

Quinn Roemer

CISP - 440

Assignment 1

9/13/2018

Part 0 - Number Bases Implementation

Description:

The goal for this assignment was to implement code that was capable of performing conversions between any base from 2-36. To do this we were to take the number that was entered and convert it to an internal binary representation. After this, we used the MOD DIV algorithm to successfully convert to any desired base. In addition to this, the program needed to be capable of performing addition in base. That is, add 2 numbers of some arbitrary base with the sum being in the same base of the original numbers.

Please note: I wrote the entirety of the following code from scratch and based it on what Professor Ross had given us to start with.

Source Code:

//Written by Quinn Roemer. Based on code by Professor Ross.

```
#include <iostream>
#include <string>
#include <math.h>
#include <stack>
#include <algorithm>

using namespace std;

//Converts an entered number into an internal representation.
long int atoi(string, int);

//Converts an internal rep number into any base from 2-36.
string itoa(long int, int);

//Perform addition in base.
string baseAddition(string, string, int);

//Convert from encoded to decoded DNA sequence.
string dnaCodonA(string);

//Convert from decoded to encoded DNA sequence.
string aDnaCodon(string);

//Global Ref Array for determining notation.
char refArray[] = {
'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J','K','L','M','N',
',','O','P','Q','R','S','T','U','V','W','X','Y','Z' };
```

```

int main()
{
    char answer;

    string base;
    string addNum1;
    string addNum2;
    string num;
    string original;

    bool enteredDNA = false;
    bool outputDNA = false;

    int enteredBase;
    int conversionBase;
    long int result;

    cout << "Menu" << endl;
    cout << "===== " << endl;
    cout << "1. Perform Base Conversion" << endl;
    cout << "2. Perform Base Addition" << endl;
    cout << "3. Quit" << endl;
    cin >> answer;

    while (answer != '1' && answer != '2' && answer != '3')
    {
        cout << "\nInvalid input. Please try again." << endl;
        cin >> answer;
    }
    switch (answer)
    {
    case '1':
        cout << "\nPlease enter the base of the number: (Enter DNA for DNA conversion)" <<
endl;
        cin >> base;

        transform(base.begin(), base.end(), base.begin(), ::toupper);

        if (base == "DNA")
        {
            cout << "\nPlease enter the sequence:" << endl;
            cin >> base;

            transform(base.begin(), base.end(), base.begin(), ::toupper);

            original = base;

            num = dnaCodonA(base);

```

```

        enteredBase = 4;
        enteredDNA = true;
    }
    else
    {
        enteredBase = stoi(base);

        cout << "\nPlease enter the sequence:" << endl;
        cin >> num;

        original = num;
    }

    cout << "\nPlease enter the base for the conversion: (Enter DNA for DNA conversion)"
<< endl;
    cin >> base;

    transform(base.begin(), base.end(), base.begin(), ::toupper);

    if (base == "DNA")
    {
        result = atoi(num, enteredBase);
        num = itoa(result, 4);

        num = aDnaCodon(num);

        conversionBase = 4;
        outputDNA = true;
    }
    else
    {
        conversionBase = stoi(base);
        result = atoi(num, enteredBase);
        num = itoa(result, conversionBase);
    }

    if (enteredDNA == true && outputDNA == true)
    {
        cout << "\n" << original << " in base DNA" << " equals " << num << " in base
DNA" << endl;
        enteredDNA = false;
        outputDNA = false;
    }
    else if (outputDNA == true)
    {
        cout << "\n" << original << " in base " << enteredBase << " equals " << num <<
" in base DNA" << endl;
        outputDNA = false;
    }
    else if (enteredDNA == true)

```

```

        {
            cout << "\n" << original << " in base DNA" << " equals " << num << " in base "
<< conversionBase << endl;
            enteredDNA = false;
        }
        else
        {
            cout << "\n" << original << " in base " << enteredBase << " equals " << num <<
" in base " << conversionBase << endl;
        }
        break;

    case '2':
        cout << "\nPlease enter the base for the addition:" << endl;
        cin >> enteredBase;

        cout << "\nPlease enter the first number:" << endl;
        cin >> addNum1;

        cout << "\nPlease enter the second number:" << endl;
        cin >> addNum2;

        num = baseAddition(addNum1, addNum2, enteredBase);

        cout << "\n" << addNum1 << " + " << addNum2 << " = " << num << " in base " <<
enteredBase << endl;
        break;

    case '3':
        break;
    }
}
long int atoi(string num, int enteredBase)
{
    long int total = 0;
    int length = num.length();
    int digit;
    int exponent = 0;
    char character;

    //Converting the entered number to its internal representation.
    for (int count = length - 1; count >= 0; count--)
    {
        character = num[count];

        //Finding number in ref array.
        for (int index = 0; index < 36; index++)
        {
            if (refArray[index] == character)
            {

```

```

        digit = index;
        break;
    }
}

//Adding totals for internal representation.
total = total + (digit * (pow(enteredBase, exponent)));
exponent++;
}

//Returning the result.
return total;
}

string itoa(long int num, int conversionBase)
{
    string result;
    int remainder = 0;
    int rDivide;
    int size;

    char digit;

    //Stack for reversing result.
    stack<int> rStore;

    //Using MOD DIV to convert to any given base.
    while (num != 0)
    {
        rDivide = num / conversionBase;
        remainder = num % conversionBase;
        num = rDivide;
        rStore.push(remainder);
    }

    size = rStore.size();

    //Converting the stack to a string.
    for (int count = 0; count < size; count++)
    {
        remainder = rStore.top();
        digit = refArray[remainder];
        result += digit;
        rStore.pop();
    }

    //Returning the result.
    return result;;
}

string baseAddition(string num1, string num2, int base)
{

