

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

```
#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;

int main( const int argc, const char* argv[] ){

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

    int iter;
    for( iter = 0; iter < argc; ++iter ){
        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]) };

    fprintf( stdout, "%p\n", &static_str );
    fprintf( stdout, "%p %u\n", &static_str.the_uint, static_str.the_uint );
    fprintf( stdout, "%p %d\n", &static_str.the_int, static_str.the_int );
    fprintf( stdout, "%p %f\n", &static_str.the_float, static_str.the_float );
    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]);

    fprintf( stdout, "%p %p\n", &dyn_str, dyn_str );
    fprintf( stdout, "%p %u\n", &dyn_str->the_uint, dyn_str->the_uint );
    fprintf( stdout, "%p %d\n", &dyn_str->the_int, dyn_str->the_int );
    fprintf( stdout, "%p %f\n", &dyn_str->the_float, dyn_str->the_float );
    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

```
0x7ffca3cf9b1c 5
0x7ffca3cf9b10 0x7ffca3cf9c58
0x7ffca3cf9c58 0x7ffca3cfba1f ./mem_map
0x7ffca3cf9c60 0x7ffca3cfba29 121
0x7ffca3cf9c68 0x7ffca3cfba2d -26
0x7ffca3cf9c70 0x7ffca3cfba31 42.3
0x7ffca3cf9c78 0x7ffca3cfba36 -31.78
0x7ffca3cf9b30
0x7ffca3cf9b30 121
0x7ffca3cf9b34 -26
0x7ffca3cf9b38 42.299999
0x7ffca3cf9b40 -31.780000
0x7ffca3cf9b28 0x5605f46866b0
0x5605f46866b0 121
0x5605f46866b4 -26
0x5605f46866b8 42.299999
0x5605f46866c0 -31.780000
```