

## Point value

This assignment is worth 20 points. The points will be distributed as follows:

Criteria	Value
Style/Comments	3
Control Algorithm as Described Below	8
User Input Parsing	3
Rational Number Output	3
Use of Procedures	1
Use of Macros	1
Use of Structs	1

## Program Objectives

an ability to design software and/or hardware experiments, as well as to analyze and interpret data (ABET c);

## Delivery Method

This is a multimodule program, so bundle your files as a single zip file named pa8.zip. Send this file as an attachment in an email you send to the class account. Put the assignment number in the subject field and your name in the body of the email. Submission should be as follows:

1. *pa8.asm*. This is the driver file with the startup procedure MAIN.
2. *pa8functions.inc*. This is the file containing the prototypes of your helper procedures.
3. *pa8functions.asm*. This is the implementation file for your helper procedures.
4. *pa8structs.inc*. This is the file that contains your structs.
5. *pa8macros.inc*. This is the file that contains your macros.
6. Include your entire solution and project files in a single zip file.
7. The solution should be compilable when I extract it.

## Problem

Perform basic arithmetic on rational numbers. A rational number, which is a number that can be represented by the ratio of two integers, will be represented by a struct that has two decimal members, each of type SWORD.

The program must execute as follows:

1. Prompt user for a rational number (two inputs, must be comma separated integers). We will call this input #1.
2. Display rational number input #1 as a tuple, as well as an real expression with up to 10 decimal places. For example, 4/3 would display as “[4, 3]: 1.3333333333”.
3. Prompt the user to select a standard arithmetic operators (+, -, \* and /) on the rational numbers, as well as the option ‘e’ to exit.
4. If the user input the character ‘e’ or ‘E’, then goto step 8. Otherwise, if the input is invalid, notify user and goto step 3.
5. Prompt user for an additional rational number. We will call this input #2.
6. Perform the arithmetic operator on inputs #1 and #2. The result now becomes input #1.
7. Goto step 2.
8. Exit program.

**Extra Credit (+5):**

- Allow the user to input either comma separated integers (3,10), or a decimal value (0.3)

**Notes**

- 1. The resultant rational of each operation must be reduced.**
2. The rational number input by user must be valid. For example, “**1, 0**” would be invalid.
3. Main procedure must be concise and consist of procedure/macro calls defined in other files.
4. Do not use directives such as .IF, .WHILE and the like.
5. You may use Irvine32 procedures.