PROGRAMMING PROJECT 1
Due: 22 / 04 / 2022 – 23:30 (No late submission)

In this project, you are required to implement some procedures in MIPS assembly language. You will use a MIPS simulator (QTSPIM or MARS) to develop and test your code. There will be three questions in the project which are unrelated.

QUESTION 1. *(12 points)*

convertNumber (input_list, output_list, type)

Write a MIPS procedure that will convert the number given as character list (***input_list***) to the desired number format (according to the value of ***type*** argument). If ***type*** is equal to 1, the input character list represents an ASCII string of 2's complement binary digits. It should be converted into an integer and stored in $v0 register.

Example Run 1:
Input: 1101
Type: 1
Output: -3 (the value in $v0 register)
Example Run 2:
Input: 0101
Type: 1
Output: 5 (the value in $v0 register)
Example Run 3:
Input: 1111 1101
Type: 1
Output: -3 (the value in $v0 register)

If ***type*** is equal to 2; the input character list represents an ASCII string of 2's complement binary digits. It should be converted into base-16 digits represented by an ASCII string of digits and stored in ***output_list*** argument.

Example Run 4:
Input: 00111010
Type: 2
Output: 3A

QUESTION 2. *(12 points)*

Write a procedure in MIPS to print the sum of two rational numbers (fractions) as a rational number too. The following restrictions apply:

- The numbers must be entered by the user and printed by the program as **pairs of integers** separated by a proper delimiter (e.g. 5/8).
- The program must use **integer arithmetic only**.
- The result of the program must be a rational number in its **canonical (simplified)** form. For example, if you add (1/5 + 3/10) the result must be printed as 1/2 (not 25/50, which you can get by the following transformation: 1/5+3/10 = (1*10+3*5)/(5*10) = 25/50).
- The reduction to canonical form must be done by using the **Euclid's algorithm**.

An Example Run:
Input:
Enter the first numerator: 1
Enter the first denominator: 5
Enter the second numerator: 3
Enter the second denominator: 10

Output:
1/5+3/10=1/2

QUESTION 3. *(12 points)*

Write a MIPS procedure that takes an input sentence and prints the words in it by parsing with given set of characters. A *word* is a sequence of characters without any whitespaces and given set of punctuation marks. The only punctuation marks that you have to consider will be given as second input. Size of punctuation array can be at most 8 ASCII characters.

Example Run 1 :
Input:
Input text : !!!The first CSE-3038 project is due on 20-April-2022 (Wednesday).!!!
Parser characters: , . ! ? _ - ( )
Output:
The
first
CSE
3038
project
is
due
on
20
April
2022
Wednesday

Example Run 2 :
Input:
Input text : (Midterms start on 11-April-2022 for this semester.)
Parser characters: - ( )
Output:
Midterms
start
on
11
April
2022
for
this
semester.

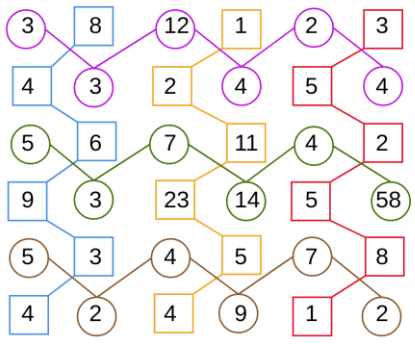
QUESTION 4. *(21 points)* Write a MIPS procedure to calculate following given mystery operation on a square matrix. Matrix dimension size will be always even and depending on their index id, elements will be multiplied by their cross neighbours either horizontally or vertically. Assuming that input matrix in size of NxN***, result will be two vectors with size of N/2***. Matrix dimension size can be any even number. Assuming that input matrix is 4x4:

$A_{11}\ A_{12}\ A_{13}\ A_{14}$
$A_{21}\ A_{22}\ A_{23}\ A_{24}$
$A_{31}\ A_{32}\ A_{33}\ A_{34}$
$A_{41}\ A_{42}\ A_{43}\ A_{44}$

Output of mystery operation would be:

$A_{11} \times A_{22} \times A_{13} \times A_{24}$ $\quad$ $A_{31} \times A_{42} \times A_{33} \times A_{44}$
$A_{12} \times A_{21} \times A_{32} \times A_{41}$ $\quad$ $A_{14} \times A_{23} \times A_{34} \times A_{43}$

Following 6x6 matrix is given as a color highlight of matching neighbours for each element.



