

CSE 379 Lab Report

Timothy Scholtz - tjscholt - 50393083

Thomas Mahok - tjmebok - 50407528

Lab 3

Date - 13 Feb

Lab section R2



Table of Contents

| | |
|----------------------------|----------|
| Section 1 | 1 |
| Project Breakdown..... | 1 |
| Section 2 | 1 |
| Program Overview..... | 1 |
| Program Summary..... | 2 |
| High Level Flowchart..... | 2 |
| Section 3 | 3 |
| Subroutine List..... | 3 |
| Section 4 | 3 |
| Subroutine Flowcharts..... | 3 |

Section 1

Project Breakdown

| Task Description | Person who completed it |
|---|-------------------------|
| Output_character - wrote a subroutine which takes a value from a register and sends it to the Uart | Tim |
| Read_character - wrote a subroutine which takes a value from the uart and stores it in a resistor | Tom |
| Debugging - putting all the individual parts together and testing that it works | Tim & Tom |
| Uart_init - converted given C code to arm assembly and tested it | Tim & Tom |
| Output_string - wrote a subroutine which transmits a NULL-terminated ASCII string for display in PuTTY. | Tim |
| read_string - wrote a subroutine reads a string entered in PuTTY and stores it as a NULL-terminated ASCII string in memory | Tom |
| int2string - stores the integer passed into the routine in r1 as a NULL terminated ASCII string in memory | Tim |

| | |
|---|------------------|
| string2int - converts the NULL terminated ASCII string pointed to by the address passed into the routine in r0 to an integer | Tom |
| Div_and_mod - adapted code from lab 2 to work with lab 3 | Re-used old code |

Section 2

Program Summary

The program allows the user to input two integers and receive the quotient and the remainder of the inputs. It achieves this by creating a Uart connection to the serial display, which enables the user to send data to the program and receive feedback from it.

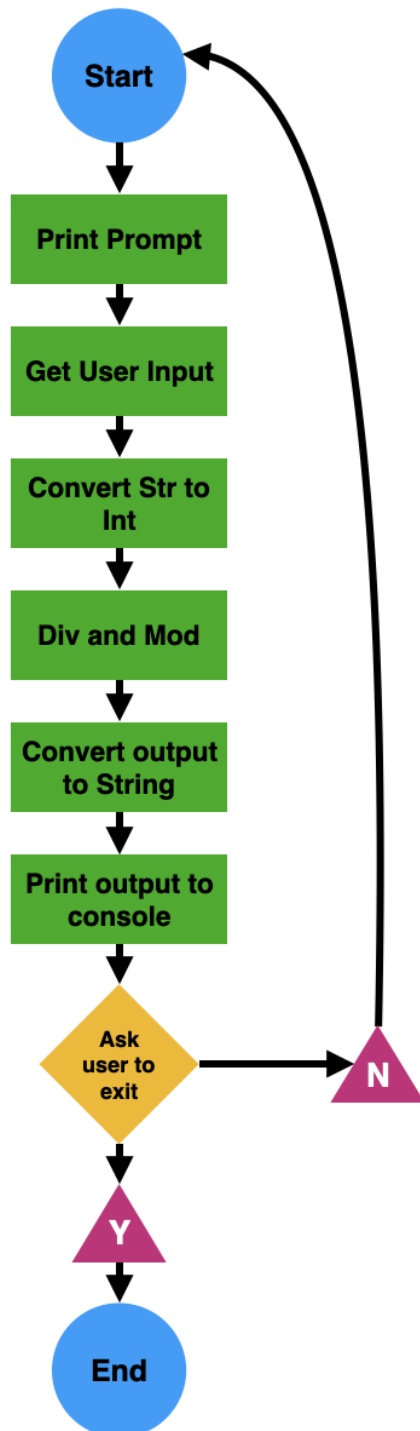
Program Overview

Use instructions:

- Start the program
- Directions are displayed in the Terminal
- You will input 2 numbers, a divided and a divisor
- You will be returned the number of divisions the program executed and the remainder(MOD)
- Once the operation is completed you will be given an option to END or RESTART
 - If RESTART, you will be greeted by the same direction and the program will act as described above
 - If END, you will be greeted with an ending message and the program will finish its execution and finish.

