CIS2107
Computer Systems & Low-Level Programming

Lab05. Array of Pointers to Functions

Format

- ➔ Upload `.c` file (**`ArraysofPointers.c`**) to Canvas
  - *As Always! Test on cis-linux2 server !!!!*

- ➔ Comments at top of the file:
  - Name, Date, Course
  - Homework number (Lab 5 Arrays of Pointers to Functions...)
  - Statement of problem

Recommendations

Array of Pointers to Functions

- ➔ Each of the functions MUST have the same parameters and return type, ***see following slides for an example***

Hints

- Problem 02: **Figure 6.22** mentioned in assignment is in module presentation **02_06_C Arrays.pdf**. Also, you need to use an array of pointers to functions (similarly as in **Fig. 7.28** (*slide 47 of 02_07_C Pointers.pdf*). Your array of pointers to functions should have 4 elements (pointers to functions: `minimum, maximum, average, printArray`). All of those functions should return `void`.

- Example of passing an array to a function: `void printArray(int m, int n, int arr[m][n]),` ie. pass **m** and **n** first.

- Dimensions and element values of 2-D arrays should be read from the input for both part1 and part2.

```c
// Fig. 7.28: fig07_28.c
// Demonstrating an array of pointers to functions.
#include <stdio.h>

// prototypes
void function1(int a);
void function2(int b);
void function3(int c);

int main(void)
{
   // initialize array of 3 pointers to functions that each take an
   // int argument and return void
   void (*f[3])(int) = { function1, function2, function3 };

   printf("%s", "Enter a number between 0 and 2, 3 to end: ");
   size_t choice; // variable to hold user's choice
   scanf("%u", &choice);

```

```
20      // process user's choice
21      while (choice >= 0 && choice < 3) {
22
23         // invoke function at location choice in array f and pass
24         // choice as an argument
25         (*f[choice])(choice);
26
27         printf("%s", "Enter a number between 0 and 2, 3 to end: ");
28         scanf("%u", &choice);
29      }
30
31      puts("Program execution completed.");
32   }
33
34   void function1(int a)
35   {
36      printf("You entered %d so function1 was called\n\n", a);
37   }
38
39   void function2(int b)
40   {
41      printf("You entered %d so function2 was called\n\n", b);
42   }
43
```

```
44  void function3(int c)
45  {
46      printf("You entered %d so function3 was called\n\n", c);
47  }
```

```
Enter a number between 0 and 2, 3 to end: 0
You entered 0 so function1 was called

Enter a number between 0 and 2, 3 to end: 1
You entered 1 so function2 was called

Enter a number between 0 and 2, 3 to end: 2
You entered 2 so function3 was called

Enter a number between 0 and 2, 3 to end: 3
Program execution completed.
```

Checklist

- Is my output readable?
- Could a user understand what my program is doing if they did not have the lab document in front of them?
- Does my program compile and run on the **`cis-linux2 server`**?