Output for each multiplication of cross sets must be given as:

```
3456    341040  5040
20736   10120   1200
```

Input for this problem will be given as a string which must be parsed by space and converted into array of integers. Elements will be given in order of row and column as in the following example run. Then, depending on the number of elements in the array, you must assume both dimensions of the matrices are equal and the dimension size is square root of the array size.

Example Run 1:

Input:
3 8 12 1 2 3 4 3 2 4 5 4 5 6 7 11 4 2 9 3 23 14 5 58 5 3 4 5 7 8 4 2 4 9 1 2

Output:
3456 341040 5040
20736 10120 1200

MENU *(8 points):* Your program should support a *Menu* including all questions above. A sample execution scenario given below:

```
Welcome to our MIPS project!
Main Menu:
1. Base Converter
2. Add Rational Number
3. Text Parser
4. Mystery Matrix Operation
5. Exit
Please select an option: 1
```

These options must be printed inside a loop until "Exit" option is selected.
When the user select option 1, you should print the followings:

```
Input: 1101
Type: 1
Output: -3
```

```
Main Menu:
1. Base Converter
2. Add Rational Number
3. Text Parser
4. Mystery Matrix Operation
5. Exit
Please select an option: 2
Enter the first numerator: 1
Enter the first denominator: 5
Enter the second numerator: 3
Enter the second denominator: 10

Output:
1/5+3/10=1/2
```

```
Main Menu:
1. Base Converter
2. Add Rational Number
3. Text Parser
4. Mystery Matrix Operation
5. Exit
Please select an option:  3
Input text :
!!!The first CSE-3038 project is due on 20-April-2022 (Wednesday).!!!
Parser characters: , . ! ? _ - ( )

Output:
The
first
CSE
3038
project
is
due
on
20
April
2022
Wednesday
```

```
Main Menu:
1. Base Converter
2. Add Rational Number
3. Text Parser
4. Mystery Matrix Operation
5. Exit
Please select an option: 4
Input:3 8 12 1 2 3 4 3 2 4 5 4 5 6 7 11 4 2 9 3 23 14 5 58 5
3 4 5 7 8 4 2 4 9 1 2

Output:
3456      341040    5040
20736     10120     1200


Main Menu:
1. Base Converter
2. Add Rational Number
3. Text Parser
4. Mystery Matrix Operation
5. Exit
Please select an option: 1
Please select an option: 5
Program ends. Bye :)
```

Assumptions and Requirements

- The arguments to the procedures are stored in $a registers; i.e., the first one is in $a0, the second one is in $a1, and so on.
- Only valid arguments are passed into the procedures. Therefore, you do not need to check the arguments for their validity.
- When you invoke a procedure, the values of all $a registers should be preserved. Their values should be same at the end of the procedure call as they were at the time of call.
- You have to use QtSpim or MARS simulator in your implementation. Any other simulator is not allowed.
- You are required to submit a minimum 2-page report (***5 points***) explaining implementation details of your project. Your report will have four parts (one for each question) and it will also include screenshot of your sample runs, as well.
- You should submit a fully commented source code that includes details of your implementation. Note that the name of the file should include surnames of the group members. (ex: surname1_surname2_surname3_surname4.s)
- Zip your fully commented source code file and the project report into a single file and submit the zip file via Canvas.

General Policies for the Project

- *You have to work in groups of 3 or 4*. You will select your partners and partners will not be changed throughout the semester. *It is not acceptable of a partner team to work with other teams.*
- *A portion of your project grade will be set with a Project Quiz.* Note that if you do not submit the project, you will not attend the Project Quiz.
- Copying (partially or full) solutions from other students is a form of cheating. Copying (partially or full) solutions from Web including Github (and similar sites) is another form of cheating. It is NOT acceptable to copy (or start your) solutions from Web. **In case of any forms of cheating or copying among the groups, the penalties will be severe. Both Giver and Receiver are equally culpable and suffer equal penalties!!!**
- No late submission will be accepted!