Numeral system converter

Is a C program that converts numbers from one numeral system to another, currently It’s only supporting standard positional numeral systems.

In the program I use this algorithms: <https://blogs.sas.com/content/iml/2022/09/12/convert-base-10.html> for conversion from base10

<https://testbook.com/gate/conversion-to-base-10-notes> for conversion to base 10

If I want to convert from base X system to a base Y system

where X≠10 and Y≠10.

I just convert base X to base 10 first and then I convert this base 10 to base Y.

Version basic:

Its the version of the program that is the most barebone and simplistic one, you can only convert from bases 1-10.

Functions:

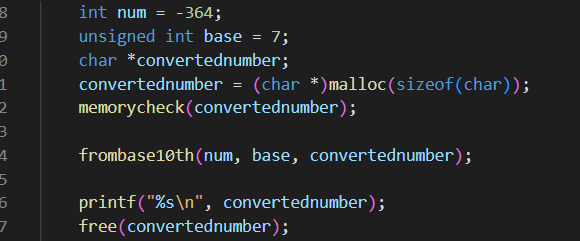
void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base smaller than base 10  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-10

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-10 base. it is a char because the converted number can be negative so it give it a “-” sign beforehand. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

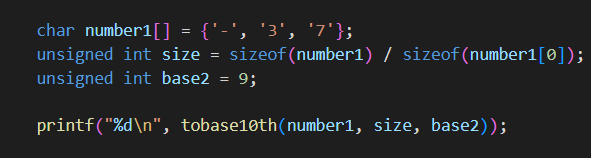
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.

unsigned int base

base of the number you are going to convert, has to be base 1-10

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

size\_t size

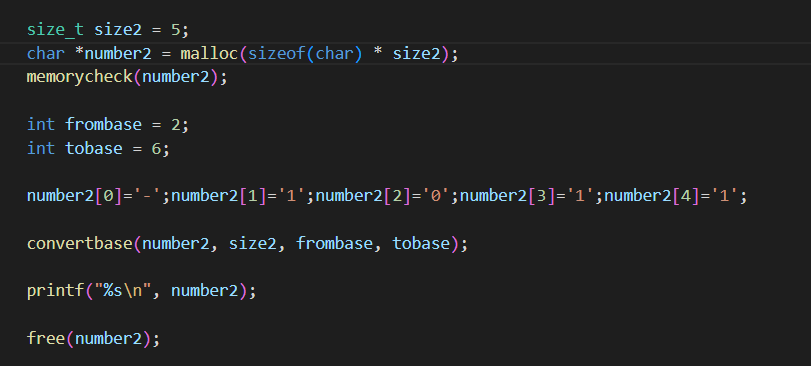
size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage

int



console



Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

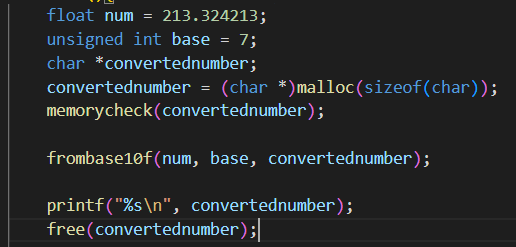
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base smaller than base 10  
float number - Floating number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-10

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-10 base. it is a char because the converted number can be negative so it give it a “-” sign beforehand. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting floating numbers from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

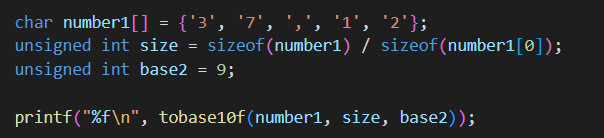
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.

unsigned int base

base of the number you are going to convert, has to be base 1-10

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

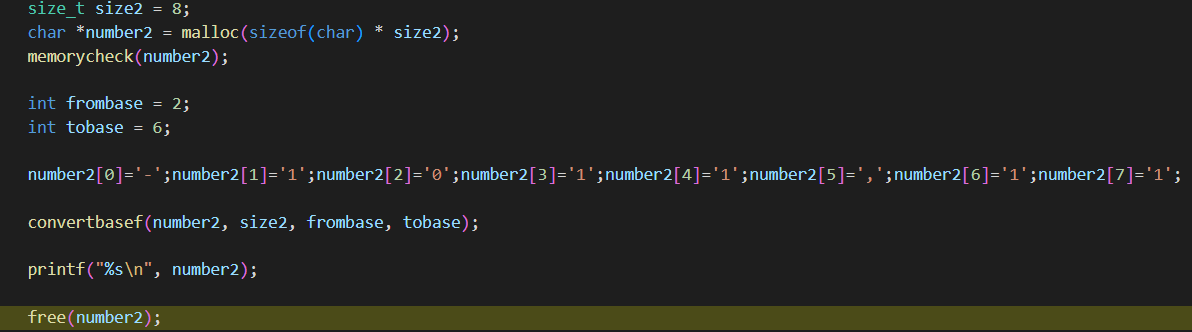
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



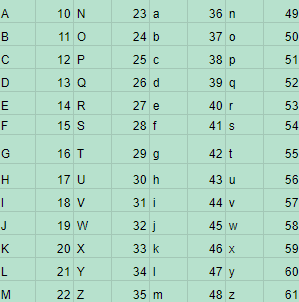
console



Potential use cases:

It can be easily embedded to the calculator application and from that used for personal user use. It could also be used when you have a lot of below base 10 (for example octal or binary) data and need to convert it regularly to decimal for users to read, and using a more general (like the next version) is not needed or detrimental.

