Program Design & Testing Document for Program 3
Lyell C Read

Problem Statement

Design a program that will make algebra fun. This program will start by asking the user how many “floors” to have, then will prompt the user to to answer an algebra problem for each floor. If they succeed on all “floors”, they win, and get to the top, while if they fail, they descend a level. There will be two modes: difficult and easy. Easy mode will allow the user to choose what operations they are required to conduct, while difficult mode will choose randomly for the user. In easy mode, numbers generated to fill the equations will be on [0..100], while in difficult mode, these numbers will be on [-100..100]. ASCII art must be used to represent the building progress.

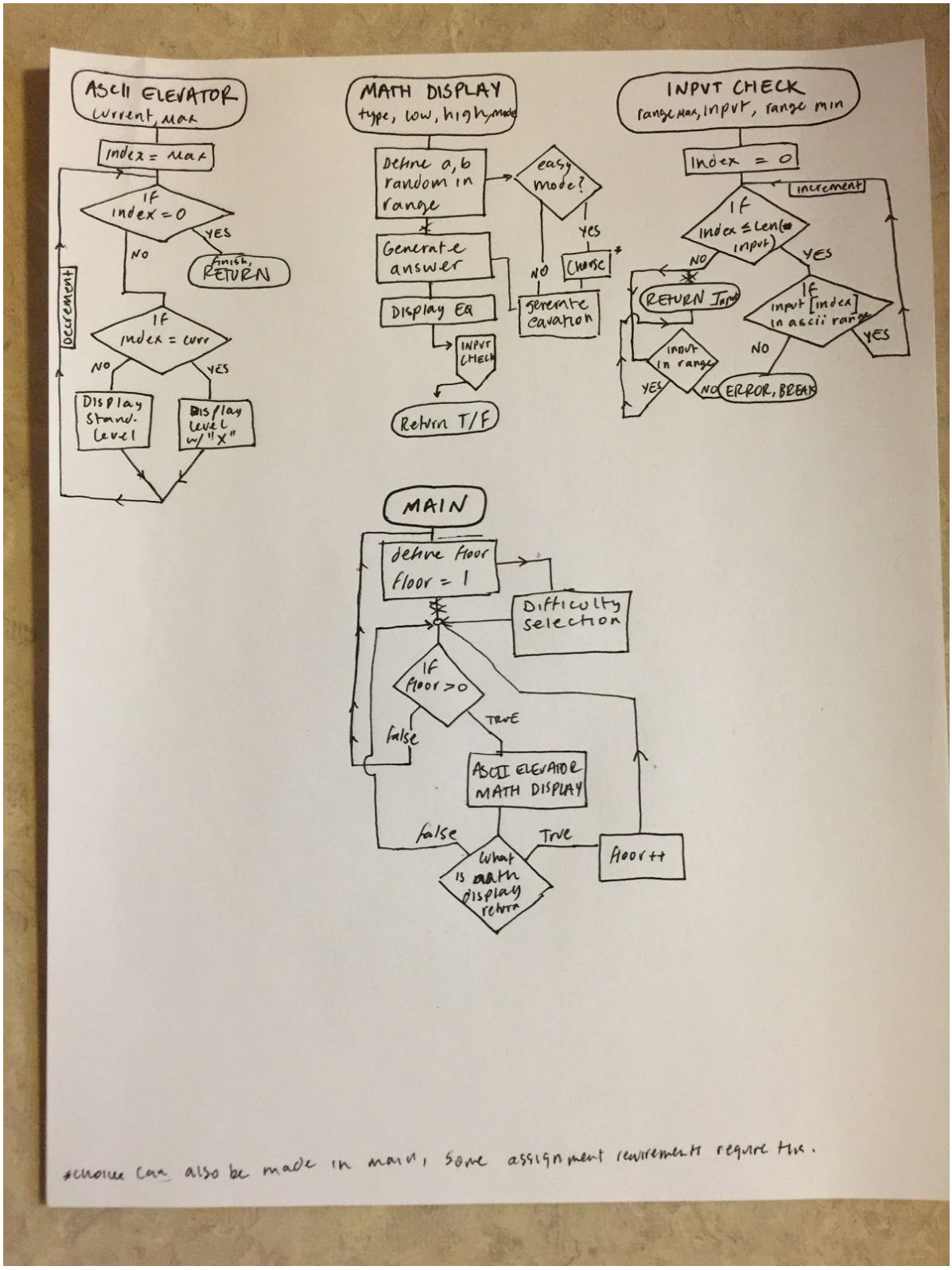
More Requirements:

