

پروژه جیمزباند

ریحانه حیدری 9731483

در این پروژه دو کلاس داریم:

❖ **Main:** ورودی می گیرد و دیکشنری مورد نظر را در **set** می ریزد و یک لیست نیز می سازیم که همه ی جوابهای نهایی ممکن در آن وجود خواهد داشت و ما هم آخرین جواب این لیست را به عنوان جمله ی خروجی که حروف آن کوچک شده و بین کلمات فاصله گذاشته شده را چاپ می کنیم.

❖ **wordBreakUtil**

در کلاس **wordBreakUtil** سه متد داریم.

🚦 **getWordFromText:** آدرس یک فایل را میگیرد و آن را باز می کند و در یک **set** وارد می کند (کلمه تکراری داخلش اضافه نمی شود و میتونیم مطمئن باشیم دیکشنری ما کلمه ی تکراری ندارد.) و **set** را برمی گرداند.

در این متد کلمات را **lower case** نیز میکنیم و در صورتی که ابتدا و انتهای آن فاصله ای وجود داشته باشد آن را نیز حذف می کنیم.

🚦 **breakStringUtil:** با توجه به آرگومان هایی که به آن می دهیم تابع **breakString** را فراخوانی می کند.

🚦 **breakString:** یک حلقه برای حرکت روی کاراکترهای جمله داریم که هر بار باتوجه به اینکه اندیس آن چند است مشخص می شود که زیررشته ما چیست و اگر آن زیررشته یکی از اعضا دیکشنری ما باشد یعنی میتواند به لیست جوابهای ما اضافه گردد و ما از کاراکتر بعدی این کار را تکرار نماییم تا کلمات جدیدی که ممکن است در دیکشنری ما باشند را یافت کنیم و آنها را به لیست جوابهایمان اضافه نماییم باتوجه به اینکه اساس کار این پروژه روی بازگشت به عقب هست پس باید شرط امید بخ بودن را طوری درنظر بگیریم که نیاز به چک کردن تمام حالات موجود نباشیم پس توجه میکنیم که هنگامی که با اضافه کردن اندیس درحال حرکت بر روی کاراکترهای جمله هستیم شرط امید بخش بودن ما که این است که در دیکشنریمان کلمه ای وجود داشته باشد که با زیررشته ی موجود ما شروع شود را درنظر بگیریم و درصورتی که به جایی رسیدیم که با اضافه کردن کاراکتر بعدی زیررشته ی ما شرط اینکه کلمه ای از دیکشنری باشد را از دست داد باید بازگشت به عقب داشته باشیم و از همان کاراکتر دوباره به روش قبل به دنبال جواب های ممکن باشیم.

نحوه محاسبه پیچیدگی زمانی:

از آنجا که این مسئله با روش بازگشت به عقب حل شده است پیچیدگی زمانی آن با روش مونت کارلو بدست می آید.

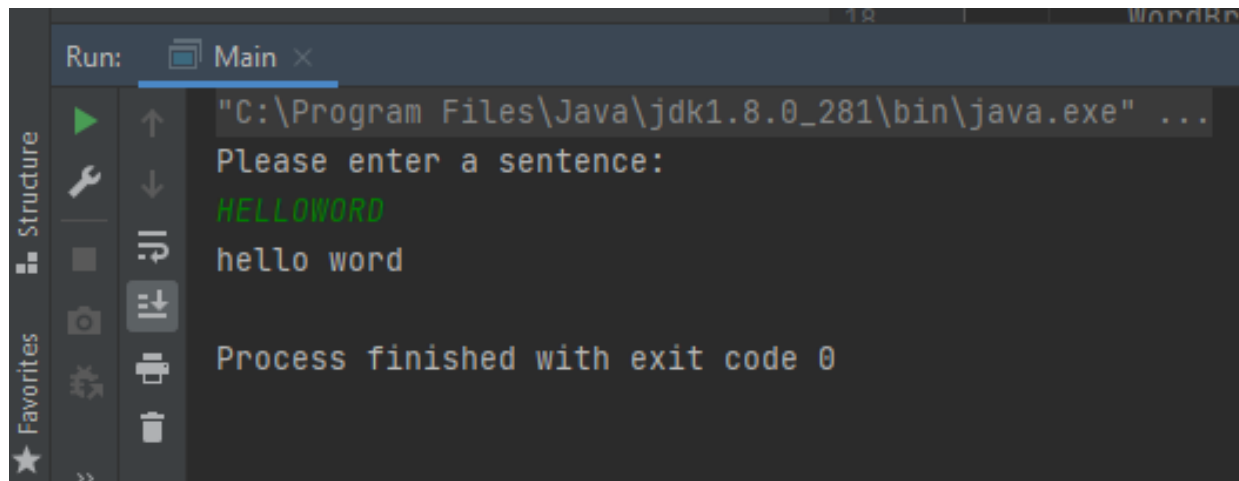
برای مثال کلمه ی فرضی زیر را در نظر بگیریم:

Abcde...

