

## Assignment 1 (10% of Course Total)

Due date: 11:59pm, Jan 29, 2021

The assignment will be graded automatically. Make sure that your code compiles without warnings/errors and produces the required output. Also use the file names and structures indicated as requested. Deviation from that might result in 0 mark.

Your code **MUST** compile and run in the CSIL machines with the Makefile provided. It is possible that even with warnings your code would compile. But this still indicates there is something wrong with your code and you have to fix them, or marks will be deducted.

Your code **MUST** be readable and have reasonable documentation (comments) explaining what it does. Use your own judgement, for example, no need to explain `i += 2` is increasing `i` by 2, but explain how variables are used to achieve something.

### Description

There is a total of 4 questions in this assignment. For each question, write your answer in a single file that contains your student information. Unless otherwise specified, do not include any libraries. You can however write your own helper functions. Also do not print anything unless the question asks you to do so. None of these files should contain the main function.

#### Question 1 [8 marks]

Write a function that interlaces a positive 4-digit int between 1000 and 9999 with another positive 5-digit int between 10000 and 99999, and returns the result as a positive 9-digit int. Use this function header:

```
unsigned int interlace_ints(unsigned int firstNum, unsigned int secondNum)
```

For example:

`interlace_ints(1000, 99999)` should return 919090909

`interlace_ints(2345, 11111)` should return 121314151

`interlace_ints(2468, 13579)` should return 123456789

Only include the function definition in your answer and name the source file containing it as **question1.c**. Do not use recursion in your answer.

#### Question 2 [8 marks]

Write a function that takes in 4 parameters: an int array, its size, an int as a target, and a char array of size 32; and puts a cstring in the format **A+B** to that char array, where A and B are the indexes of a pair of elements in the int array that add up to the target value (note that A and B cannot be the same). If there are multiple pairs, any pair will do. If no such pair exists, put the cstring that reads **not\_found**. Use this function header:

```
void findPair(int array[], unsigned int size, int target, char* result)
```

For example:

Given the array `intArr1: {-1, 0, 2, 3, 4, 5, 6, 7}` and the char array `pair`

`findPair(intArr1, 8, -1, pair)` should put the cstring `0+1` in `pair`

`findPair(intArr1, 8, 5, pair)` should put the cstring `0+6` in `pair` (`1+5` and `2+3` are also valid)

`findPair(intArr, 8, 25, pair)` should put the cstring `not_found` in `pair`

You can assume there are at least 2 elements in the int array and the size is a valid number (not negative and is the actual size of the array). You can also assume that the resulting pair will not exceed 31 characters. As a hint, look up the `sprintf` function instead of `printf`. Only include the function definition in your answer and name the source file containing it as **question2.c**. Do not use recursion in your answer.

### Question 3 [8 marks]

Write a function that takes in the dimensions (rows and columns) of a 2D array of ints and the array itself, and performs a row subtraction operation as indicated by 2 other parameters representing the row numbers (indexes, start with 0, not 1). This function does not return anything as the change is within the 2D array. Use this function header:

```
void rowSubtraction(unsigned int rows, unsigned int columns,  
                    int matrix[rows][columns], unsigned int row1, unsigned int row2)
```

For example:

Given the 3x4 2D array `matrix1: {{1, 2, 3, 2}, {2, 3, 4, 3}, {1, 1, 1, 1}}`,

`rowSubtraction(3, 4, matrix1, 0, 1)` modifies `matrix1` into `{{-1, -1, -1, -1}, {2, 3, 4, 3}, {1, 1, 1, 1}}`

`rowSubtraction(3, 4, matrix1, 1, 2)` further modifies `matrix1` into `{{-1, -1, -1, -1}, {1, 2, 3, 2}, {1, 1, 1, 1}}`

`rowSubtraction(3, 4, matrix1, 2, 3)` does not cause any changes to `matrix1` as there is no row indexed 3

You can assume rows and columns represent the dimensions of the array correctly and the dimensions are either both positive (regular matrix) or both zero (empty matrix). However, any of `row1` and `row2` can have values accessing beyond the number of the rows and if that happens the matrix remains unchanged. Only include the function definition in your answer and name the source file containing it as **question3.c**.