Basic extended version



Symbols for digits, 62 is the highest supported base

Using upper case letters for digits above 9 is pretty common so its should self explanatory, after 35 I use lowercase letters as it more readable and doesn’t use symbols allowing for traditional mathematic notation.

Functions:

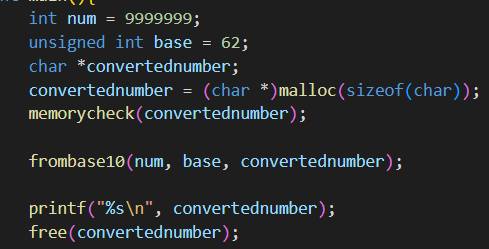
Void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base smaller than base 63  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-62

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-62 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console



int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

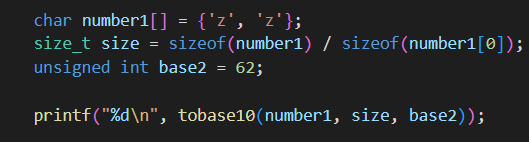
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-62

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

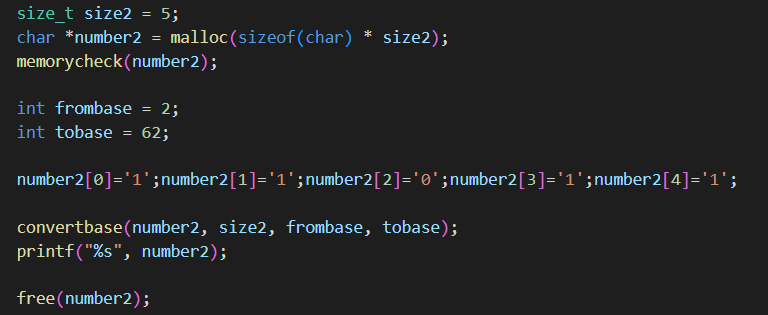
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console



Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

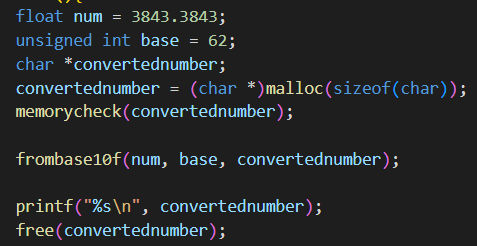
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base smaller than base 63  
float number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-62

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-62 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

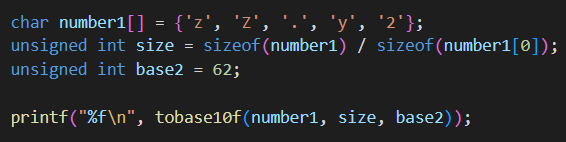
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-62

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

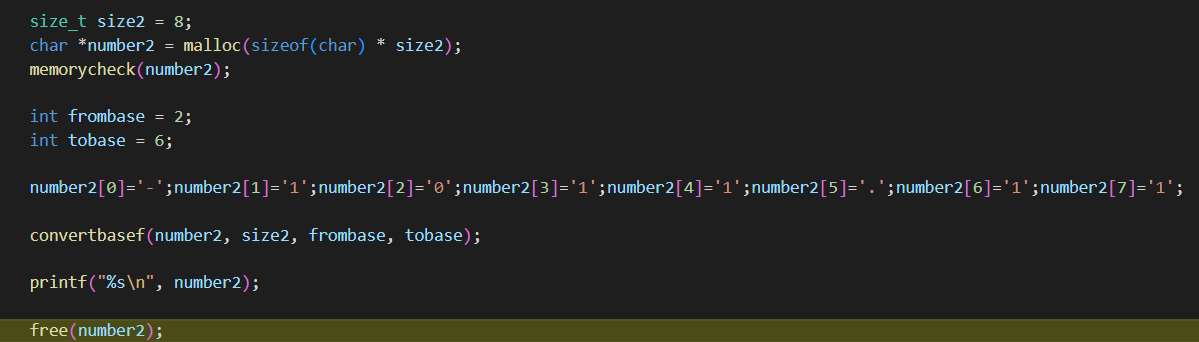
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



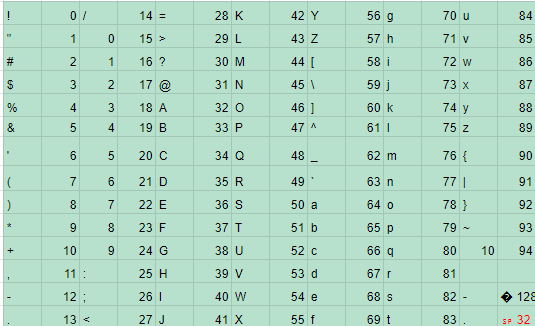
console



Potential use cases:

Can be embedded into calculator application, also can be used for storing large numbers as base 62 takes less digits to write than other bases, and still keeps the numbers readable for normal users.

Programmer version:



Symbols for digits, 94 is the highest supported base

As digits and symbols this version uses every printable ascii character (32-126), symbols are � or ascii character 128, this symbol represents negative numbers, and ␠ or ascii character 32 which represents decimal separator (.)