```

```

int carry = 0;
int total, size;
int place1, place2;
int difference = 0;

stack<int> answer;
string result;
char digit;

//Equalizing number length.
if (num1.size() != num2.size())
{
    if (num1.size() > num2.size())
    {
        difference = num1.size() - num2.size();

        for (int count = 0; count < difference; count++)
        {
            num2.insert(0, "0");
        }
    }
    else
    {
        difference = num2.size() - num1.size();

        for (int count = 0; count < difference; count++)
        {
            num1.insert(0, "0");
        }
    }
}

//Performing the addition.
for (int count = num1.size() - 1; count >= 0; count--)
{
    for (int index = 0; index < 36; index++)
    {
        if (refArray[index] == num1[count])
        {
            place1 = index;
            break;
        }
    }
    for (int index = 0; index < 36; index++)
    {
        if (refArray[index] == num2[count])
        {
            place2 = index;
            break;
        }
    }
}

```

```

    }
    total = place1 + place2 + carry;

    if (total - base >= 0)
    {
        total = total - base;
        carry = 1;
    }
    else
    {
        carry = 0;
    }
    answer.push(total);

    if (carry > 0 && count == 0)
    {
        answer.push(carry);
    }
}

size = answer.size();

//Converting the stack into a string.
for (int count = 0; count < size; count++)
{
    carry = answer.top();
    digit = refArray[carry];
    result += digit;
    answer.pop();
}
return result;
}
string dnaCodonA(string codon)
{
    int size = codon.size();
    string result;
    char digit;
    int number;

    stack<int> answer;

    //Performing operation.
    for (int count = codon.size() - 1; count >= 0; count--)
    {
        switch (codon[count])
        {
            case 'C':
                answer.push(0);
                break;

```



```

        case 'T':
            answer.push(1);
            break;

        case 'A':
            answer.push(2);
            break;

        case 'G':
            answer.push(3);
            break;
    }
}

size = answer.size();

//Converting the stack to a string.
for (int count = 0; count < size; count++)
{
    number = answer.top();
    digit = refArray[number];
    result += digit;
    answer.pop();
}

return result;
}

string aDnaCodon(string codon)
{
    int size = codon.size();
    string result;
    char digit;

    stack<char> answer;

    //Performing operation.
    for (int count = codon.size() - 1; count >= 0; count--)
    {
        switch (codon[count])
        {
            case '0':
                answer.push('C');
                break;

            case '1':
                answer.push('T');
                break;

            case '2':

```

```

        answer.push('A');
        break;

    case '3':
        answer.push('G');
        break;
    }
}
//Converting the stack to a string.
for (int count = 0; count < size; count++)
{
    digit = answer.top();

    result += digit;

    answer.pop();
}
return result;
}

```

Output:

```

C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
2

Please enter the sequence:
1010

Please enter the base for the conversion: (Enter DNA for DNA conversion)
10

1010 in base 2 equals 10 in base 10
Press any key to continue . . .

```

(1 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
16

Please enter the sequence:
AEFF2

Please enter the base for the conversion: (Enter DNA for DNA conversion)
8

AEFF2 in base 16 equals 2567762 in base 8
Press any key to continue . . .
```

(2 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
9

Please enter the sequence:
45346

Please enter the base for the conversion: (Enter DNA for DNA conversion)
36

45346 in base 9 equals NA6 in base 36
Press any key to continue . . .
```

(3 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
36

Please enter the sequence:
YAY

Please enter the base for the conversion: (Enter DNA for DNA conversion)
10

YAY in base 36 equals 44458 in base 10
Press any key to continue . . .
```

(4 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
dna

Please enter the sequence:
tag

Please enter the base for the conversion: (Enter DNA for DNA conversion)
2

TAG in base DNA equals 11011 in base 2
Press any key to continue . . .
```

(5 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
1

Please enter the base of the number: (Enter DNA for DNA conversion)
36

Please enter the sequence:
U2

Please enter the base for the conversion: (Enter DNA for DNA conversion)
dna

U2 in base 36 equals TCCGAA in base DNA
Press any key to continue . . .
```

(6 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
2

Please enter the base for the addition:
16

Please enter the first number:
ABBA

Please enter the second number:
ACDC

ABBA + ACDC = 15896 in base 16
Press any key to continue . . .
```

(7 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
2

Please enter the base for the addition:
8

Please enter the first number:
567

Please enter the second number:
12034

567 + 12034 = 12623 in base 8
Press any key to continue . . .
```

(8 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
2

Please enter the base for the addition:
10

Please enter the first number:
23

Please enter the second number:
45

23 + 45 = 68 in base 10
Press any key to continue . . .
```

(9 of 10)

```
C:\WINDOWS\system32\cmd.exe
Menu
=====
1. Perform Base Conversion
2. Perform Base Addition
3. Quit
2

Please enter the base for the addition:
2

Please enter the first number:
101101

Please enter the second number:
1000110

101101 + 1000110 = 1110011 in base 2
Press any key to continue . . .
```

(10 of 10)

Conclusion

Very glad this assignment involved programming. It feels good to flex my fingers and write some code. In addition, by writing a program that implements the features that we reviewed a couple weeks ago I feel that my knowledge has increased. By coding this program I feel more confident when dealing with numbers in different bases. Looking forward to the next assignment!