

Cách viết 2, 3, 4 lines

Người tổng hợp + soạn file: @apple1835

Nguồn gốc phương pháp spell: tieba, @I like turtle, @580vnx_tricker

File dành cho những người đã thạo spell 1 line

| là con trỏ trên máy tính :v

Ôn lại bài cũ :))

a, Công thức lấy các kí tự in đậm và @

A	SHIFT 7 3 7
B	SHIFT 7 3 8
C	SHIFT 7 3 9
D	SHIFT 7 3 (-)
E	SHIFT 7 3 °,,”
F	SHIFT 7 3 x ¹
(4 D) @	SHIFT 7 4 8
(4 E) @	SHIFT 7 4 9
(4 F) @	SHIFT 7 1 4

Khái niệm mới

a, Lựa chọn số an

110an: 34 bytes

100 + 60an: 34 + 60 = 94 bytes

100 + 64an: 34 + 64 = 98 bytes

...

- Tùy theo việc câu bạn dài hay ngắn, chỉ cần lựa chọn số an có đủ bytes cho câu của mình là được rồi.

- VD:

+ Bạn muốn viết một câu như sau:

`HiImJustALittleCatWhoIsVeryCute...`

+ Câu trên chỉ cần dùng có mỗi 34 bytes nên ta có thể dùng 110an hoặc tối đa 100+24an và làm như spell 1 line TV, chỉ thay đổi hackstring lúc cuối để biến nó thành 2/3/4 lines.

- Trong file mình tạm thời sẽ dùng 100+64an nhé.

b, Cache Area

- Là một khu vực dùng để lưu trữ dữ liệu sau khi ấn **CALC** hoặc **=**
- Công dụng:
 - Chuyển đổi các kí tự có Công thức 2 trong file Character Set.docx
 - Nhân đôi, ...
- Ta có thể truy cập Cache Area bằng cách dùng:
 - Hex: 19
 - Token:

Vì nó có hex là 19 nên ta cứ nhét 19 vào ABC là được

HD Spell 2 lines TV

B1:

SHIFT **MENU** **1** **3**

$x = \sum(x, 9, 999)$

CALC **=** **AC** **◀** (**=** **AC** phải ấn nhanh)

DEL **DEL**



B2:

-
- Lấy các chữ cái in đậm mà bạn cần rồi nhập vào form như sau: (chi tiết ở B2 và B3 trong file How To Spell 1 Line Vietnamese.docx).

x:

A = 1. 0 0 0 0 **F B** 0 1 _ _ _ _

_ $\times 10$:

B = 1. _ _ _ _ _ _ _ _ _ _ _ _

_ $\times 10$:

C = 1. _ _ _ _ _ _ _ _ _ **6 2 1**

9 $\times 10$ 2 3

Chỗ màu vàng là những chỗ tuyệt đối không được để chữ in đậm vào
Giống như 1 line, 621923 sẽ ở cuối sau khi mình đặt được tất cả các hex cần
vào ABC

-

B3:

$x = \sum(x, 9, 9)$

CALC **=** **◀**

SHIFT **◻** **SHIFT** **◻**

◀ **◀** **DEL** **▼**

SHIFT **8** **▼** **2** **6**

◀ **◀** **▶**

9 **DEL** **◀**

)

+

<100 số bất kì>

+

<64 số bất kì>

CALC **=**

B4:

ON

x=Σ(x,9,999

CALC **=** **AC** **◀** (**☰** **AC** phải ấn nhanh)

DEL **DEL**

CALC **=** **◀**

B5:

- **SHIFT** **◦**
- Lấy **tối đa** 3 cái @ theo thứ tự **(4F)@**, **(4E)@**, **(4D)@** rồi nhập theo form như sau:

Màn hình hiển thị:

```
x :
(4F)@=C :
(4E)@=B :
(4D)@=A
```

(4F), (4E), (4D) ở đây là hex của mấy cái kí tự @ đó chứ không phải là nhập cái (4F), (4E), (4F) đó vào máy tính

- **CALC** **=** **=** **=** **=** **=** **=** **=** **=** **=** **AC** **▼**
ye chính xác là ấn **☰** 9 lần nhé :))

- Nếu bạn thấy nó ra:

><đây kí tự>Q(□

thì xin chúc mừng, bạn đã hoàn thành được chặng đường đầu tiên r xDD

B6:

- Xoá kí tự thừa trong câu của bạn (HD chi tiết ở B6 trong file How To Spell 1 Line Vietnamese.docx).