### Question 4 [12 marks]

Write a function that takes in a cstring (i.e., a character array that has a `\0` character as the last element) that represents a mathematics operation (+, -, or \*), and returns the result as an int. Use this function header:

```
int performMathOp(char* operation)
```

For example:

`performMathOp("12+345")` should return 357

`performMathOp("0-4")` should return -4

`performMathOp("13*13")` should return 169

You can assume that the representation is valid (digits for the 2 numbers, one of the +/~/\* between them) and neither of them are negative, with no space between any of the numbers and the operator. Only include the function definition in your answer and name the source file containing it as **question4.c**.

### Coding Style [4 marks]

Your program should be correctly indented, have clear variable names and enough white space and comments to make it easy to read. Named constants should be used where appropriate. Each line of code should not exceed 80 characters. White space should be used in a consistent manner.

To help you to get into the habit of good coding style, we will read your code and marks will be deducted if your code is not styled properly.

### Using the Makefile and Other Supplied Files

The Makefile provided in this assignment is used by a command in the CSIL machines called “make” to quickly compile your code. It is especially useful if you have multiple source files. To use it, type the following command in the prompt (make sure you are in the directory with all the files of Assignment 1):

```
$ make test1
```

The example above illustrates how Question 1 is compiled into an executable called “test1” when using the Makefile. Replace the “test1” with “test2”, “test3”, ...etc. for other questions. You can then run the executable by typing “./test1” to test your code for Question 1. If you make changes to your code, use the make command again. You can also use “make all” if you want to compile all 4 at once.

The test files (test1.c, test2.c, ...etc.) are provided in this assignment for you to test your code. Each typically contains a main function along with other tester functions and/or calls. You can modify them to further test your code, but do not submit these test files because we will be using our test files that are similar but not identical to grade your assignment. This makes sure that your code is not written to produce hard-coded output.

The header files (question1.h, question2.h, ...etc.) are there to make the compilation work. Ignore them and do not modify them. You also do not have to submit them.

### Submission

Submit **only the 4 source files** indicated above (question1.c, question2.c, question3.c, question4.c) by compressing them into a zip file (**do not** put them into a folder and zip them) and upload it to Canvas **by 11:59p Jan 29**. Name the zip file in this format: **<firstname\_lastname>\_<studentID>\_Assignment1.zip**.

For example, John\_Smith\_012345678\_Assignment1.zip

Assignment late penalty: 10% per calendar day (each 0 to 24 hour period past due), max 2 days late.

### Academic Honesty

It is expected that within this course, the highest standards of academic integrity will be maintained, in keeping with SFU’s Policy S10.01, “Code of Academic Integrity and Good Conduct.” In this class, collaboration is encouraged for in-class exercises and the team components of the assignments, as well as task preparation for group discussions. However, individual work should be completed by the person who submits it. Any work that is independent work of the submitter should be clearly cited to make its

source clear. All referenced work in reports and presentations must be appropriately cited, to include websites, as well as figures and graphs in presentations. If there are any questions whatsoever, feel free to contact the course instructor about any possible grey areas.

Some examples of unacceptable behavior:

- Handing in assignments/exercises that are not 100% your own work (in design, implementation, wording, etc.), without a clear/visible citation of the source.
- Using another student's work as a template or reference for completing your own work.
- Using any unpermitted resources during an exam.
- Looking at, or attempting to look at, another student's answer during an exam.
- Submitting work that has been submitted before, for any course at any institution.

All instances of academic dishonesty will be dealt with severely and according to SFU policy. This means that Student Services will be notified, and they will record the dishonesty in the student's file. Students are strongly encouraged to review SFU's Code of Academic Integrity and Good Conduct (S10.01) available online at: <http://www.sfu.ca/policies/gazette/student/s10-01.html>.