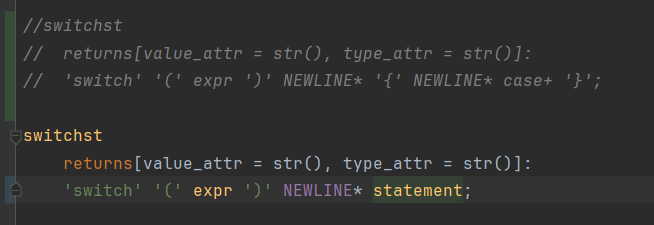
علیرضا اسلامی خواه 99521064

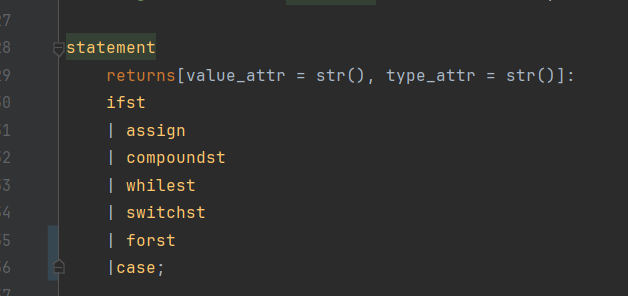
حالت اول برای این سوال انتخاب شده بود که در گام اول باید درخت را تکمیل تر میکردیم زیرا دو اشکال بزرگ داشت. اول اینکه برای switch مشخص نمیشد که روی چه متغیری انتخاب شده است و دوم اینکه برای case شرط را در درون درخت قرار نمیداد.

برای حل این مشکل درون گرامر دست به کار شدیم.

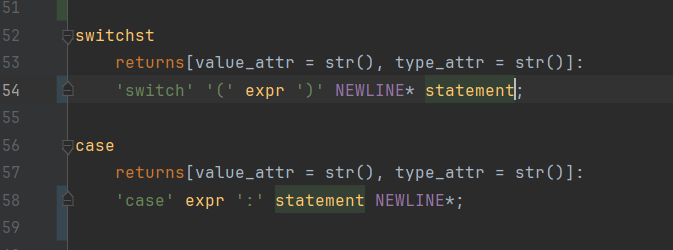
ابتدا سوییچ را عوض کردیم :



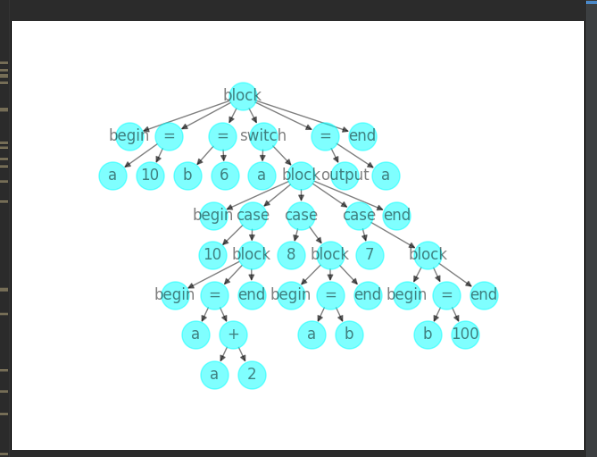
سپس درون statement کیس را هم قرار دادیم.



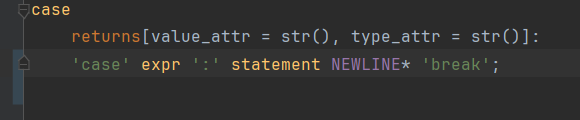
حال باید به جای INT نوعی expression میگذاشتیم تا value attrb جداگانه خودش را داشته باشد.



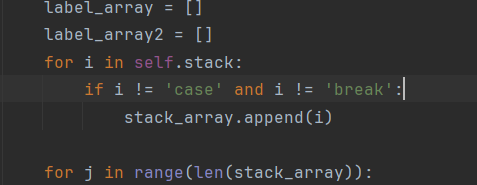
در مرحله بعد به سراغ کد il generator رفتیم که باز هم به چالش خوردیم و آن این بود که درون سوییچ یک بلاک کامل بود و نمیشد مبنایی برای مبدا آن در نظر گرفت.