B7:

- Đưa con trỏ về đầu rồi gõ **Z**
-  × 23 lần

B8:

$x = \sum(x, 9, 999)$

    (  phải ấn nhanh)

⬅ ⬅ DEL ⬇

SHIFT 8 ⬇ 2 6

⬅ ⬅ ▶

9 DEL ⬅

)

+

<100 số bất kì>

+

<64 số bất kì>

CALC =

B9:

- ⬇ (Nếu vẫn ra những kí tự mình nhập ban nãy là ok)
- ▶ ▶ ▶ DEL để xoá cái <
- Đưa con trỏ đến |Q(rồi nhập:

SHIFT 7 3 (-)

Màn hình hiển thị:

```
z<dãy kí tự>λcPQ(□
```

- Đưa con trỏ đến |Q(nhập **khoảng** 50 số bất kì sau đó **CALC** **AC** **▶**
Chi tiết ở bước hướng dẫn convert kí tự chỗ cuối
- Lại đưa con trỏ đến |Q(spam đến khi kẹt không còn nhập nổi nữa rồi **▶** **▶** **DEL**
- **▶** đến |λcP bấm **9** **DEL** để chuyển đổi kí tự đó sang token.
- **AC** **▶**

Màn hình hiển thị:

```
<dãy kí tự>D<số>Q(□
```

B10:

- Đưa con trỏ trở tới |Q(rồi xoá hết số vừa này nhập.

B11:

- Đưa con trỏ lên đầu tiên, nhập đủ **34 kí tự**.

B12: Tìm số chữ số cần nhập

- Kí tự dạng Fxyy, như kí tự TV: 2 bytes
- Dấu cách nhập bằng **SHIFT** **8** **3** **4**: 2 bytes
- Những kí tự còn lại: 1 byte

98 bytes – (tổng số bytes trong câu của bạn) = ??

- Sau khi đã tính toán xong thì ta sẽ nhập ?? số vào sau **câu của mình (34 kí tự)**.

B13:

- Cuối cùng, bạn nhập theo form như sau :

Màn hình hiển thị :

```
<17 kí tự dòng 2>  
<17 kí tự dòng 1>  
<số chữ số cần nhập>  
└─D21Q (F2x
```

-  

HD Spell 3 lines TV

...

B9:

-  (Nếu vẫn ra những kí tự mình nhập ban nãy là ok)
-     để xoá cái 
- Đưa con trỏ đến  rồi nhập:

SHIFT 7 3 (-)

SHIFT 8 1 (-)

SHIFT 7 3 (-)

SHIFT 8 1 (-)

Màn hình hiển thị:

```
z<dãy kí tự>
m n mile λ c p m n mile λ c p m n
mile Q (□
```

- Đưa con trỏ đến |Q(rồi nhập **khoảng** 50 số bất kì rồi CALC AC ▶
Chi tiết ở bước hướng dẫn convert kí tự chỗ cuối
- Lại đưa con trỏ đến |Q(rồi spam đến khi kẹt không còn nhập nổi nữa rồi ▶▶ DEL
- ▶ đến |m n mile bấm 9 DEL để chuyển đổi kí tự đó sang token, làm tương tự với những cái còn lại là λ c p m n mile λ c p m n
mile
- AC ▶

Màn hình hiển thị:

```
<dãy kí tự>  
@D@D@<số>Q (□
```

B10:

- Đưa con trỏ trở tới |Q (rồi xoá hết số vừa này nhập

B11:

- Đưa con trỏ lên đầu tiên, nhập đủ **51 kí tự**.

B12: Tìm số chữ số cần nhập

- Kí tự dạng Fxyy, như kí tự TV: 2 bytes
- Dấu cách nhập bằng **SHIFT** **8** **3** **4**: 2 bytes
- Những kí tự còn lại: 1 byte

98 bytes – (tổng số bytes trong câu của bạn) = ??

