

Source

forthright48

`baseToDecimal()` functions have been slightly modified for 16 base numbers (N_{16}). Remove `(x[i] > '9' ? ('A' - 10) : '0')` and use `res += (x[i] - '0') * coef` in `baseToDecimal()` function and `res = (res * base) + (x[i] - '0')`; in `baseToDecimalAlternative()` function to restore the code

```
int baseToDecimal ( string x, int base ) {
    int res = 0;
    int len = x.length();
    int coef = 1; // initially base^0
    for ( int i = len - 1; i >= 0; i-- ) { // Start from reverse
        res += (x[i] - (x[i] > '9' ? ('A' - 10) : '0')) * coef;
        coef *= base; // increase power of base
    }
    return res;
}

int baseToDecimalAlternate ( string x, int base ) {
    int res = 0;
    int len = x.length();
    for ( int i = 0; i < len; i++ ) {
        res = (res * base) + (x[i] - (x[i] > '9' ? ('A' - 10) : '0'));
    }
    return res;
}

char symbol[] = {'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};
string decimalToBase ( int x, int base ) {
    string res = "";
    while ( x ) {
        int r = x % base; // Find the last digit
        res = res + symbol[r]; // Change the integer value to symbol and append to res
        x /= base; // Remove the last digit
    }
    if ( res == "" ) res = symbol[0]; // If res is empty, that means x is 0.
    reverse ( res.begin(), res.end() ); // We found the digits in reverse order.
    return res;
}
```

$$\text{baseToDecimal}(\text{str}, \text{base}) \rightarrow \text{int} : (\text{str})_{\text{base}} \Rightarrow (\text{str})_{10}$$

$$\text{decimalToBase}(\text{intvalue}, \text{intbase}) \rightarrow \text{string} : (\text{value})_{10} \Rightarrow (\text{value})_{\text{base}}$$

```
int main() {
    cout << baseToDecimal("1A", 16) << endl;
    cout << decimalToBase(26, 16) << endl;
}
```