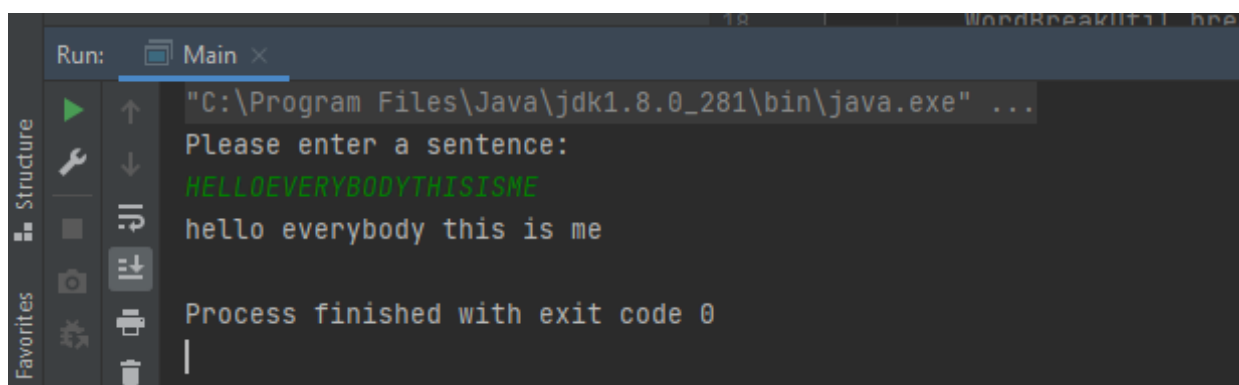
پیچیدگی زمانی:

1 + تعداد کل کلماتی که با حرف A شروع می شوند + تعداد کل کلماتی که با حرف A شروع می شوند
(تعداد کل فرزندان این سطح) * تعداد کل کلماتی که با Ab شروع می شوند (تعداد فرزندان امیدبخش
در این سطح) + تعداد کلماتی که با Ab شروع می شوند * تعداد کلماتی که با Abc شروع می شوند + ...

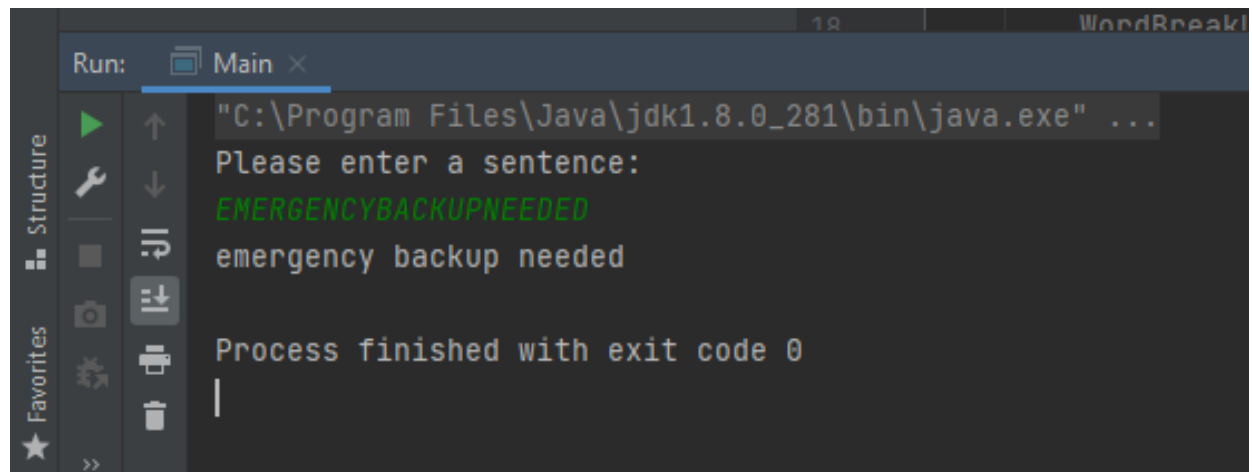
TEST CASES:



```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
Please enter a sentence:
HELLOWORD
hello word
Process finished with exit code 0
```



```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
Please enter a sentence:
HELLOEVERYBODYTHISISME
hello everybody this is me
Process finished with exit code 0
|
```



```
Run: Main x
"C:\Program Files\Java\jdk1.8.0_281\bin\java.exe" ...
Please enter a sentence:
EMERGENCYBACKUPNEEDED
emergency backup needed

Process finished with exit code 0
|
```