- Sau khi đã tính toán xong thì ta sẽ nhập ?? số vào sau **câu của mình (51 kí tự)**.

B13: Lấy kí tự F

- Đưa con trỏ tới |Q(rồi bấm:



B14:

- Cuối cùng, bạn nhập theo form như sau :

Màn hình hiển thị :

```
<17 kí tự dòng 3>
<17 kí tự dòng 2>
<17 kí tự dòng 1>
<số chữ số cần nhập>
=@2J▶cal2┘D21=@2Ran#2┘D21=@
2km▶mile2Q(F2x
```

km ▶ mile	SHIFT 8 1 8
J ▶ cal	SHIFT 8 ▼ 3 3

- CALC ≡

HD Spell 4 lines TV

...

B9:

▼ (Nếu vẫn ra những kí tự mình nhập ban này là ok)

▶▶▶ DEL để xoá cái <

- Đưa con trỏ đến |Q(rồi nhập:

SHIFT 8 1 (-)

SHIFT 7 3 (-)

SHIFT 8 1 (-)

SHIFT 7 3 (-)

SHIFT 8 1 (-)

SHIFT 7 3 (-)

SHIFT 8 1 (-)

Màn hình hiển thị:

```
z<dãy kí tự>  
m▸n mileλcPm▸n mileλcPm▸n  
mileλcPm▸n mileQ(□
```

- Đưa con trỏ đến |Q(rồi nhập **khoảng** 50 số bất kì rồi CALC AC ►

Chi tiết ở bước hướng dẫn convert kí tự chỗ cuối

- Lại đưa con trỏ đến |Q(rồi spam đến khi kết không còn nhập nổi nữa rồi ► ► DEL
- ► đến |m▸n mile bấm 9 DEL để chuyển đổi kí tự đó sang token, làm tương tự với những cái còn lại là λcPm▸n mileλcPm▸n mileλcPm▸n mile
- AC ►

Màn hình hiển thị:

```
<dãy kí tự>  
@D@D@D@<số>Q(□
```

B10:

- Đưa con trỏ trở tới |Q(rồi xoá hết số vừa nhập.

B11:

- Đưa con trỏ lên đầu tiên, nhập đủ **68 kí tự**.

B12: Tìm số chữ số cần nhập

- Kí tự dạng Fxyy, như kí tự TV: 2 bytes
- Dấu cách nhập bằng **SHIFT** **8** **3** **4**: 2 bytes
- Những kí tự còn lại: 1 byte

98 bytes – (tổng số bytes trong câu của bạn) = ??

- Sau khi đã tính toán xong thì ta sẽ nhập ?? số vào sau **câu của mình (68 kí tự)**.

B13: Lấy kí tự F

- Đưa con trỏ tới |Q(rồi bấm:



B14:

- Cuối cùng, bạn nhập theo form như sau :

Màn hình hiển thị :

```
<17 kí tự dòng 4>  
<17 kí tự dòng 3>  
<17 kí tự dòng 2>  
<17 kí tự dòng 1>  
<số chữ số cần nhập>  
=@2111┘D21=@21e2┘D21=@  
2pc ▶km2┘D21=@2in ▶cm2Q(F2x
```

pc ▶km	SHIFT 8 1 ...
in ▶cm	SHIFT 8 1 1

-  

HD convert kí tự dùng Công thức 2 từ Cache Area

Ta sẽ làm cái này sau khi vô trình edit (nên làm sau khi vô trình edit lần 2)

- Giả dụ ta cần có chữ T :
 - Hex : 54
 - Token : @
 - Công thức 2 :    
- Sẽ chỉ có 2 cách duy nhất ta có thể lấy được chữ T
 - Nhập hex của chữ T vô ABC.
 - Dùng công thức 2 sau đây chuyển đổi sang.
- Trong bước này thì mình sẽ hướng dẫn bạn chuyển đổi kí tự dùng Công thức 2.

