

## Exercise 5

จาก Lex specification ข้างล่างนี้

```
\n      { printf("1") ;}
A      { printf("2") ;}
a      { printf("3") ;}
d      { printf("4") ;}
[b-d]* { printf("5") ;}
abc    { printf("6") ;}
[A-C]* { printf("7") ;}
[a-c]* { printf("8") ;}
[^a-c]+d[^A-C]+ { printf("9") ;}
```

ถ้า input ที่รับเข้ามา เป็น ข้อความข้างล่างนี้

ABCdabc

Aabcd

BCdbA

bc

abccdba

ให้เขียนผลลัพธ์จากการประมวลผล input

จาก Lex specification ข้างบน

และบอกเหตุผลว่าทำไมถึงตัดคำได้ผลลัพธ์ดังกล่าว

Given Lex rules (verbatim ตามโจทย์กำหนด):

```
\n      { printf("1") ;}
A      { printf("2") ;}
a      { printf("3") ;}
d      { printf("4") ;}
[b-d]* { printf("5") ;}
abc    { printf("6") ;}
[A-C]* { printf("7") ;}
[a-c]* { printf("8") ;}
[^a-c]+d[^A-C]+ { printf("9") ;}
```

Input:

ABCdabc

Aabcd

BCdbA

bc

abccdba

Output:

92692151853

(assume ว่าไม่มี trailing newline)

## Exercise 5

### คำอธิบาย

จาก input:

ABCdabc

Aabcd

BCdbA

bc

abccdba

แปลงและรวมบรรทัดจะได้เป็น (เมื่อ \n แทนการขึ้นบรรทัดใหม่):

ABCdabc\nAabcd\nBCdbA\nbc\nabccdba

จาก specification ของ flex <<https://westes.github.io/flex/manual/Matching.html#Matching>> มีข้อกำหนดที่เกี่ยวข้องคือ

- If it finds more than one match, it takes the one matching the most text (for trailing context rules, this includes the length of the trailing part, even though it will then be returned to the input).

## Exercise 5

เมื่อ analyze input แล้ว จะได้ว่า

# matching	Matching	Pattern	Action
1	ABCdabc\n	[^a-c]+d[^A-C]+	{ printf("9") ; }
1.1	ABC	[^a-c]+	
1.2	d	d	
1.3	abc\n	[^A-C]+	
2	A	A	{ printf("2") ; }
3	abc	abc	{ printf("6") ; }
4	d\nBCdb	[^a-c]+d[^A-C]+	{ printf("9") ; }
5	A	A	{ printf("2") ; }
6	\n	\n	{ printf("1") ; }
7	bc	[b-d]*	{ printf("5") ; }
8	\n	\n	{ printf("1") ; }
9	abcc	[a-c]*	{ printf("8") ; }
10	db	[b-d]*	{ printf("5") ; }
11	a	a	{ printf("3") ; }