Handout 1
C++ Programming
Deadline October 5

Exercise 1
Given the following assignment of variables to values:

x	power	y	item	MIN	DAY	num	MAX	Sens
-5	1024	7	1.5	-12.0	'M'	12	1024	12

Fill in the result values of the conditions in the table below:

Condition	Result
(x>y) && !y	0
(item>MIN)||(DAY!='M')	1
((num*128)<power)&&y	0
(!(power!=MAX))&&(Sens==num)	1
((y+x)<num)||(DAY=='M')	1
(Sens*(!y))!=0	0
(!x||y)&&(!y||x)	1

Exercise 2
Write a program in C++ that performs the following tasks:

1. Read three integer values using **cin.**
2. Determine the maximum of the three values entered by the user.
3. Print the maximum of this three values using **cout.**

```
#include <iostream>

int main() {
    int numbers[3] = {};
    // array was not necessary here, I could just use 3 integer variables

    std::cout << ">> Please enter a number (integer)  ";
    std::cin >> numbers[0];
    std::cout << ">> Please enter a 2nd number ";
    std::cin >> numbers[1];
    std::cout << ">> Please enter a 3rd number ";
    std::cin >> numbers[2];

    int max = numbers[0];

    if (numbers[0] > numbers[1] && numbers[0] > numbers[2])
        max = numbers[0];
    else if (numbers[1] > numbers[0] && numbers[1] > numbers[2])
        max = numbers[1];
    else if (numbers[2] > numbers[0] && numbers[2] > numbers[1])
        max = numbers[2];
    else
        std::cout << "Something went wrong!!" << std::endl;

    // another possible solution:
    /*for (int number: numbers) {
        if (number > max)
```

```
            max = number;
    }*/

    std::cout << ">> The max is: " << max << std::endl;
    return 0;
}
```

Exercise 3
Write a program that asks the user to type numbers. After each entry, the program should report the cumulative sum of the entries. The program should terminate when the user enters 0.

```
#include <iostream>
#include <list>

using namespace std;

int main() {

    double sum = 0;
    double newNum;
    while (true) {
        std::cout << ">> Please enter a number  ";
        std::cin >> newNum;
        // if you enter something else then a number newNum will be 0

        if (newNum == 0)
            return 0;

        sum += newNum;
        std::cout << "The sum of the entries is: " << sum << std::endl;
    }

    // Or another alternative:
    /*list<double> numbers = {};
    double temp;

    while (true) {
        std::cout << ">> Please enter a number  ";
        std::cin >> temp;

        if (!temp)
            return 0;

        numbers.push_back(temp);
        double sum = 0;

        for (auto number : numbers) {
            sum += number;
        }
        std::cout << "The sum of the entries is: " << sum << std::endl;
    }*/
}
```

Exercise 4
Create a program to determine the GCD (Greatest Common Divisor) of two integers x and y using a ‘while loop’.

Formal description of the Euclidean algorithm

- **Input** Two positive integers, a and b.
- **Output** The greatest common divisor, g, of a and b.

- **Internal computation**
    1. If a<b, exchange a and b.
    2. Divide a by b and get the remainder, r.
       If r=0, report b as the GCD of a and b.
    3. Replace a by b and replace b by r. Return to the previous step.

```cpp
#include <iostream>
using namespace std;

class GCD {
public:
    double getEntry() {
        int entry;
        std::cout << ">> Please enter a number  ";
        std::cin >> entry;
        while (entry < 0) {
            std::cout << "Invalid entry. Try again!" << std::endl;
            std::cin >> entry;
        }
        return entry;
    }

    double getGCD(int a, int b) {
        while(true){
            if (a < b) {
                int temp = a;
                a = b;
                b = temp;
            }

            int r = a % b;
            if (r == 0)
                return b;

            a = b;
            b = r;
        }

        // alternative using recursion
        /*if (a < b) {
            int temp = a;
            a = b;
            b = temp;
        }

        int r = a % b;
        if (r == 0)
            return b;

        return getGCD(b, r);*/
    }
};

int main() {
    // object orientation wasn't necessary here, I just wanted to try it
    GCD gcd_obj;
    int a = gcd_obj.getEntry();
    int b = gcd_obj.getEntry();

    std::cout << "The gcd of " << a << " and " << b << " is " <<
gcd_obj.getGCD(a, b) << std::endl;
}
```