Màn hình hiển thị ban đầu:



B1 :

- Đưa con trỏ về đầu tiên r gõ **Z**

B2 :

- Nhập kí tự cần chuyển đổi bằng công thức 2 vào vị trí của nó. VD mình muốn nhập chữ T sau chữ A thì ta đưa con trỏ ra A | rồi bấm **SHIFT** **7** **1** **6**

Màn hình hiển thị:

```
zAnsZo|Q(□
```

Máy tính sẽ chỉ hiển thị token của kí tự đó trong trình edit nên đừng hỏi tại sao chữ A lại là Ans, vì token của A là Ans :v

B3:

- Dưới đây là số bytes tối đa mà Cache Area có thể chịu được được tùy theo số an:
 - + 110an: 65 bytes
 - + 100+64an: $65 + 64 = 129$ bytes

...

- Rồi ta dùng công thức sau:

<số bytes tối đa> - <số bytes trên màn hình> = ??

- Đưa con trỏ ra |Q(gõ ?? số (*quá 1 số là lag luôn :v*)
- [CALC] [AC] [▶]

B4 :

- Lại đưa con trỏ ra |Q(rồi spam số đến khi kẹt thì thôi
- [▶] [▶] [DEL]

B5 :

- [▶] đến trước kí tự cần chuyển đổi (ở đây là |Z0) rồi ấn [9] [DEL]
- [AC] [▶]

Màn hình hiển thị :

```
<dãy kí tự>  
Ans@<số>Q(□
```

B6 :

- Đưa con trỏ tới |Q(rồi xoá hết số vừa nhập

Màn hình hiển thị:

Ans@Q(□

Vậy là bạn đã hoàn thành việc convert chữ T dùng công thức 2 từ Cache Area :)

HD sử dụng F4F4

Ta sẽ làm cái này sau khi vô trình edit (nên làm sau khi vô trình edit lần 2)

F4F4 không cần dùng đến Cache Area

❖ Cách truy cập:

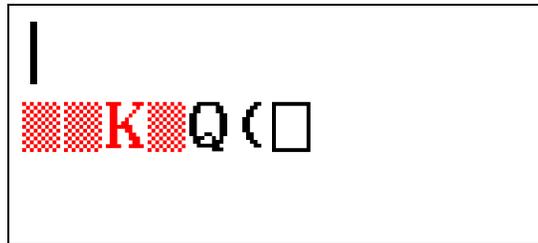
- Ta chỉ cần gõ **F4F4** vào ABC là ta sẽ có được nó trong trình edit :)

❖ Có 3 cách để gán F4F4 trong trình edit:

- Direct
- Indirect
- Character Conversion

➤ Direct:

Màn hình hiển thị ban đầu:



■■■K■■■ là 1 unstable char nên nó có thể xuất hiện ở nhiều dạng khác nhau tùy trường hợp.

- Giả sử ta muốn biến **F4F4** thành **F430 F431** thì ta thấy 30 có Ct1 là 0, 31 có Ct1 là 1 nên ta sẽ chỉ cần đưa con trỏ ra **F4F4** rồi bấm **1** ◀ **0** ◀

Màn hình hiển thị:



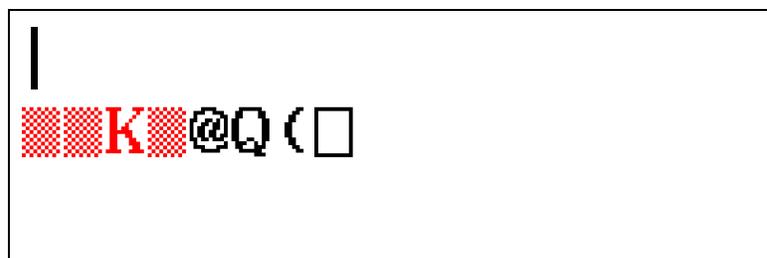
Congrats, vậy là bạn đã thành công biến F4F4 thành F430 F431 rồi :->!

