LAB 5 - Pointers and Structs Matthew Szymanski 05/26/2021

1. What is the difference between the following two declarations. 2 pts

   int *p[10];

   int (*p) [10];

   The difference between the following two declarations is that the first is an int array of[10] pointers, and the second is a pointer p to 10 integers.

2. Please explain the following two declarations. 3 pts

   int (*p)(char (*a)[]); This declaration is a function pointer to an array and returns an int.

   int *p(char (*a)[]); This declaration is a pointer to an array as a function and returns an int pointer.

```
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ gcc -o FunctionPointer FunctionPointer.c
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ gcc -o FunctionPointer FunctionPointer.c
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ ./FunctionPointer
Enter a number: 10
Enter another number: 5
Enter an operation (0=add, 1=subtract, 2=multiply, 3 = Divide ): 0
The answer is : 15
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ nano FunctionPointer.c
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ gcc -o FunctionPointer FunctionPointer.c
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ ./FunctionPointer
Enter a number: 10
Enter another number: 10
Enter an operation (0=add, 1=subtract, 2=multiply, 3 = Divide ): 1
The answer is : 0
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ ./FunctionPointer
Enter a number: 100
Enter another number: 10
Enter an operation (0=add, 1=subtract, 2=multiply, 3 = Divide ): 2
The answer is : 1000
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ ./FunctionPointer
Enter a number: 8
Enter another number: 9
Enter an operation (0=add, 1=subtract, 2=multiply, 3 = Divide ): 3
The answer is : 0
mszymanski@DESKTOP-RD9BRN8:~/Lab5$ ./FunctionPointer
Enter a number: 16
Enter another number: 2
Enter an operation (0=add, 1=subtract, 2=multiply, 3 = Divide ): 3
The answer is : 8
mszymanski@DESKTOP-RD9BRN8:~/Lab5$
```

3. Take a look at the following code snippet. Here **pFcn** is a pointer to a function that takes two integer arguments and returns an integer. To make the different cases in switch statement work, write a few functions such as 'Add' , 'Subtract', 'Multiply', 'Divide' that take two integers as arguments and return an integer. Print the value of **pFcn(X,Y)** for all these cases.
   4 pts

   Submit as a complete working code named as **FunctionPointer.c.**

```
#include <stdio.h>
int(*pFcn)(int, int);
int main(){
int X, Y, operation;
printf("Enter a number: ");
scanf(" %d", &X);
printf("Enter another number: ");
scanf(" %d", &Y);
printf("Enter an operation (0=add, 1=subtract, 2=multiply, 3
= Divide ): ");
scanf(" %d", &operation);
switch (operation) {
     //  case 0: pFcn = Add; break;
    //   case 1: pFcn = Subtract; break;
    //   case 2: pFcn = Multiply; break;
    //   case 3: pFcn = Divide; break;
       }
    //   printf("The answer is : %d\n", pFcn(X,Y));
return 0;
}
```

4. Take a look at the following code snippet: 2 pts

```
struct Person{       char
     name[BUFSIZ];
     char ssn[BUFSIZ];
     int age;       float
     height;       float
     weight;
};  struct Person
p1;

strcpy(p1.name, "Alfred Morino");
strcpy(p1.ssn, "496-50-2260");
p1.age = 50;  p1.height = 170.5;
p1.weight = 70.5;


struct Person *ptr = &p1;
```

What will be printed by the following expressions? Provide the screenshot.

```
printf("Name = %s\nSSN = %s\nAge = %d\nHeight(cm) = %g\nWeight(kg) = %g\n", p1.name, p1.ssn,p1.age, p1.height, p1.weight);

printf("Name = %s\nSSN = %s\nAge = %d\nHeight(cm) = %g\nWeight(kg) = %g\n", ptr->name, ptr->ssn, ptr->age, ptr->height, ptr->weight);

printf("Name = %s\nSSN = %s\nAge = %d\nHeight(cm) = %g\nWeight(kg) = %g\n", (*ptr).name, (*ptr).ssn, (*ptr).age, (*ptr).height, (*ptr).weight);

printf("Name = %s\nSSN = %s\nAge = %d\nHeight(cm) = %g\nWeight(kg) = %g\n", (&p1)->name, (&p1)->ssn, (&p1)->age, (&p1)->height, (&p1)->weight);
```

5. Take a look at the attached file "**structConversion.c**". Use the following struct template named "**Person**" in the program. Modify existing **printData** and **readData** functions as follows. (9 pts in total)

void printData(struct Person x);	3 pts
struct Person readData();	3 pts
Replace **gets** with **fgets**.	3 pts

You can use any additional helper functions. Submit the complete file as "**structConversionLab5.c**" file.

Submission:

A zip file containing:

- Your Complete C code named **FunctionPointer.c, structConversionLab5.c** and a pdf file named **PointersAndStructLab5.pdf** containing the answers to questions 1, 2 with output capture for C code for question 4.

Name your zip file with your last name first letter of your first name Lab5.zip (ex: **yasminsLab5.zip**)

Submission deadline is: 11:59 pm, Friday, May 28. No late submissions will be considered.