

توضیح کد: به پاراگراف گرفته و طول هر خط را گرفته و بهینه‌ترین حالت تقسیم‌بندی را در خروجی نشان می‌دهد.
نمونه خروجی:

```
PS D:\A.R.SH.A\Learning\term_4\Algorithm Design\projects\project2\WordWrapping> python .\main.py
Please enter your paragraph:
    adkjjsa mapdvmm KLASMVCSc KS XLKNA CKSAC;KLMSALC A;C ;ASNC SAKCNSA;NC AS;LC ASLNC
Please enter line length: 12
Please enter bigger number(line length): 15
Line 1 : adkjjsa      1 to 1
Line 2 : mapdvmm     2 to 2
Line 3 : KLASMVCSc   3 to 3
Line 4 : KS XLKNA    4 to 5
Line 5 : CKSAC;KLMSALC      6 to 6
Line 6 : A;C ;ASNC      7 to 8
Line 7 : SAKCNSA;NC     9 to 9
Line 8 : AS;LC ASLNC    10 to 11
PS D:\A.R.SH.A\Learning\term_4\Algorithm Design\projects\project2\WordWrapping> python .\main.py
Please enter your paragraph:
    Ali javad Reaxa a barbara adacav
Please enter line length: 9
Line 1 : Ali javad     1 to 2
Line 2 : Reaxa a       3 to 4
Line 3 : barbara       5 to 5
Line 4 : adacav        6 to 6
PS D:\A.R.SH.A\Learning\term_4\Algorithm Design\projects\project2\WordWrapping> python .\main.py
Please enter your paragraph:
    AAA BB CC DDDDD
Please enter line length: 6
Line 1 : AAA          1 to 1
Line 2 : BB CC        2 to 3
Line 3 : DDDDD        4 to 4
PS D:\A.R.SH.A\Learning\term_4\Algorithm Design\projects\project2\WordWrapping> |
```

توضیح الگوریتم: ابتدا پاراگراف را گرفته و آن را به کلمات تجزیه کرده و طول آن را در لیست ذخیره می‌کند. سپس در یک حلقه دوبعدی تعداد مکان‌های خالی i تا j را می‌شمرد و میزان هزینه آن را در یک آرایه دیگر ذخیره می‌کند. در آخر هم کمترین هزینه‌ها را محاسبه می‌کند که مکان‌های شکست را نشان می‌دهد.
محاسبه مرتبه زمانی: این الگوریتم 2 تا for تودرتو دارد $(2 * (n^2))$ ، که در مجموع مرتبه الگوریتم از مرتبه $O(n^2)$ است.