

Assignment 3

CAAM 519 Fall 2021

due Wednesday October 27 (10:59 am)

In this assignment you will practice setting up source and header files in C. You will make a program that will take in a positive base-10 number as a command line input and convert it to a base-2 number. Please heavily comment each part of your code.

- (2pt) Create a header file called `conversion_functions.h`.
- (2pt) In this header file, define a prototype for a function called `convert_from_base_10_to_base_2`. This function should have one `int` parameter and return a `long long int`. The input parameter will be a base-10 number and the return value will be the base-2 representation of the input, stored as a `long long int`.
- (2pt) Create a source file called `convert.c`. Within this source file, you will implement `convert_from_base_10_to_base_2` as well as `main`.
- (2pt) In `convert.c`, you will want to include `stdlib.h`, `stdio.h` as well as your header file you created. This will need to be included as

```
#include "conversion_functions.h"
```

with quotations, so that the compiler will look for this in your working directory.

- (13pts) The implementation of `convert_from_base_10_to_base_2` should follow the algorithm given in class using the remainder and quotient operators available in C. Note that this approach will give you the digits (either zero or one) of the binary representation, but you need to figure out how to put these digits into a `long long int` representation. Address why it is important to use a `long long int` as opposed to just an `int` data type. When would this approach using a `long long int` representation fail? Please answer these questions in the L^AT_EX document you create.
- (2pt) Your main function should take in the base-10 integer on the command line. For example, if you want to pass in the base-10 integer 314 on the command line, you would call the executable like

```
./main 314
```

You can take in a command line input using the following implementation

```
int main(int argc, char *argv[])
{
    char *input = argv[1];
    int base_10_number = atoi(input);
    ...
}
```

It should return the binary representation of the command line input within the terminal. Note that the format specifier for a long long int is %lld.

- (2pt) Insert your code into a tex file and generate a pdf using L^AT_EX. Also, please include the output from your code by running it with (1005)₁₀ as an input on the command line. Here is an example of inserting some code using the command

```
\lstinputlisting{../example_hello_world/hello_world.c}
```

```
1 // hello world example
2 #include <stdio.h>
3
4 int main()
5 {
6     printf("%s", "hello world!!\n");
7 }
```

You will need to include the following at the top of your tex file, before \begin{document}

```
\usepackage{color}
\definecolor{shadecolor}{rgb}{0.8,0.8,0.8}
\usepackage{listings}
\lstset{
    language=C,                % choose the language of the code
    numbers=left,              % where to put the line-numbers
    stepnumber=1,              % the step between two line-numbers.
    numbersep=5pt,             % how far the line-numbers are from the code
    backgroundcolor=\color{shadecolor}, % choose the background color. You must add \usepackage{color}
    showspaces=false,          % show spaces adding particular underscores
    showstringspaces=false,    % underline spaces within strings
    showtabs=false,            % show tabs within strings adding particular underscores
    tabsize=2,                  % sets default tabsize to 2 spaces
    captionpos=b,              % sets the caption-position to bottom
    breaklines=true,           % sets automatic line breaking
    breakatwhitespace=true,    % sets if automatic breaks should only happen at whitespace
    % title=\lstname,          % show the filename of files included with \lstinputlisting;
}
\lstset{basicstyle=\ttfamily\footnotesize,breaklines=true}
```

Please upload your pdf generated from L^AT_EX and source and header files to Canvas.