدی برویم، bp را نیز افزایش داده تا در خانه متناظر ans قرار بگیرد و دو باره for را تکرار میکنیم.



نتیجه کد ران شده را نیز میتوان در زیر دید.