Functions:

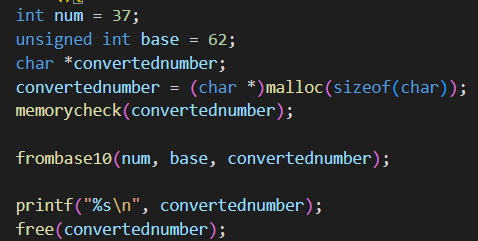
Void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base smaller than base 95  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-94

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-94 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console



int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

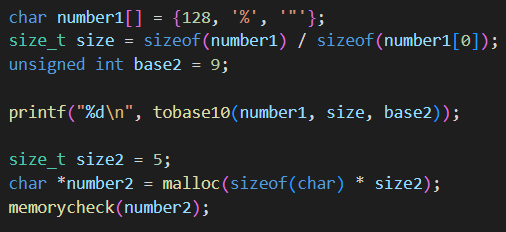
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-94

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

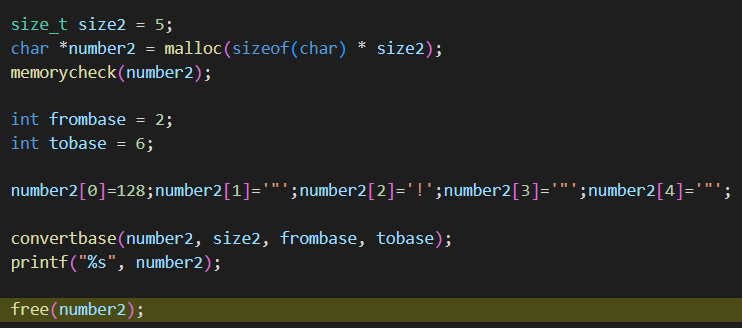
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console



Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

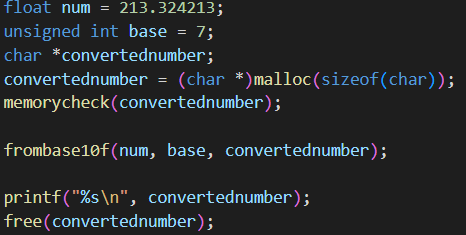
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base smaller than base 95  
float number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-92

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-92 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

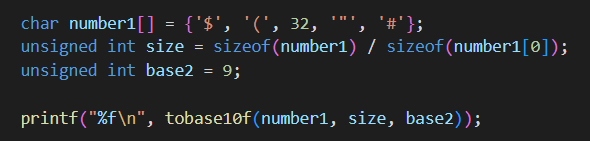
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-92

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

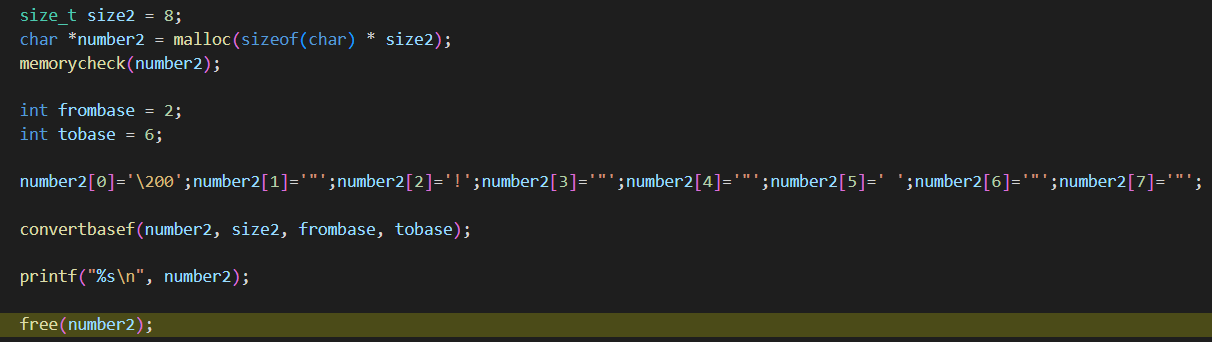
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console

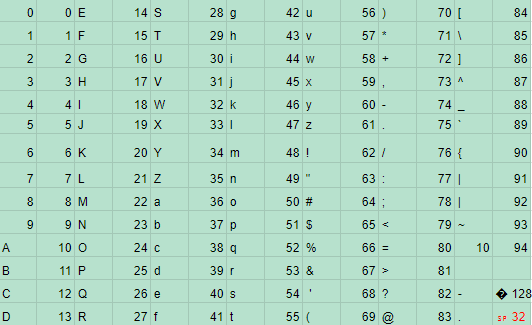


Potential use cases:

Can be embedded into calculator application, also can be used for storing large numbers as base 94 takes less digits to write than other bases, and still keeps the numbers readable for people acquainted with the program.

Programmer extended version:

is a Basic extended and Programmer version conjoined, it uses 0-z as digits for 0-61 numbers and from 62 onwards it uses non number/letter ascii characters as digits. Thanks to that it is more readable than normal programmer version. It uses the same symbols as programmer version.



