HW #7: Data Types

Due dates:

Part I: Monday Feb 29th, at the beginning of the class. Make sure to write your name and msunetid on your paper.

Part II: Sunday Feb 28th, 11:59 pm through Handin (https://secure.cse.msu.edu/handin)

Part I: Comprehension Questions

1. Run the following program. What is the output? (1.5 pts)

```
#include<stdio.h>
int main(void) {
        float f1 = 1.00001, f2 = 1.000001, f3 = 1.0000001;
        printf("%.10f, %.10f, %.10f\n", f1, f2, f3);

        return 0;
}
```

```
1.0000100136, 1.0000009537, 1.0000001192
```

The float data type does not precisely store the decimal values.
If f1, f2, f3 were declared as doubles instead the answer would be more accurate:

```
1.0000100000, 1.0000010000, 1.0000001000
```

2. What is the output of the following programs if the user enters 3 6.7 for both: (4pts)

<u>Program A:</u>

```
#include <stdio.h>
int main(void) {
     int i1, i2;
     printf(“Enter an int and a float\n”);
     scanf(“%d %d”, &i1, &i2);
     printf(“%d %d %f %f\n”, i1, i2, i1, i2);
     return 0;
}
```

```
3 6 0.000000 0.000000
```

The program reads the first integer as 3. It reads the second as 6 (it stops reading when it hits the decimal point since this cannot be part of an integer).
When displaying them back, they are written correctly as integers, but incorrectly as floats.

Program B:

```
#include <stdio.h>
int main(void) {
        float f1, f2;
        printf(“Enter an int and a float\n”);
        scanf(“%f %f”, &f1, &f2);
        printf(“%d %d %f %f\n”, f1, f2, f1, f2);

        return 0;
}
```

```
-1264875600 2 3.000000 6.700000
```

The program reads both values correctly as decimals and outputs them correctly as floats. When displaying them back while interpreting them as integer, they are not interpreted correctly.

3. Rewrite the following program using macro definitions for the constant and a new type definition Money to represent money amounts. (3 pts)

```
#include <stdio.h>
int main(void) {
        float usdAmount, cadAmount;
        printf(“Enter the USD amount\n”);
        scanf(“%f”, &usdAmount);
        cadAmount = usdAmount*1.38;
        printf(“The amount in CAD is %f\n”, cadAmount);

        return 0;
}
```

```
#include <stdio.h>
#define USD_TO_CAD 1.38
typedef float Money;

int main(void) {
      Money usdAmount, cadAmount;
      printf("Enter the USD amount\n");
      scanf("%f", &usdAmount);
      cadAmount = usdAmount*USD_TO_CAD;
      printf("The amount in CAD is %f\n", cadAmount);
```

```
    return 0;
}
```

4. What are the values of x1, x2 and x3? (1.5 pts)

```
int x1 = sizeof('x');
int x2 = sizeof(x1);
int x[5];
int x3 = sizeof(x);
```

4 4 20

By default, 'x' is considered an integer not a char. So x1 is the number of bytes in an integer.
X2 is the number of bytes of an integer.
X3 is the number of bytes in the array which is size of 5 integers,

Part II: Lab Assignment

Getting started

Change into the cse220 directory
Create a new directory called lab07
Change into the new directory
Implement the program below in your lab07 directory

Integer Bounds

Write a program (largest.c) that finds the largest short. Your program should define a short int and initialize it to 0. Then it should enter an infinite loop that keeps on incrementing the short by 1 and displaying the result.
When the program exceeds the limit that can be stored in a short, it will fail and start printing the wrong answers.
To be able to view the results, modify your loop so that it waits for any user input (one char is enough) after every 1000 iterations.
Kill the program after the overflow is reached.

Compile your program into the executable largest and run your program. What is the largest short?

```
#include<stdio.h>

int main(void) {
    short x;
    char y;
    for (x = (short) 0; ; x++) {
        printf("%d\n", x);
        if (x % 1000 == 0) {
            printf(“Enter any character to continue….\n”);
            scanf("%c", &y);
        }
    }

    return 0;
}
```

ASCII Translator

Write a program (asciitrans.c) that gives the user the following options:

+ Enter A to output the ASCII code for a given char
+ Enter B to output the char corresponding to the given ASCII code

If the user enters A, the program should ask to enter one char and outputs the corresponding ASCII code. For example, if the user enters b, the program should output 98.
If the user enters B, the program should ask to enter an integer, and output the char with the given ASCII code. For example, if the user enters 98, the program should output b.
If the user enters a different option, the program should output an error message that the option is not valid.

Modify the program so that it asks the user the following:

Do you want to try another conversion (y/n)?

If the user types y, the options are displayed again and the program executes the conversion according the option selection. If the user types anything other than y, the program should exit.

Compile your program into an executable called asciitrans.

```
#include<stdio.h>

int main() {
	int input;
	char option, repeat;

	do {
	   //Show options
	   printf("Select one of the following options:\n");
	   printf("\t- Enter A to output the ASCII code for a given char\n");
	   printf("\t- Enter B to output the char corresponding to the given ASCII code\n");

	   /* Read selected option.
	         Add a space before %c so all whitepace including newline
	         (when user hits 'Enter') is consumed */
	   scanf(" %c", &option);
	   if (option == 'A') {
			printf("Enter a char:\n");
			scanf(" %c", &input);
			printf("The ASCII code for %c is: %d\n", input, input);
	   } else if (option == 'B') {
			printf("Enter an ASCII code:\n");
			scanf("%d", &input);
			printf("The corresponding char for %d is: %c\n", input, input);
	   } else {
			printf("This is not a valid option\n");
	   }

	   printf("Would you like to try another conversion? (y/n) \n");
	   scanf(" %c", &repeat);
	} while (repeat == 'y' || repeat == 'Y');

	return 0;

}
```

Handin

Submit through the handin system your C code and the executables generated.

The “handin” system has options to allow you to review your files online and to download them. You Should always verify that you submitted the correct files and they were received by the handin system. You can submit files as many times as you like for a particular assignment. Handin will only keep the last version of each file. Remember to submit your files prior to the deadline as you won’t be able to use handin if the deadline has passed.