

3. Kangaroo Words คือคำที่มีความหมายเดียวกันแต่ซ่อนอยู่ในคำที่มีความยาวมากกว่า เช่น AffeCT มีคำว่า ACT ซ่อนอยู่ AREnA มีคำว่า AREA ซ่อนอยู่ BLOssOM มีคำว่า BLOOM ซ่อนอยู่ จงเขียนโปรแกรมเพื่อทดสอบว่ามีคำขนาดเล็กซ่อนอยู่ในคำขนาดใหญ่
 hint ใช้ Loop แบบเดียวกันกับการบวกพหุนาม และไม่จำเป็นต้องใช้ Linked List ใช้ time complexity ไม่เกิน $O(n)$

คำข้างหน้ากับข้างหลังจำเป็นต้องเรียงกันใหม่ครับ เช่น AffeCT กับ CAT คือ kangaroo ใหม่ครับ

หากเรียง :

time complexity = $O(N)$

Code : <https://ideone.com/SoSbD0>

```

1 #include<iostream>
2 #include<string.h>
3 using namespace std;
4
5 int main()
6 {
7     char str1[10010], str2[10010];
8     cout << "Enter Words : ";
9     cin >> str1;
10    cout << "Find : ";
11    cin >> str2;
12    int now = 0;
13    for(int i = 0; i < strlen(str1); i++)
14    {
15        if(str2[now] == str1[i]){now++;}
16    }
17    (now == strlen(str2)) ? cout << "It's Kangaroo" : cout << "It's not Kangaroo";
18 }
```

OUTPUT

```

Enter Words : EWWssssa
Find : EWsa
Kangaroo Words
```

หากไม่เรียง :

time complexity = $O(N+M) = O(N)$

Code : <https://ideone.com/G5j7IP>

```
1 #include<iostream>
2 #include<string.h>
3 using namespace std;
4
5 int main()
6 {
7     int check[130] = {};
8     char str1[10010], str2[10010];
9     cout << "Enter Words : ";
10    cin >> str1;
11    cout << "Find : ";
12    cin >> str2;
13    for(int i = 0; i < strlen(str1); i++)
14    {
15        check[str1[i]]++;
16    }
17    for(int i = 0; i < strlen(str2); i++)
18    {
19        if(check[str2[i]]-- <= 0){cout << "It's not Kangaroo"; return 0;}
20    }
21    cout << "It's Kangaroo";
22 }
```

OUTPUT

```
Enter Words : WEQSAdsdxcS
Find : QEW
It's Kangaroo
```