Functions:

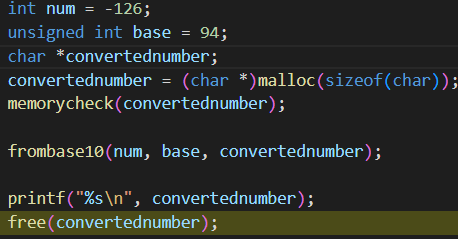
Void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base smaller than base 95  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-94

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-94 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console



int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

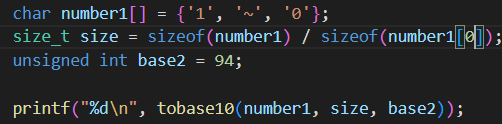
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-94

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

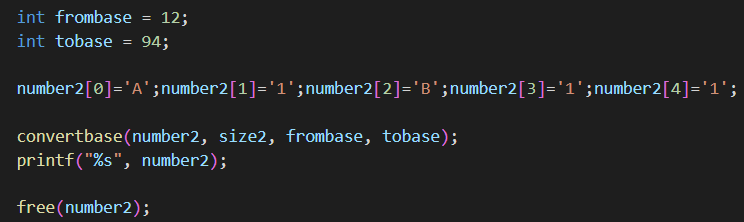
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console



Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

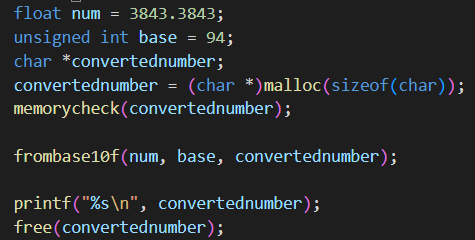
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base smaller than base 95  
float number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-92

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-92 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

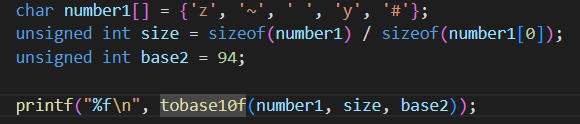
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-92

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

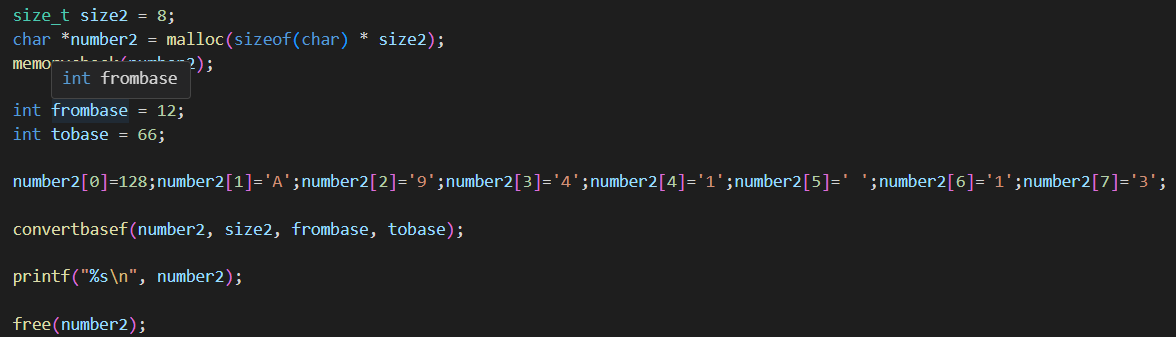
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console



Potential use cases:

Can be embedded into calculator application, also can be used for storing large numbers as base 94 takes less digits to write than other bases, and still keeps the numbers readable.

Unreadable version:

uses ascii characters from 0 to 253 as numbers, 255 as - sign and 254 as decimal separator. due to how it functions it is completely unreadable and can potentially unusable. Treat it more as a experiment that actual program with uses.

Functions:

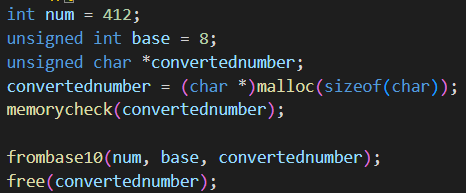
Void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base smaller than base 254  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-253

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-253 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console

doesn’t return anything but in array number looks like this:

␆␃␄

int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

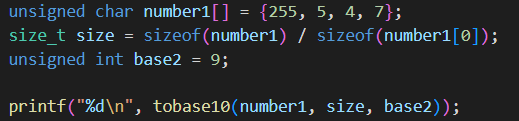
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-253

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

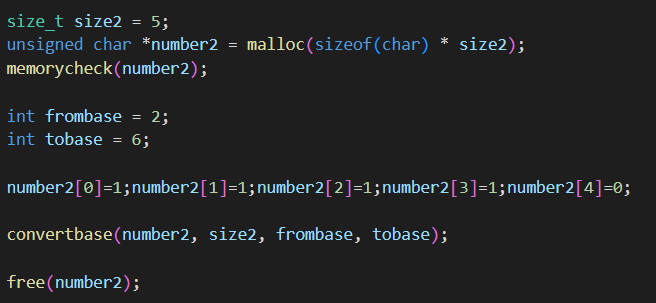
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console

doesn’t return anything, in array number looks like this:  
␅␀

Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

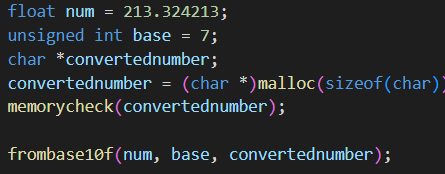
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base smaller than base 254  
float number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base 1-253

char convertednumber[] - array that will store the converted number to the base of yours choosing, that is between 1-253 base. it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:

console doesn’t return anything, array looks like this:

‹þ~␐␅

float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

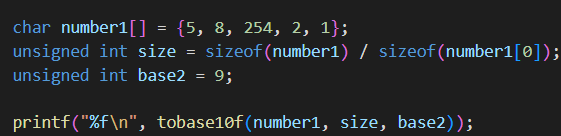
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert, has to be base 1-253

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

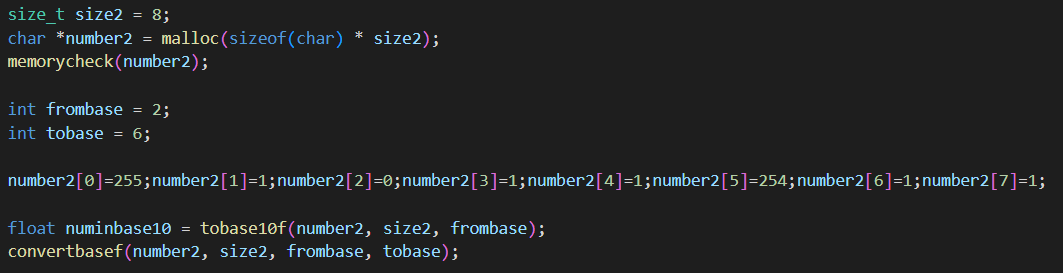
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console

doesn’t return anything, array looks like this:

ÿ␁␅þ␂␁␅␂

Infinite version:

Is an another experiment, this time without it potentially destroying your data so it can be used. it stores 62 base numbers in bracelets which then represent digits in other bases, that means that numbers in bracelets for example (a1b) represents digit with value 138483. Thanks to this you can use infinitely huge base (in practice you can only use base as big as the integer limit), tho its practical usage is questionable, as it takes more memory than just using basic extended or more optimised version. It uses the same - sign and decimal separator as basic extended.

Functions:

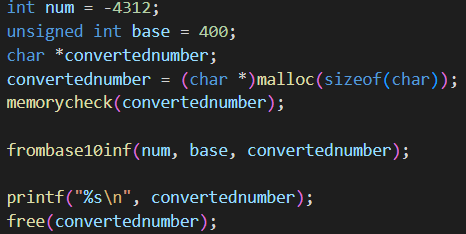
Void frombase10(int number, unsigned int base, char convertednumber[])

This function converts number from base 10 to any other base  
int number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into, has to be base >1

char convertednumber[] - array that will store the converted number to the base of yours choosing, it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console



int tobase10(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

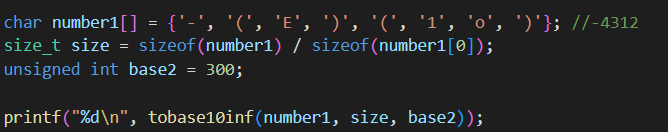
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert

example of usage



console:



void convertbase(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

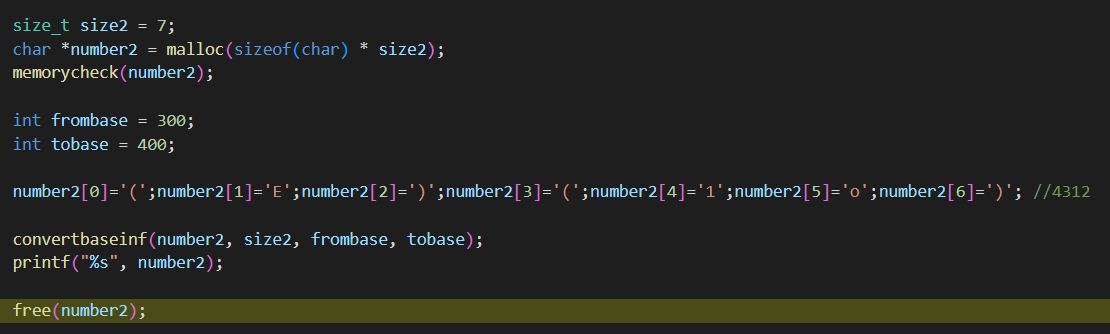
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



console



Floats:  
due to way C handles multiplying floating numbers some value will become corrupted, it isn’t recommended to use it for high precision decimal points.

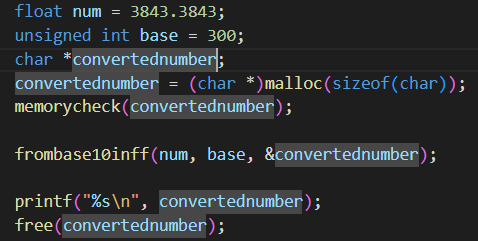
void frombase10f(float number, unsigned int base, char convertednumber[])

This function converts floating numbers from base 10 to any other base  
float number - Number in the base 10 that you want to convert

unsigned int base - Base that you want to convert number into

char convertednumber[] - array that will store the converted number to the base of yours choosing, it is a char as digits can be letters too. You have to allocate memory to it beforehand, (even if only sizeof(char)), because it reallocates its size automatically so the convertednumber fits into the array. (for example if you want to convert 4 to base 2 the size of array will automatically change to sizeof(char)\*3 to accommodate the new digits.

example of usage



console:



float tobase10f(char number[], size\_t size, unsigned int base)

Function for converting number from any base to base 10.

char number[]

number in other base to be converted, it doesn’t have to be allocated as this function doesn’t change anything about it.

