Given the following C code and output run, draw the corresponding memory map

Page 1 of 2
- Mapping the print statements to the output statements

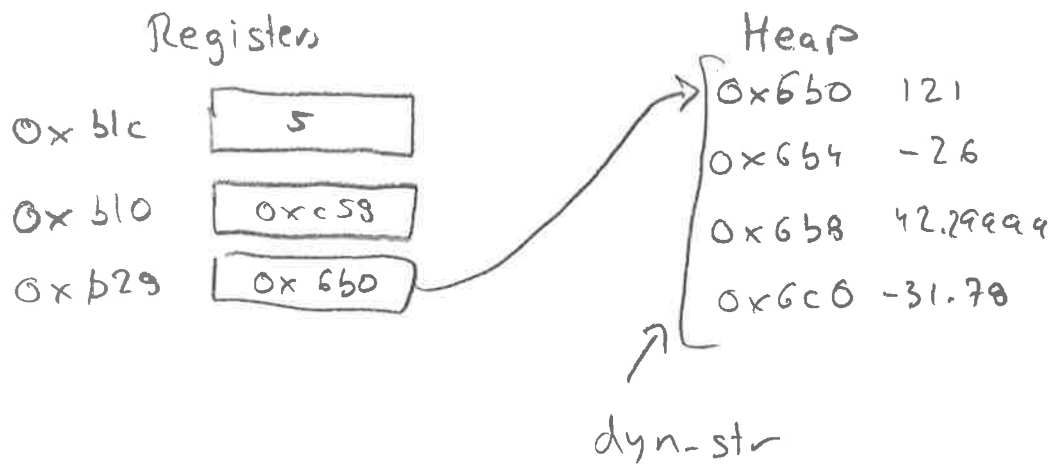
```
#include <stdio.h>
#include <stdlib.h>

typedef struct prob_struct{
  unsigned int the_uint;
  int the_int;
  float the_float;
  double the_double;
}prob_struct;
                              argc = 5
int main( const int argc, const char* argv[] ){

1  fprintf( stdout, "%p %d\n", &argc, argc );
2  fprintf( stdout, "%p %p\n", &argv, argv );

  int iter;
  for( iter = 0; iter < argc; ++iter ){           argc = 5 => 3 - 7
    fprintf( stdout, "%p %p %s\n", &argv[iter], argv[iter], argv[iter] );
  }

  prob_struct static_str = { (unsigned int) atoi(argv[1]), atoi(argv[2]), (float)atof(argv[3]), atof(argv[4]) };

8  fprintf( stdout, "%p\n", &static_str );
9  fprintf( stdout, "%p %u\n", &static_str.the_uint, static_str.the_uint );
10 fprintf( stdout, "%p %d\n", &static_str.the_int, static_str.the_int );
11 fprintf( stdout, "%p %f\n", &static_str.the_float, static_str.the_float );
12 fprintf( stdout, "%p %f\n", &static_str.the_double, static_str.the_double );

  prob_struct* dyn_str = (prob_struct*)malloc( sizeof(prob_struct) );
  dyn_str->the_uint = (unsigned int) atoi(argv[1]);
  dyn_str->the_int = atoi(argv[2]);
  dyn_str->the_float = (float)atof(argv[3]);
  dyn_str->the_double = atof(argv[4]);

13 fprintf( stdout, "%p %p\n", &dyn_str, dyn_str );
14 fprintf( stdout, "%p %u\n", &dyn_str->the_uint, dyn_str->the_uint );
15 fprintf( stdout, "%p %d\n", &dyn_str->the_int, dyn_str->the_int );
16 fprintf( stdout, "%p %f\n", &dyn_str->the_float, dyn_str->the_float );
17 fprintf( stdout, "%p %f\n", &dyn_str->the_double, dyn_str->the_double );

  free(dyn_str);

  return EXIT_SUCCESS;
}
```

Output: from ./mem_map 121 -26 42.3 -31.78 ← Step 1: Count the input argument

```
1  0x7ffca3cf9b1c 5  ←———— argc = 5
2  0x7ffca3cf9b10 0x7ffca3cf9c58
3  0x7ffca3cf9c58 0x7ffca3cfba1f ./mem_map
4  0x7ffca3cf9c60 0x7ffca3cfba29 121
5  0x7ffca3cf9c68 0x7ffca3cfba2d -26
6  0x7ffca3cf9c70 0x7ffca3cfba31 42.3
7  0x7ffca3cf9c78 0x7ffca3cfba36 -31.78
8  0x7ffca3cf9b30
9  0x7ffca3cf9b30 121
10 0x7ffca3cf9b34 -26
11 0x7ffca3cf9b38 42.299999
12 0x7ffca3cf9b40 -31.780000
13 0x7ffca3cf9b28 0x5605f46866b0
14 0x5605f46866b0 121
15 0x5605f46866b4 -26
16 0x5605f46866b8 42.299999
17 0x5605f46866c0 -31.780000
```

Stack

0xa1f	0xc58
0xa29	0xc60
0xa2d	0xc68
0xa31	0xc70
0xa36	0xc78



0xa1f	./mem_map \0
0xa29	121 \0
0xa2d	-26 \0
0xa31	42.3 \0
0xa36	-31.78 \0

↘ argv values

0xb30	121
0xb34	-26
0xb38	42.29999
0xb40	-31.78

→ static_str