

Computer Programming 2 Lab

2023/03/29 Andy Hung



Outline

- HW 4
- Function in struct
- Using .h files

HW4

Description

There's some rumor that students will fry sweet potato balls in front of the NCCUCS department office, which is untrue. In fact, students farm some normal potatoes. These potatoes then get fried by themselves and become potato balls, the process does not change the weight of the potato (after frying a potato weight a , you get a potato ball with weight a). The magical potatoes have magical effect on them, which is kept after frying. Now, you need to fry two different types of potato balls in the same way.



HW4

Input

The first line gives a positive integer `N``, which is the rand seed.

Output

Please print out informations if it runs to print.

HW4

Input

```
1 1
```

Output

```
1 Potato:
2   producer: Tom
3   weight: 887
4 Potato Ball:
5   producer: Tom
6   weight: 384
```

HW4

What you need

```
1  struct PotatoBall {
2      struct PotatoProducer* producer;
3      int weight;
4      int magical;
5      char effect[32];
6      void (*print)(struct PotatoBall** potatoBall);
7      void (*dtor)(struct PotatoBall** potatoBall);
8  };
9
10 struct Potato {
11     struct PotatoProducer* producer;
12     int weight;
13     int magical;
14     char effect[32];
15     void (*print)(struct Potato** potato);
16     struct PotatoBall* (*fry)(struct Potato** potato);
17 };
```

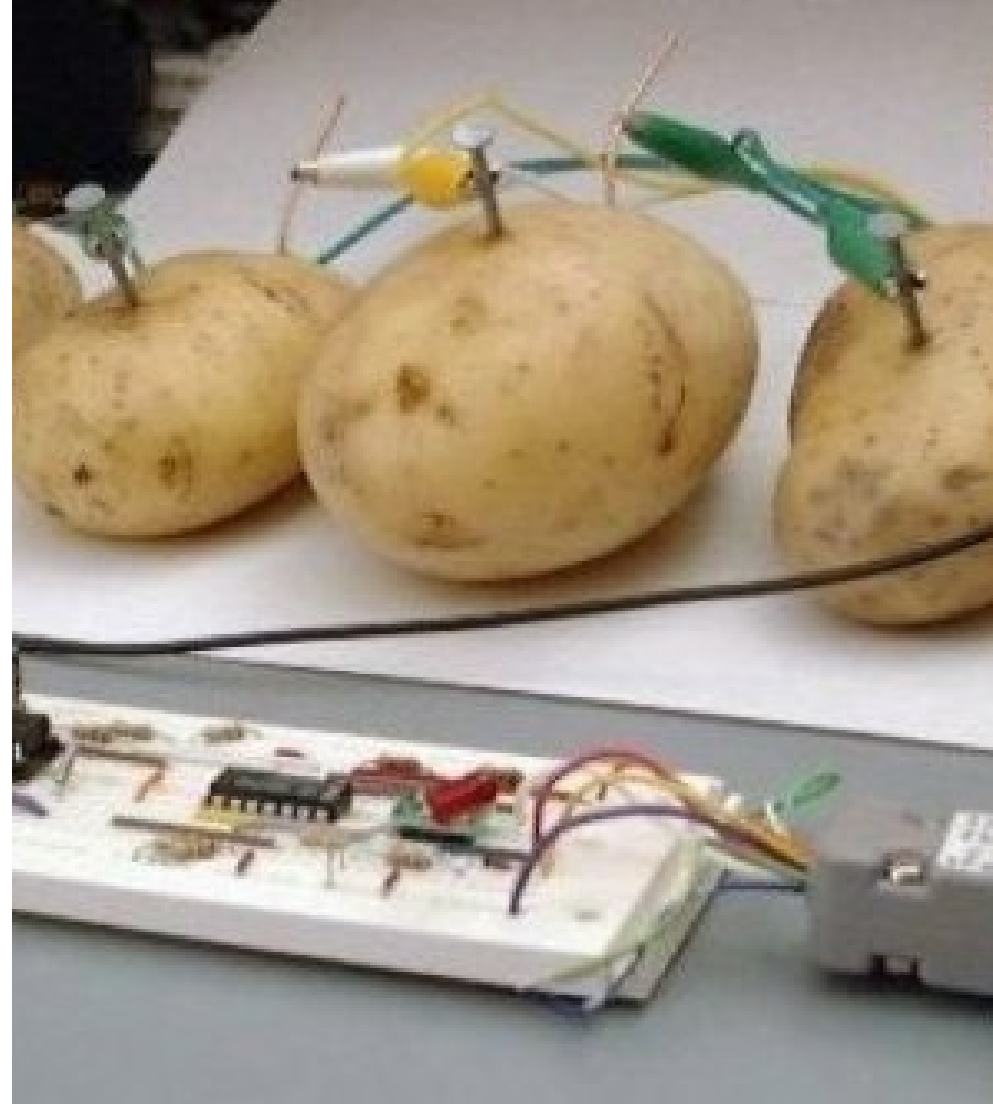
HW4

What you need

```
1  struct PotatoProducer {  
2      char name[20];  
3      int magical;  
4      struct Potato* (*produce)(struct PotatoProducer* producer);  
5  };  
6  
7  void init_potato_producer(struct PotatoProducer* producer, const char* name, int magical);
```

HW4

- Producer name may change
- Destroy itself when ``fry`` or ``dtor``
- Print detail of the potato/potato ball
- Magical effect will always be "slowness"
- Potato weight will be ``(rand() % 1000) + 1``, which is an integer between 1 and 1000



HW4

Print Sample

```
1  Potato:
2      producer: Tom
3      weight: 95
4  Magical Potato:
5      producer: Jane
6      weight: 995
7      effect: slowness
8  Potato Ball:
9      producer: Tom
10     weight: 95
11  Magical Potato Ball:
12     producer: Jane
13     weight: 995
14     effect: slowness
```

Function in struct

```
1  struct test {  
2      void (*test_function)();  
3  };  
4  
5  void this_is_test() {  
6      printf("Hello world");  
7  }  
8  
9  // ...  
10 test->test_function = this_is_test;
```

Using .h files

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include "potato.h"
```

Using .h files

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include "potato.h"
```

Using .h files

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include "potato.h"
```

- To prevent muti declaration

Using .h files

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include "potato.h"
```

- To prevent muti declaration

```
1  // some_header_file.h
2  #ifndef SOME_HEADER_FILE_H
3  #define SOME_HEADER_FILE_H
4  // your code
5  #endif
```

Thanks for listening
Any Questions?