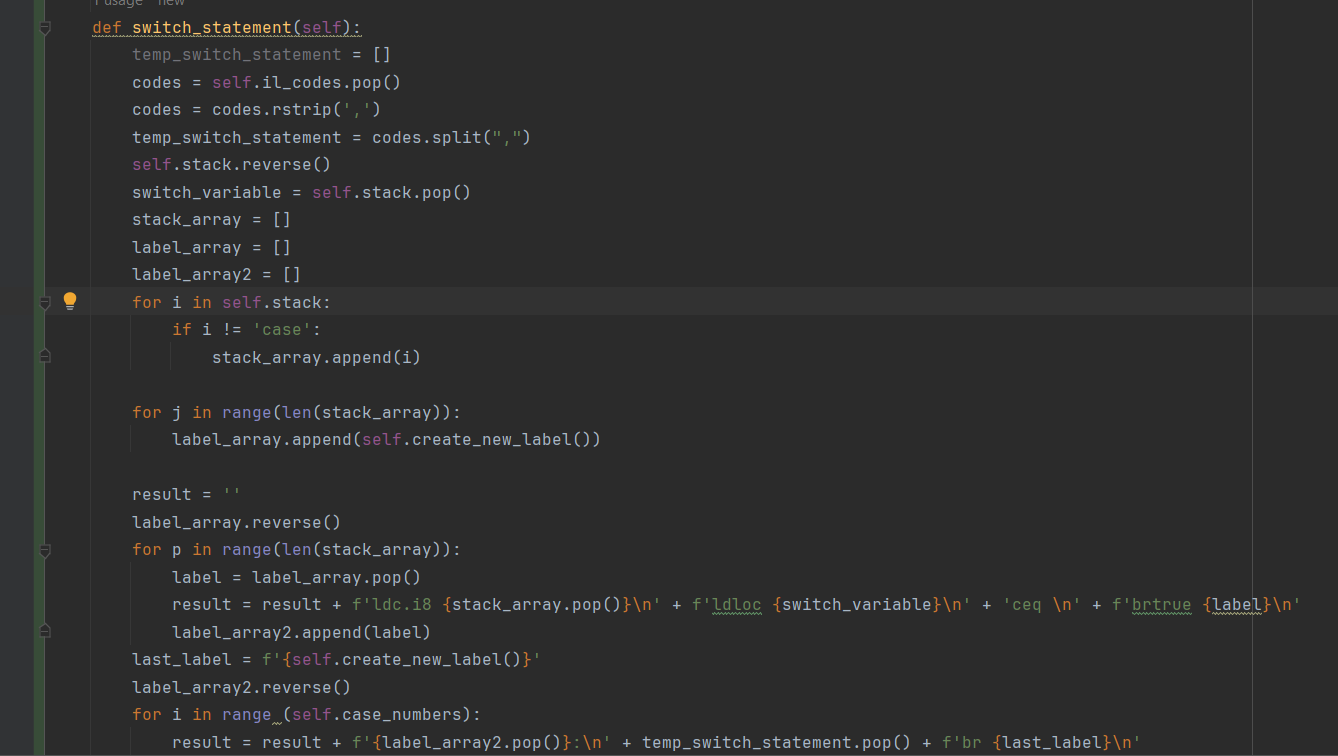
در گرامر البته میتوانستیم break را هم جایگذاری کنیم به طوری که حتما در آخر case باشد اما کمبود وقت به ما اجازه نداد. ولی اینگونه میشد :



ودرون switch statement هم بین گونه :

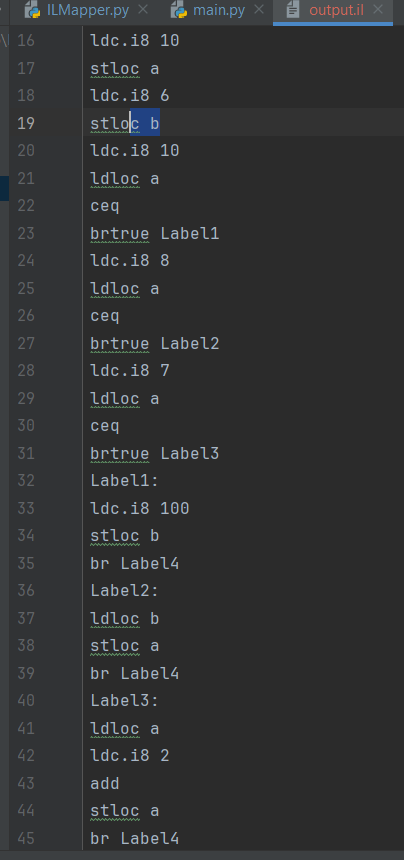


الگوریتم ما بدین صورت است که با توجه به درخت هرجا به case برخوردیم یک کاما(,) قرار دهیم. اینگونه میتوانیم کیس های یک بلاک را از هم تشخیص دهیم.



کد سوویچ کیس ما بدین صورت است که ابتدا با توجه به کاما هایی که گذاشتیم کیس هارا جدا میکنیم سپس متغیر سوییچ و شرط های آن را از استک جدا سازی میکنیم . **البته اینجا هم اگر breakداشتیم مانند بقیه case ها آن را از استک پاک میکردیم.**

در آخر هم کد را مانند چیزی که در ouput.il شرح داده شده تحویل میگیریم.



در اینجا هر شرط را بررسی کرده و با توجه به آن به برنچ مورد نظر پرش میکنیم.

در اخر هم نمایی از خروجی کامپایل شده کد il میبینیم:

