

Louden Demers
Professor Yasmin
CSCD 240-040
28 May 2021

LAB 5 - Pointers and Structs

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

```
int *p[10];  
int (*p) [10];
```

Ans: The first declaration is an array of ten **int** pointers while the second declaration is a pointer to an array of ten integers.

2. Please explain the following two declarations. 3 pts

```
int (*p)(char (*a)[]);  
int *p(char (*a)[]);
```

Ans: The first declaration is a function pointer **p** that takes a pointer **a** to a **char** array, this returns an **int**. The second declaration is just a regular function **p**, which takes a pointer **a** to a **char** array and returns an **int** pointer. The big difference is the parentheses around ***p**.

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;

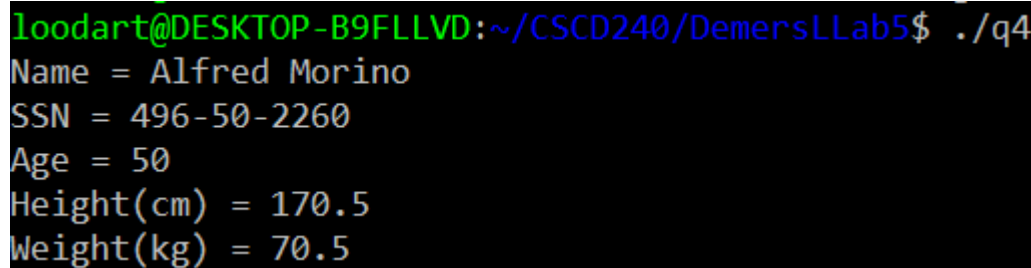
strcpy(p1.name, "Alfred Morino");
strcpy(p1.ssn, "496-50-2260");
p1.age = 50; p1.height = 170.5;
p1.weight = 70.5;

struct Person *ptr = &p1;

```

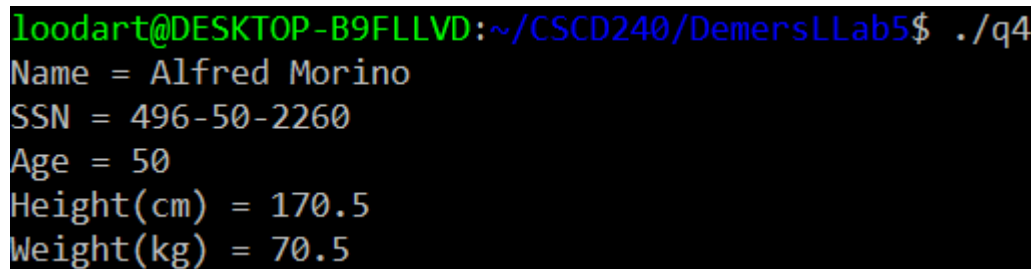
What will be printed by the following expressions? Provide the screenshot.

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



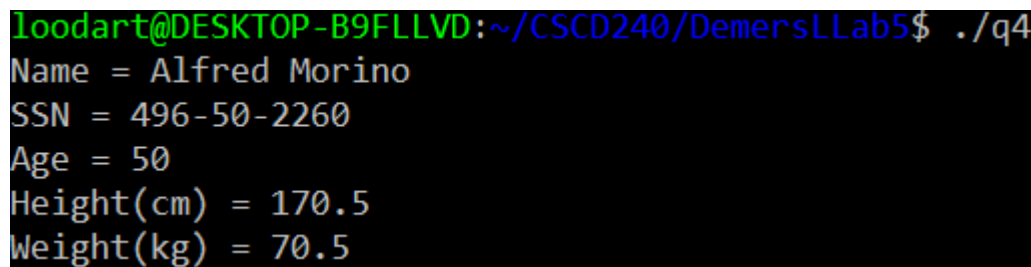
```
loodart@DESKTOP-B9FLLVD:~/CSCD240/DemersLLab5$ ./q4
Name = Alfred Morino
SSN = 496-50-2260
Age = 50
Height(cm) = 170.5
Weight(kg) = 70.5
```

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



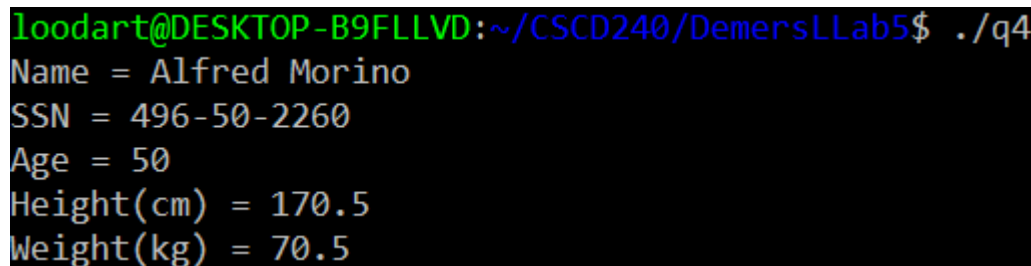
```
loodart@DESKTOP-B9FLLVD:~/CSCD240/DemersLLab5$ ./q4
Name = Alfred Morino
SSN = 496-50-2260
Age = 50
Height(cm) = 170.5
Weight(kg) = 70.5
```

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



```
loodart@DESKTOP-B9FLLVD:~/CSCD240/DemersLLab5$ ./q4
Name = Alfred Morino
SSN = 496-50-2260
Age = 50
Height(cm) = 170.5
Weight(kg) = 70.5
```

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



```
loodart@DESKTOP-B9FLLVD:~/CSCD240/DemersLLab5$ ./q4
Name = Alfred Morino
SSN = 496-50-2260
Age = 50
Height(cm) = 170.5
Weight(kg) = 70.5
```

All outputs ended up being identical.

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.