- Must use at least three functions
- Must use self made function to check and convert input (str) to signed int.
    - If the str contains illegal characters, return INT_MAX from <climits>
- Must use a self made function to print the image of an elevator using ASCII characters.
    - This should accept two parameters: current level, max level
- Must use a self made function to print the math problem to screen.
    - Must be named display_equation()
    - Must accept three parameters: (type of problem, lowest acceptable random number, highest acceptable random number)
        - 1 for addition, 2 for subtraction, 3 for multiplication, 4 for division (canvas)
    - Must return answer as a signed int
    - MUST USE: **int display_equation(int eq_type, int range_low, int range_high);**
- Must catch all bad input
- Math answers should always be ints
- Elevator cannot go below the first floor...
- User must be able to choose to repeat.

Understanding the Problem

The problem is asking me to develop an algebra game, the exact intricacies of which are listed above. This game will include as many (up to 10) floors as the user decrees and each floor will have an algebra problem with randomly generated values (and thus answer) and user input will be checked to make sure that it is indeed an int, otherwise, MAX_INT will be returned.

Devising a Plan



Pseudo Code

```
Ascii_art (current level, max level):
    For x on 0..max level:
        If x == current level
            Print  |__X__|
        Else:
            Print |_____|
    Print _____

display_equation(type, low, high)
    While not a%b ==0: //make sure that they are divisible cleanly so no float work.
        Int a = random in range low..high
        Int b = random in range low..high

    If type == 1:
        //addition
        Print what is a + b?
        Input = input_check(high, low)
        While == INT_MAX:
            Print enter again:
            Input = input_check(high, low)
        If input == a+b
            Return true
        Return false
    If type == 2:
        //subtraction
        Print what is a - b?
        Input = input_check(high, low)
        While == INT_MAX:
            Print enter again:
            Input = input_check(high, low)
        If input == a-b
            Return true
        Return false
    If type == 3:
        //multiplication
        Print what is a X b?
        Input = input_check(high, low)
        While == INT_MAX:
            Print enter again:
            Input = input_check(high, low)
        If input == a*b
```

```
                    Return true
                Return false
        If type == 4:
                //division
                Print what is a / b?
                Input = input_check(high, low)
                While == INT_MAX:
                    Print enter again:
                    Input = input_check(high, low)
                If input == a/b
                    Return true
                Return false

input_check(max, min):
        index=0
        String input = user input get line
        For index in range 0..len(input):
            If not input[index] in range 48..57:
                Return INT_MAX
        If input in range min..max:
            Return input
        Else: return INT_MAX


main():
Cont = 1
While cont == 1:
        Current floor = 1
        Print what difficulty 2 for hard, 1 for easy
        Diff  = input_check(2,1)
        While diff == INT_MAX:
            Print Choose Again:
            Diff = input_check(2,1)
        floors_total = input_check(10,1)
        While floors_total == INT_MAX:
            Print Choose Again:
            Floors_total = input_check(2,1)
        While floor > 0:
            ascii_art(floor, floors_total)
            Problem = 0
            If diff == 1: //easy mode
                Print choose operation 1a 2s 3m 4d
                Problem = input_check(4,1)
```

```
                While problem == INT_MAX
                        Problem = input_check(4,1)
                solution= display_equation(problem, 0, 100)
        Else:
                //hard mode
                Problem = random in range 1..4
                solution= display_equation(problem, -100, 100)
        If solution == False
                Floor--
        If solution == True
                Floor++
Print play again?
Cont =input_check(1,0)
```

Testing User Input Function (“input_check”)

Value (passed range 1..10)	**What Should Happen**	**Does This Happen**
2	Return 2	yes
3j	Return INT_MAX	yes
00010	Return 10	yes
‘\t’	Return INT_MAX	yes
thisisanintegertrustme	Return INT_MAX	yes
;’ ‘’; ‘	Return INT_MAX	yes
	Return INT_MAX	yes