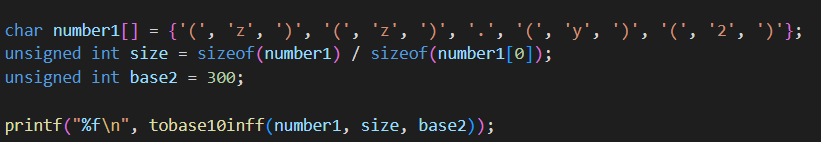
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along, made in case someone wanted to work with them.

unsigned int base

base of the number you are going to convert

example of usage



console:



void convertbasef(char number[], size\_t size, unsigned int frombase, unsigned int tobase)

Function for converting floating number from one base to another, it works by first converting the number to base 10 and then this newly converted number to base of yours choosing.

char number[]

Number that you want to convert, you have to allocate memory to it because inside of it a converted number will be inserted

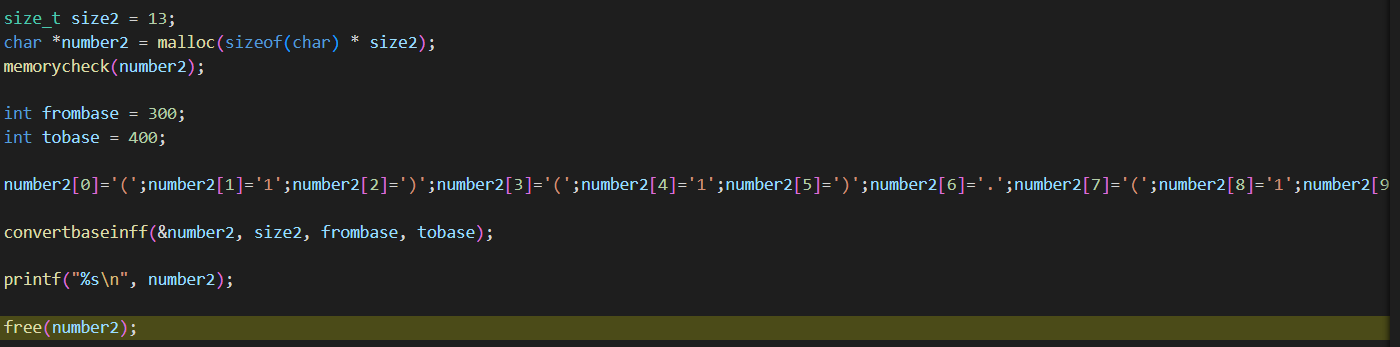
size\_t size

size of the array, has to be given to the function as you can’t know the size of an allocated array if it isn’t passed along.   
unsigned int frombase()  
Base numeral system of the number

unsigned int tobase()

Base numeral system you want the number to convert to.

