# CSCE 240 - Programming Assignment One

**Due:** 11:59pm on Wednesday, February 5<sup>th</sup>

<u>Program Purpose</u> - Convert numeric values expressed in bases between 2 and 9, inclusive, to the base 10 equivalents of those values. The program will output the base 10 values, the number of values input, and the largest value input.

#### Overview

In a positional numeration system with base b, each position represents a power of b.

In the decimal system, which is a positional numeration system with base 10, each position represents a power of 10. So, for example, the decimal number 4809 is  $4 \cdot 10^3 + 8 \cdot 10^2 + 0 \cdot 10^1 + 9 \cdot 10^0$ .

Similarly, in binary, which is a positional numeration system with base 2, each position represents a power of 2. For example, the binary number 11001 is  $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$ . So, the value 11001 in binary would be expressed as 25 in decimal.

For more detail and examples about positional numeration systems, read over the "Numeration Systems" PDF.

#### <u>Program Details</u>

For each numeric value, the user will enter two integers from the standard input device (use cin). The first integer is a numeric value, and the second integer is the base in which that number is expressed.

Your program should check each base input to ensure that it is between 2 and 9, inclusive. If the base is outside of this accepted range, your program should output "Base Not Accepted" to the standard output device (using cout), and the program should end.

Your program should also check each numeric value input to ensure that it only contains valid digits for the given base. The number of unique digits in a positional numeration system is equal to the base. In base 10, for example, the valid digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The binary numeration system, on the other hand, only uses the unique digits 0 and 1. If the numeric value input uses invalid digits, your program should output "Invalid Digit(s) in Number" to the standard output device (using cout), and the program should end.

If the input numeric value and base are valid, the program should output the number expressed in decimal (base 10) to the standard output device (using cout)

Following each numeric value entered, the user will input the character 'y' if another numeric value will be input, or 'n' if no more numeric values will be input. When the user enters 'n' the program should output the number of values input and the largest value input (expressed in decimal) in the following format:

Of the *num* values input, the *which* value entered (*decimalvalue*) was the largest Where *num* is the number of values input, *which* is the position of that value (e.g. "1st", "2nd", "3rd", etc), and *decimal* is the base 10 equivalent of that value.

#### Example Input/Output Pairs

**Example Input**1101 1 n

Example Output
Base Not Accepted

1030 3 n	Invalid Digit(s) in Number
Example Input	Example Output
1101 2 y	13
1101 3 y	37
320 4 y	56
43 5 v	23

Of the 6 values input, the 3rd value entered (56) was the largest

### Input Assumptions

11000 2 n

Example Input

Assume that the user input will match the expected format.

Example Output

24

The user will input integer values, as described above, for the numeric value and base. If the user enters non-integer values, it is ok (expected) for your program to crash.

The user will input a single character (y or n as described above) to signify whether or not another numeric value will be input. If the user enters more than one character, it is ok (expected) for your program to crash.

## **Additional Specifications**

- All output should be directed to the standard output device using cout.
- All input should be accepted from the standard input device using cin.
- The final output in every execution of your program must be an endl.
- Do not prompt for input.
- All of your source code for the program must be contained in a single file named program1.cc
- Submit your program1.cc file to the assignment in Blackboard.
- The only header file that can be included in your code is iostream. Programs that include other headers will not be eligible for correctness points.
- Programs must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).
- Be sure to review the program expectations section of the course syllabus.

### **Initial Testing**

Initial tests for the functions are attached to the assignment in Blackboard. A makefile has been included to run your functions with the sample tests. To use the makefile, ensure that your program1.cc file and all of the files attached to the assignment are in the same directory. Your program will be graded using this same method with additional tests.

The commands to run the sample tests are given below:

make test1 make test2 make test3 make test4 make test5 make test6

You are strongly encouraged to create additional, more rigorous tests.

The six included tests and one new test will be used to grade your program.

## <u>Grade Breakdown</u>

Style: 1 point Documentation: 1 point

Clean compile/link of program1.cc: 1 point

Runs correctly with instructor's test input 1: 1 point Runs correctly with instructor's test input 2: 1 point Runs correctly with instructor's test input 3: 1 point Runs correctly with instructor's test input 4: 1 point Runs correctly with instructor's test input 5: 1 point Runs correctly with instructor's test input 6: 1 point Runs correctly with instructor's test input 6: 1 point Runs correctly with instructor's test input 7: 1 point

The penalty for late assignment submissions is 10% per day up to three days after the assignment due date. No assignment submissions will be accepted more than 3 days after the due date.