CSCI 201 – Computer Science 1

Lab Assignment 11. Working with addresses, pointers etc.
Due on Tuesday April 23
You will have to use the fraction class that was created for Lab 10.

Objectives: *Become familiar with the use of pointer variables and memory addresses in C++. Understand what happens when arrays are passed as parameters in C++. Understand how passing a parameter using a pointer is different from passing in a variable reference. Learn how to dynamically allocate memory on the heap in C++.*

Question 1. Create a program with the variable declarations shown below. Following the example in `pointers/angletest2.cpp`, use the & operator, and add statements that print out the addresses of all the variables. Make sure that the fraction files (`fraction.h and fraction.cpp`) from Lab 10 have been copied to the directory. They will have to be linked to this program during compilation.

```
int i1; int *ip1; int i2; fraction farray[5]; fraction f1(2,5);
float x1; double x2;  int i3; int A[6];
 float *fp1; fraction *ffp1; double *dp1;
```

Question 2. Following the examples we worked in class, create a "blank memory map" for the above program. Recall that the addresses will be known only after the program is executed. Compile (with `fraction.o`) and test the program. Draw the complete memory map. Explain as best as you can, why the variables were placed in those locations.

Question 3. Add statements to your program (from Question 1) that do each of the following:
1. store the address of `i1` in `ip1`.
2. store the value 12 in `i2` and print it via `cout`.
3. (changing `i2` via address of `i1`) compute the difference between the addresses of `i2` and `i1`. By using this difference and the address of `i1` (stored in `ip1`), write an expression for computing the address of `i2`. Dereference this expression to access `i2`, and change its value to 23.
4. add a statement to print `i2` after the change.

Compile (with `fraction.o`) and test the program **in a script session.** Draw the complete memory map. Use the pointer arithmetic examples from `pointers/angletest4.cpp`.

Question 4: Modify the program to add the functions,
`void exchange (int &pi, int &pj)`, and `void exchange (int *pi, int *pj)`. Both functions exchange the values of the two integer parameters. Use `sum_and_diff` and `sum_and_diff2` in `pointers/angletest7.cpp` as examples. In main, add statements to test the functions by exchanging the values of `i1` and `i2`. Compile and run in script session.

Question 5: Using the code in `pointers/angletest8.cpp` as an example, use the "new" command to make space for a fraction whose address is to be kept in `ffp1`. Add statements to print the fraction and the contents of `ffp1`. Why does this address look somewhat different from the other addresses?

Question 6: Set the numerator and denominator to 17 and 13 respectively by applying setNumer-

ator and setDenominator to the fraction pointed to by `ffp1`. (These statements will use '*' and '.' to change the data members; see `pointers/angletest8.cpp`.) Print the fraction before and after the change. Repeat the exercise using the `->` operator (instead of * and .) to change the values to 27 and 23 respectively. Print the contents again. Run the program in a script session.

What to submit:

1. Answers to Question 2(memory map + explanations) and Question 5.
2. Script files for Question 3, Question 4, and Question 6.
3. Final source code (.cpp file).