High-Level Flowchart

Provides an overview of the entire program A single box in the flowchart may indicate a call to a subroutine such as `read_string` or `output_string`



Section 3

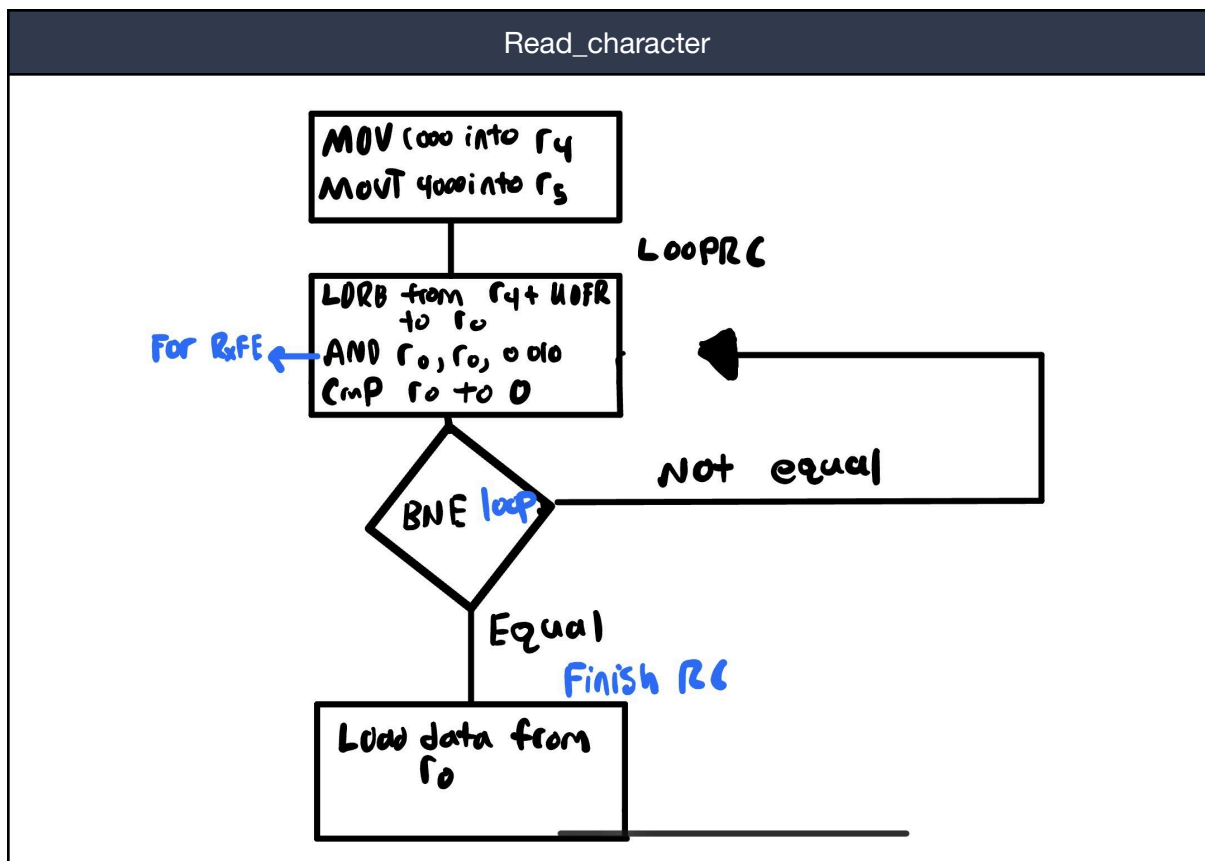
Subroutine List

- **Uart_init:**
 - Functionality: Initialises the UART for use
 - Arguments: None
 - Return Values: None
- **output_character:**
 - Functionality: Transmits a character from the UART to PuTTY.
 - Arguments:
 - R0 - Character value
 - Return Values: None
- **output_string:**
 - Functionality: Transmits a NULL-terminated ASCII string for display in PuTTY.
 - Arguments:
 - R0 - memory address to string
 - Return Values: None
- **read_character:**
 - Functionality: reads a character that is received by the UART and returns it
 - Arguments: None
 - Return Values:
 - R0 - return val of char
- **read_string:**
 - Functionality: reads a string entered in PuTTY and stores it as a NULL-terminated ASCII string in memory.
 - Arguments:
 - R0 - base address to store string
 - Return Values: None
- **int2string**
 - Functionality: Takes an integer, converts it as a string, and stores it in memory
 - Arguments:
 - R0 - memory address
 - R1 - integer value
 - Return Values: None
- **string2int**
 - Functionality: takes a memory address of a null-terminated string and returns that string converted to a number
 - Arguments:
 - R0 - memory address
 - Return Values:
 - R0 - Int

