

CS 280
Spring 2023
Recitation Assignment 6
March 2nd, 2023

Due Date: Monday, March 6, 2023, 23:59
Total Points: 8

Write a C++ recursive function that performs conversion from decimal to any number of base n , where n is in the range 2-10 or $n = 16$ (hexadecimal). The algorithm for this conversion is to repeatedly divide the decimal number by n , until it is 0. Each division produces a remainder, which becomes a digit in the converted number. For example, if we want to convert the decimal 14 number to a number of base 3, we would apply the following steps:

$14/3 = 4$	remainder 2
$4/3 = 1$	remainder 1
$1/3 = 0$	remainder 1

Thus, the result is $(112)_3$. The only problem with this algorithm is that the first division generates the low-order digit, the next division generates the second-order digit, and so on, until the last division produces the high order digit. Thus, if we output the digits as they are generated, they will be in reverse order. Thus, you should use recursion to reverse the order of output.

Implement a recursive C++ function that performs the conversion from decimal to any number of base n that has the following header definition:

```
string DecToBaseN(int num, int base);
```

Where, *num* is the passed decimal number and *base* is the base of the number to be converted to in the range 2-10 or 16. The function returns the converted number as a string of digits. In the case of base 16, the function returns the converted number as a string of digits and letters (A-F which represent the values 10-15, respectively). For any other illegal base n , the function displays the message “Invalid Base: n ” and returns a null string.

Vocareum Automatic Grading

- Implement the `DecToBaseN()` function in a file, called “DecToBaseN.cpp” and upload it to Vocareum.
- You can implement any driver program for testing your implementation. However, a driver program is provided on Vocareum for testing the implementation, called “progRA6.cpp”. The “progRA6.cpp” will be propagated to your work directory. The program asks the user to enter a binary number, calls the `DecToBaseN()` function, then prints out the result of conversion to the required base.
- The testing is based on the correct results returned by the call to your recursive function `DecToBaseN()`, and the satisfaction of using recursive approach for the solution of the problem, rather than an iterative one. Test case files will be used instead of the input from

the keyboard on Vocareum. There are 7 testing files associated with RA 6. Vocareum automatic grading will be based on these testing files. You may use them to check and test your implementation. These are available in a compressed archive “RA 6 Test Cases.zip” on Canvas assignment. The testing case of each file is defined in the Grading table below.

- “prog.cpp” is available with the other assignment material on Canvas.

Submission Guidelines

- Please upload your implementation of the function in a file to Vocareum as a “DecToBaseN.cpp”. The file should include the implementation of the function `DecToBaseN()`.
- **Submissions after the due date are accepted with a fixed penalty of 25% from the student’s score. No submission is accepted after Wednesday 11:59 pm, March 8, 2022.**

Grading Table

Item		Points
Function Implementation	Case 1: Base 2	1
	Case 2: Base 5	1
	Case 3: Base 10	1
	Case 4: Base 16	1
	Case 5: Base 12	1
	Case 6: Base 0	1
	Case 7: Base -1	1
	Compiles Successfully	1
Total		8