example of usage



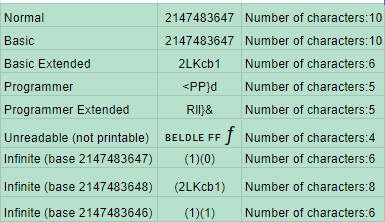
console



Potential versions

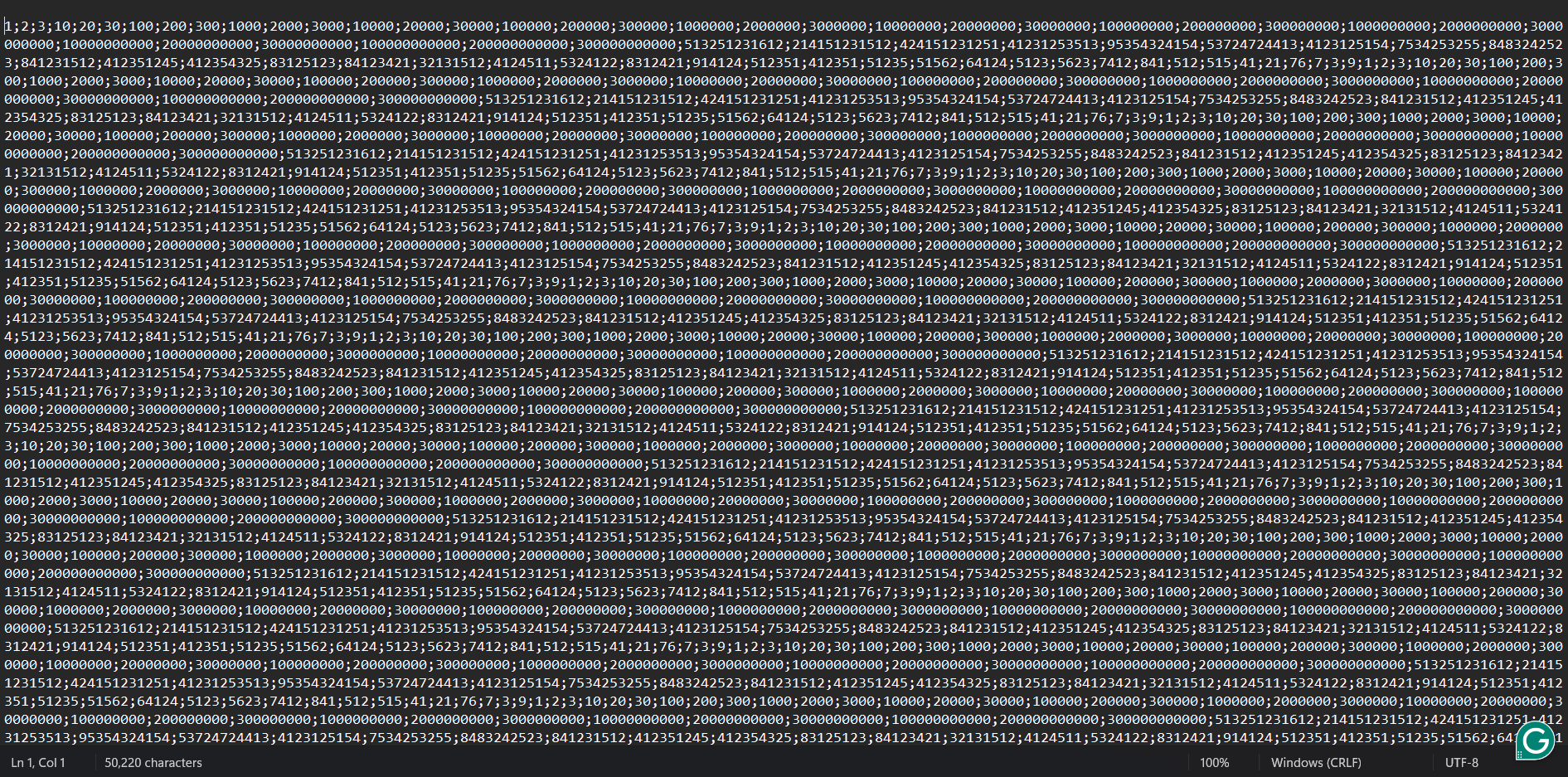
In theory you could use unicode characters which (as of version 15.1) defines over 149 813 characters giving you even more more room for optimization (tho you have to keep in mind that it takes up more bytes per digit)

Number 2147483647 in different versions (maximal base used)