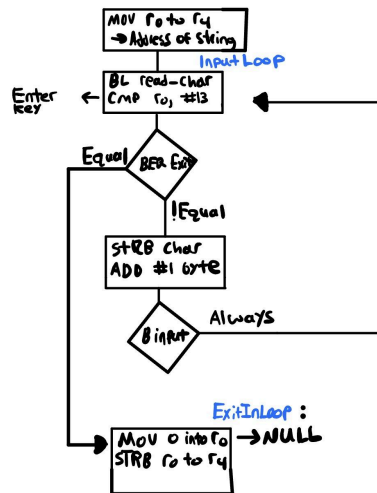
- **div_and_mod**
 - Functionality: Takes a divisor and dividend and returns the quotient and remainder(positive)
 - Arguments:
 - R0 - Dividend
 - R1 - Divisor
 - Return Values:
 - R0 - Quotient
 - R1 - Remainder

Section 4

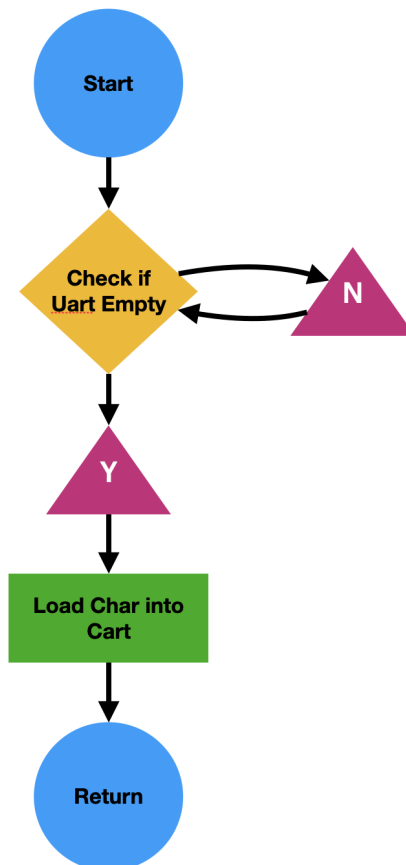
Subroutine Flowcharts



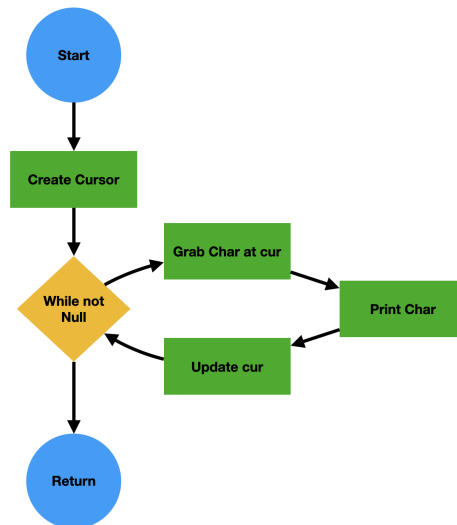
Read_string



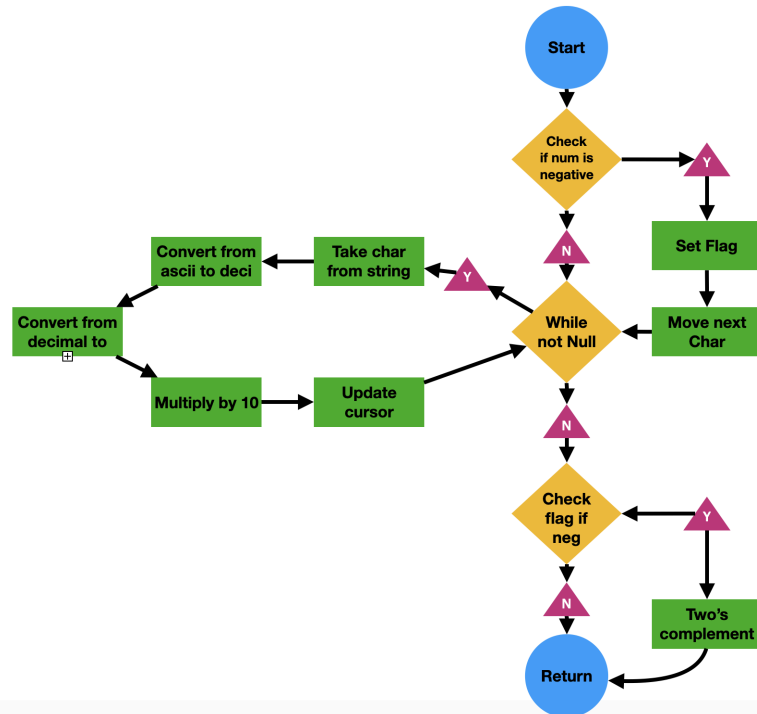
Output_char



Output_string



string2int



int2string

