Frequently Encountered Problem

20160206
Code below are simplized.
Finding out mistakes is not easy, because there are actually several hundred lines in code.

1. Minus value

```
wvariable value
: calc value W@ d1000 - value W! value W@ . ;
```

Word'calc' correctly operate if value's data is more than d1000.
But if it want to correctly operate any value, value should define 'variable'.

2. Array

```
fl
variable data1 d150 allot
```

When copy/paste this, it occur error on PropForth.

```
Prop0 Cog6 ok
– Executing copy/paste –
fl                          ← Echo from PropForth
Prop0 Cog6 ok
CON:Prop0 Cog5 RESET – last status: 246 UNKNOWN ERROR
```

Because word'fl' use input buffer. (refering word'(fl) inside StartKernel-1.f)
In case of DevKernel, error don't happen at d137.

To avoid this error, memory should be allocated to data1 before loading forth codes.
This is performed by manual.
Procedure; variable data1 here W@ d146 + here W!
[here] is top address for free space.
(1) Allocating 4byte to data1 (because it is variable)
(2) Loading [here] on stack
(3) Add d146(150-4=146) to [here]address
(4) Save value to here

3. Miss-alignment

Case of getting data

```
Copy/paste statements after reboot on DevKernel

wvariable t
wvariable Font -2 allot
h10 c, h20 c, h30 c, h40 c, h50 c, h60 c, h70 c, h80 c,

: test1 Font C@ . ;
: test2 Font W@ . ;
: test3 Font L@ . ;
```

'test1' and 'test2' have no problem. Result is h40302010.
Result of test3 might be h2010004F.

```
Font 4 dump

44CE 0004:
44CE: 10 20 30 40 50 60 70 80 C& 44 85 74 65 73 74 31  . O@P'p..D.test1
```

[Font L@] get address[44CC].
Alignment of 'wvariable' and 'variable' is different.
If needing 'Font L@', code be should 'variable Font -4 allot'.

Case of storing data

Copy/paste statements after reboot on DevKernel

```
here W@ .
17596 Prop0 Cog6 ok  ← Top address of free area after rebooting
hex
Prop0 Cog6 ok
```

Next, Loading code below;

```
fl

0 c, 0 c,
wvariable che d18 allot  \ 20byte
: clear
che
5 0 do
    dup i 4 * + h12345678 swap L!
loop
drop
;
```

Checking memory;

```
hex
Prop0 Cog6 ok
d17596 50 dump

44BC 0050:
44BC: 00 00 52 42 83 63 68 65 4F 00 00 00 00 00 00 00   ..RB.cheO.......
44CC: 00 00 00 00 00 00 00 00 00 00 00 00 C0 44 85 63   ............D.c
44DC: 6C 65 61 72 C4 44 5D 00 05 00 C6 0C 79 00 C4 0E   lear.D].....y...
44EC: 60 1A 5D 00 04 00 CC 3C 66 0F 57 00 78 56 34 12   `.]....<f.W.xV4.
44FC: EA 0F 20 0E 80 00 E8 FF 26 00 61 00 00 00 00 00   .. .....&.a.....
Prop0 Cog6 ok
che .
44C6 Prop0 Cog6 ok
```

Executing word”clear”;

```
clear
Prop0 Cog6 ok
clear

CON:Prop0 Cog6 RESET - last status: 2 RETURN STACK OVERFLOW

Prop0 Cog6 RESET - last status: 2 RETURN STACK OVERFLOW
Prop0 Cog6 ok
che .

CON:Prop0 Cog6 RESET - last status: 2 RETURN STACK OVERFLOW

Prop0 Cog6 RESET - last status: 2 RETURN STACK OVERFLOW
Prop0 Cog6 ok
```

Checking memory;

```
d17596 50 dump

44BC 0050:
44BC: 00 00 52 42 83 63 68 65 78 56 34 12 78 56 34 12   ..RB.chexV4.xV4.
44CC: 78 56 34 12 78 56 34 12 78 56 34 12 C0 44 85 63   xV4.xV4.xV4..D.c
44DC: 6C 65 61 72 C4 44 5D 00 05 00 C6 0C 79 00 C4 0E   lear.D].....y...
44EC: 60 1A 5D 00 04 00 CC 3C 66 0F 57 00 78 56 34 12   `.]....<f.W.xV4.
44FC: EA 0F 20 0E 80 00 E8 FF 26 00 61 00 00 00 00 00   .. .....&.a.....
Prop0 Cog6 ok
```

Because of miss-alignment, data is writing in incorrect area.
So, error occuur.

Code should be below;

```
fl

0 c, 0 c,
variable che d16 allot  \ 20byte
: clear
che
5 0 do
    dup i 4 * + h12345678 swap L!
loop
drop
;
```

4. Changing dictionary by mistake

```
variable array 6 allot
: test1 d10 0 do i . loop ;
: test2 array d14 0 do dup 0 swap C! 1+ loop drop ;
```

After executed 'test2', any word can't execute.
Because lfa of 'test1' is deleted by 'test2'.

5. Prompt”Cog6 ok” don't be displayed when copy/paste statements.

When copy/paste whole forth statements from editor to TeraTerm, sometimes prompt still display “Cog5 ok”.
Prompt”Cog6 ok” return back by hitting return-key.
This is bug in “fl”.
To avoid, last line be must empty line.

```
fl
…
…
    ← Last line must not include any character (space or tab, other)
```