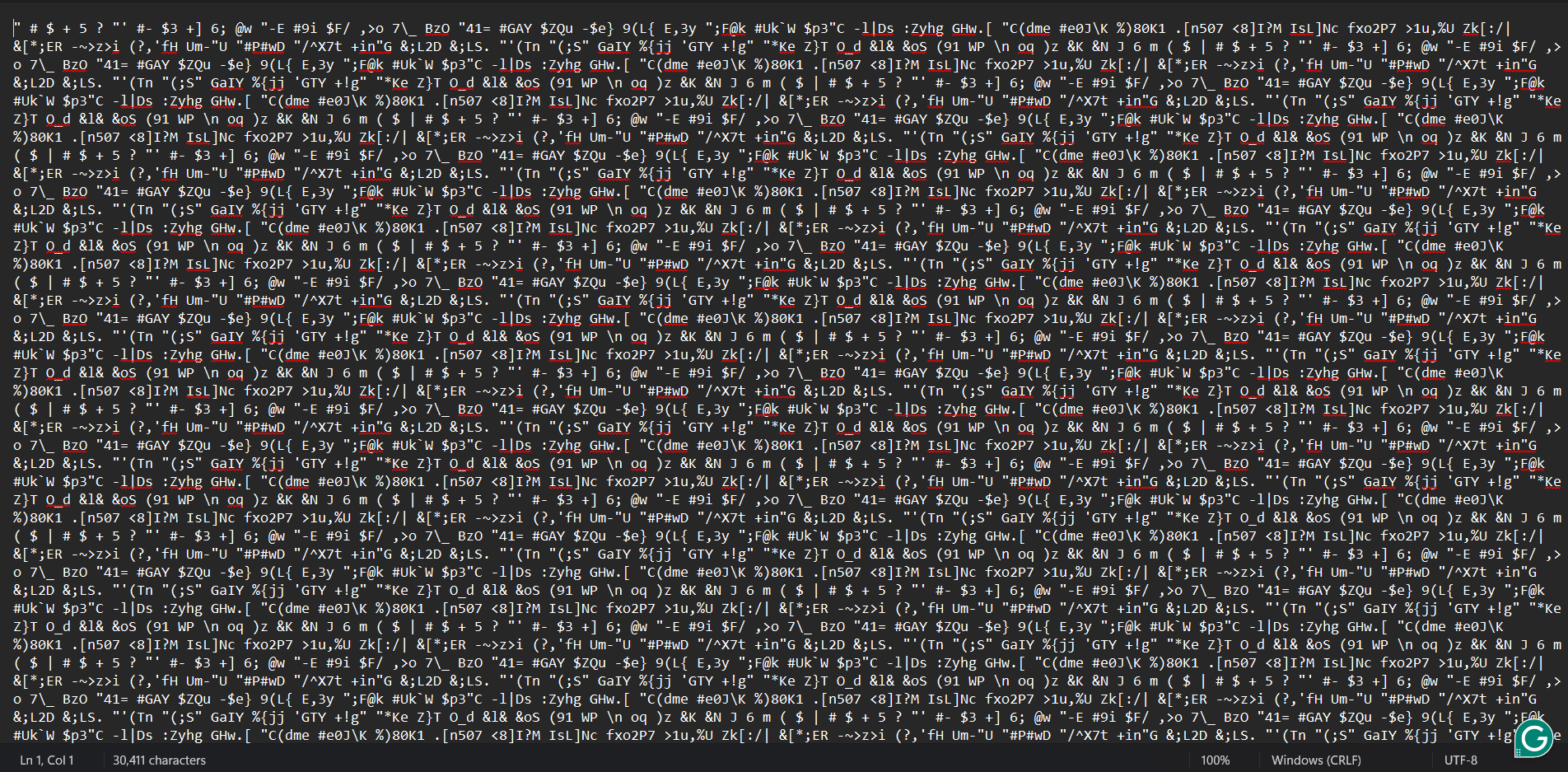


“Functional” program using Numeral system converter

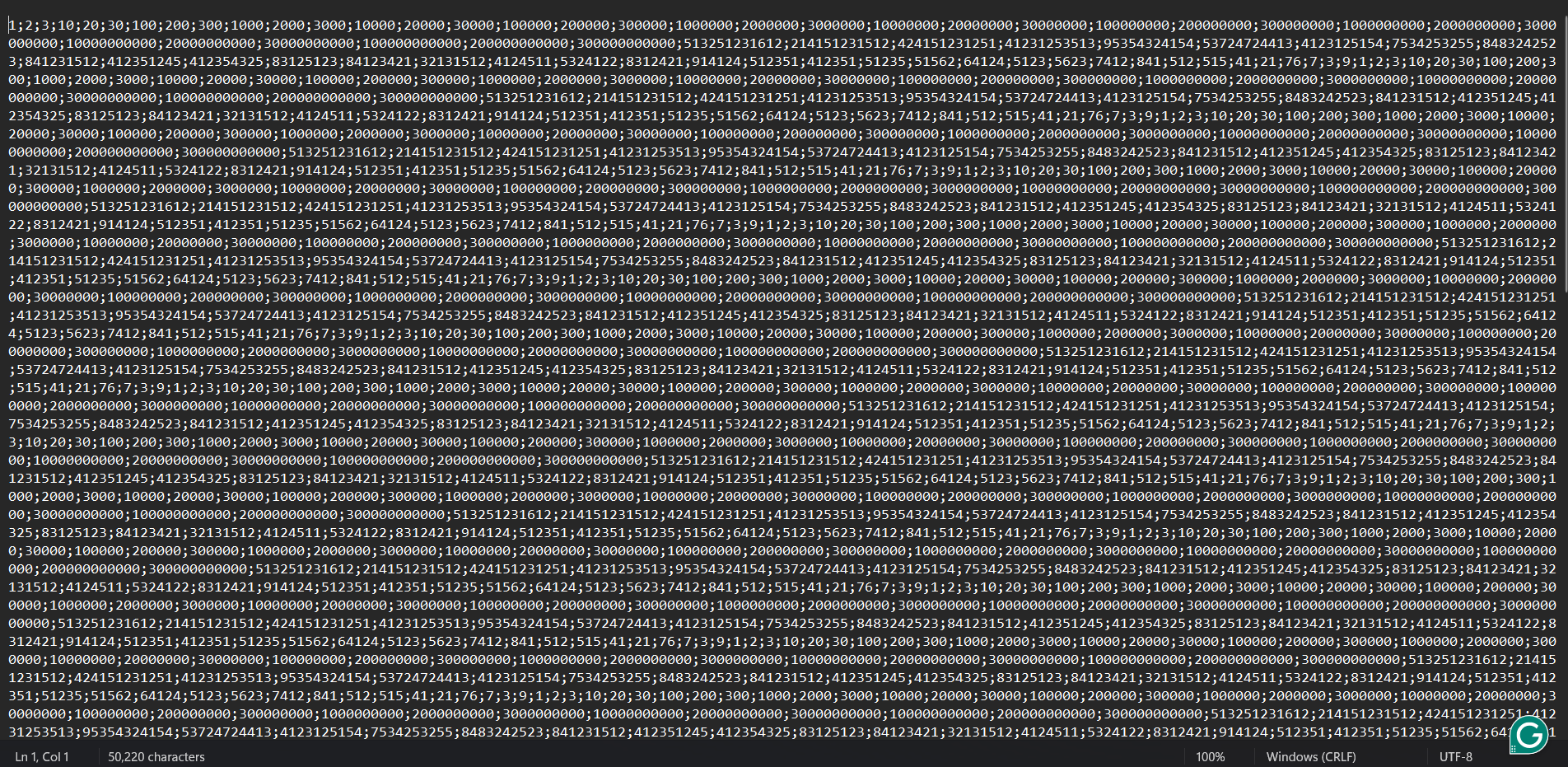
This program converts from file (for example sql database in .txt file) with base 10 numbers to other base numbers (in my example base 94).



File with numbers (50kb), ‘;’ is separator



File with numbers (30kb), ‘ ‘ is separator



File with numbers (50kb), ‘;’ is separator

File is compressed by about 40%, I use many small numbers (like 1-9) so it isn’t as compressed as it could if I only used big numbers (12 digits numbers, now takes 6-7 digits or even bigger numbers). It would take even less memory, if you decide to use floating numbers remember to change the separator.