➤ **Indirect:**

- Giả sử ta muốn có F480, nhưng 80 không thể dùng Ct1 cũng không thể dùng Ct2, ta có thể dùng ABC kiểu như này:

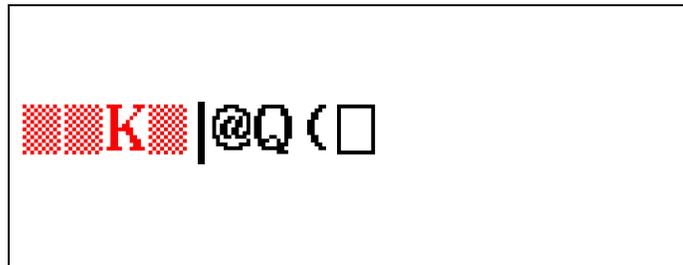
A=1.0000F4F480481
9_{x10}23

Màn hình hiển thị:



- Ta để F4F4 và 80 đứng cạnh nhau trong trình edit sau đấy đưa con trỏ qua F4F4|80

Màn hình hiển thị:



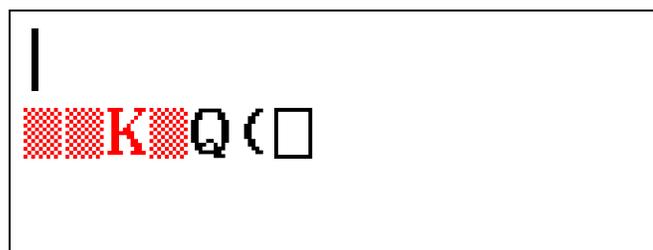
- ◀
- Sau đấy ta được quyền gán thêm 1 kí tự có Ct1 nữa, giả sử như là 0x31. Ta bấm thêm 1 ▶

Chúc mừng, vậy là ta đã thành công biến F4F4 thành F431 F480!

➤ Character Conversion:

Đây là cách gán F4F4 với kí tự dùng Ct2, đáng buồn là cách gán này chỉ biến F4F4 thành duy nhất 1 kí tự :(

Màn hình hiển thị ban đầu:



- Giả sử ta muốn có F420, mà 20 chỉ có Ct2 là
SHIFT **8** **▼** **2** **6**, ta đưa con trỏ ra **F4F4** rồi
bấm **SHIFT** **8** **▼** **2** **6**

Màn hình hiển thị:



- Bấm **◀** **9** **DEL** **▶** **9** **DEL** **◀**

Chúc mừng, vậy là bạn đã thành công biến F4F4 thành F420 r đó :)

Lưu ý nhỏ: Bạn nên đặt Ran# sau F4F4 rồi mới đi xoá các kí tự thừa, sau đó mới gán F4F4. Gán xong thì ta có thể xoá cái Ran# đi.

HD nhân đôi, sắp xếp kí tự

Ta cũng sẽ bắt đầu làm cái này sau khi vô trình edit lần 2

a, Hướng dẫn nhân đôi kí tự :>

B1:

- Ta đưa con trỏ lên đầu tiên và gõ **Z**

B2:

-   

Bước này nhằm lưu dữ liệu trên màn hình vô Cache Area

B3:

- Ta đưa con trỏ ra **|Q(** spam số đến khi kẹt thì thôi

Màn hình hiển thị:

```
z<dãy kí tự><số>Q(□
```

-   

B4:

-  rồi  hết tất cả chỉ giữ lại những kí tự cần nhân đôi.

b, Hướng dẫn sắp xếp kí tự :v

Nên dồn hết các kí tự TV lên đầu để sắp xếp cho dễ

- Thực chất nó chính là dùng nhân đôi kí tự để sắp xếp.
- Giả sử bạn muốn acbd thành abcd thì bạn sẽ nhân đôi acbd thành **acbdacbd** và xoá những kí tự thừa (không in đậm :v) để ra được abcd.

HD dùng <01>

- Rất đơn giản, ta chỉ cần hiểu <01> là một kí tự:
 - 1 byte.
 - = 5/11 dấu cách bình thường khi hiển thị.
- Cách lấy:
 - Convert từ Công thức 2.
 - Nhét vô ABC

Byte <01> chỉ = 5/11 dấu cách bình thường nên 1 dòng có thể chứa tối đa 39 kí tự <01>, tổng độ dài của của câu ko đc vượt quá 17,9 khi có chứa <01>