```
 1 #include <stdio.h>
 2
 3 int NUM = 1000;
 4 char HI[] = "Hi!";
 5
 6 int static_counter_fun() {
 7    static int ctr = 0; // Only happens on the first call!
 8    ctr += 1;
 9    printf("ctr:\t%d\n", ctr);
10    printf("&ctr:\t%p\n", &ctr);
11    return ctr;
12 }
13
14 int main(int argc, char** argv) {
15    char hello[] = "hello everyone";
16    char* ptr = hello;// could also write &hello
17    char* hiptr = HI; // could also write &HI
18    int* numptr = &NUM; // could NOT also write NUM
19    printf("&main:\t%p\n", &main);
20    printf("&main:\t%p\n", &main);
21    printf("&NUM:\t%p\n", &NUM);
22    printf("numptr:\t%p\n", numptr);
23    printf("&HI:\t%p\n", &HI);
24    printf("hiptr:\t%p\n", hiptr);
25    printf("stdin:\t%p\n", &stdin);
26    static_counter_fun();
27    static_counter_fun();
28    printf("&argv:\t%p\n", &argv);
29    printf("ptr:\t%p\n", ptr);
30    printf("&ptr:\t%p\n", &ptr);
31    printf("&hiptr:\t%p\n", &hiptr);
32    printf("hello:\t%p\n", hello);
33    printf("argv:\t%p\n", argv);
34    printf("argv[0]:%p\n", argv[0]);
35 }
```

```
$ ./layout
&static_counter_fun:     0x6046a1dcf169
&main:   0x6046a1dcf1c2
&NUM:    0x6046a1dd2010
numptr: 0x6046a1dd2010
&HI:     0x6046a1dd2014
hiptr:  0x6046a1dd2014
stdin:  0x6046a1dd2020
ctr:    1
&ctr:   0x6046a1dd202c
ctr:    2
&ctr:   0x6046a1dd202c
&argv:  0x7ffe89c5a5b0
ptr:    0x7ffe89c5a5d9
&ptr:   0x7ffe89c5a5c0
&hiptr: 0x7ffe89c5a5c8
hello:  0x7ffe89c5a5d9
argv:   0x7ffe89c5a718
argv[0]:        0x7ffe89c5c73c
```

```c
#include <stdio.h>
#include <string.h>



int parse_args(char* input, char** args) {
  char* current = strtok(input, " ");
  int current_index = 0;
  while(current != NULL) {
    args[current_index] = current;
    current = strtok(NULL, " ");
    current_index += 1;
  }
  return current_index;
}

int main() {
  while(1) {
    printf(" ");
    char input[2048];
    fgets(input, 2048, stdin);
    printf("Now computer run this: %s\n", input);

    char* args[1000];
    int argc = parse_args(input, args);
    for(int i = 0; i < argc; i += 1) {
      printf("args[%d]: %s\n", i, args[i]);
    }













  }
}
```