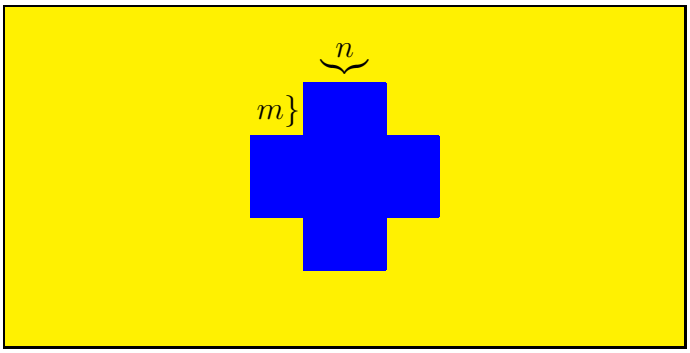
CSCI 260 – Project

Due: Mar 23 11:59 PM

The goal of this homework is to write a non-trivial MIPS program, and execute it using a MIPS simulator (MARS). It is an **individual** assignment and no collaboration on coding is allowed (other than general discussions).

Assignment

Write MIPS code to draw the following shape on the bitmap display supplied as part of MARS:



The center (blue) figure should be centered (in the yellow frame). The variables m and n refer to the length of the corresponding dimensions and should **not** be drawn (the figure is completely symmetric). Additional notes:

- If m or n is odd, you should increment it to the next even number, so that a perfect centering is possible.
- If the parameters are not realizable (*e.g.,*the combination of m and n is too big), your code should display nothing.

Submission Instructions

Please submit a single .asm file using blackboard. The file should contain your name (not password) in header comments, and follow normal MIPS conventions. Your program should have variables named m and n in the data segment, and will be tested on multiple values of these variables. You may use any of the instructions and pseudo-instructions we covered in class excluding multiplication and division (you're probably thinking the wrong way if you think you need these).

Writing The Program

Accessing the Bitmap Display

In your program, your data segment should start with:

```
.data
frameBuffer:  .space  0x80000  # 512 wide X 256 high pixels
```

The framebuffer is essentially memory-mapped I/O storing a 512-pixel×256-pixel×32-bit image in row-major order. For example, MEM[frameBuffer+4] would contain the pixel at row 0, column 1. You may add additional variables to the data segment (such as m and n) *after* the frameBuffer.

Each **pixel** is a 32-bit value consisting of 8-bits each for the red, blue, and green components in bits 23:16, 15:8, and 7:0 respectively (the upper 8 bits are ignored). For example, the value 0x0000FF00 would correspond to bright green.

Using the MARS Bitmap Display

Please see the other guide posted on bb for how to use MARS. After reading that, you will need to do a few additional things to use the bitmap display as follows:

1. Select `Tools`→`Bitmap Display`
2. Click the `Connect to MIPS` button
3. Follow the instructions on the MARS guide to assemble and run your program.

The data segment defaults to starting at 0x10010000.

Sample code: The following snippet (at the beginning of the text segment) draws a 3-pixel green line segment somewhere near the top center of the display (assuming you have the data segment from above):

```
.text
drawLine:  la       $t1,frameBuffer
           li       $t3,0x0000FF00   #$t3← green
           sw       $t3,56300($t1)
           sw       $t3,56304($t1)
           sw       $t3,56308($t1)
           li       $v0,10           # exit code
           syscall
```