我根據matrix template裡的geeksforgeeks網址中第三個方法—Tabulation去實作assembly code，將最後算出來的答案存進x31裡。

1. 實作想法與細節:

我一開始是用網址中的第一個方法—直覺、無記憶化的遞迴版本下去實作，但我在處理assembly的遞迴上遇到了相當大的困難，雖然HW2的7.也是要將遞迴C code轉成riscv，但我覺得那題跟這次的難度果然還是有一定的差距，所以我後來就選擇第三個方法去實作了。

然而一開始選擇第三個方法去寫assembly的時候，我不知道該怎麼allocate二維陣列，這時突然想到m[n][n]可以把它想成m[n\*n]的一維陣列(如下圖的紅框部分)，所以存取m[i][j]就是去m[ i\*n + j ]的地方取值就好 (後來發現55行其實是多餘的)。

藍框的部分是一開始不知道該怎麼個別存取Matrix\_array中的個別元素，後來google發現可以藉由la將Matrix\_array的位址存進x10，再透過lw,offset去存取個別元素就好。

一張含有 文字 的圖片

自動產生的描述一張含有 文字 的圖片

自動產生的描述

1. Hazards:

Type (1)(2):

一張含有 文字 的圖片

自動產生的描述

Type 1 hazard發生在131,132行的x24，如下圖黃線所示，mul x24,x24,x25在執行完EX階段後會將x24 \* x25的結果forward到位於EX階段mul x24,x24,x26的rs1 = x24。

一張含有 文字, 電子用品, 計分板 的圖片

自動產生的描述

Type 2 hazard發生在132,136行的x24，如下圖黃線所示，mul在執行完MEM階段後會將x24的結果forward到EX階段add的rs2 = x24。

一張含有 文字, 電子用品, 計分板, 電腦 的圖片

自動產生的描述

Type(3):

一張含有 文字, 室外, 黑色 的圖片

自動產生的描述

lw的下一行就直接使用x25，發生了type 3的hazard。當lw做完EX階段後，Ripes偵測到了hazard就將bge的指令stall 1 cycle，這樣lw才能在做完MEM階段後將memory中0(x21) = x25的值forward到位於EX階段bge的rs2 = x25中，如下圖黃線所示。

一張含有 文字, 電子用品, 計分板, 電腦 的圖片

自動產生的描述

Type(4):

一張含有 文字 的圖片

自動產生的描述

Type 4 hazard發生在127, 132行的x26，因為lw, mul相差兩個cycle所以不需要stall，lw會在MEM階段做完後將memory中0(x21) = x26的值forward到位在EX階段mul的rs2 = x26，如下圖黃線所示，。

一張含有 文字, 電子用品, 電腦 的圖片

自動產生的描述

Type(5):

一張含有 文字, 螢幕, 計分板 的圖片

自動產生的描述

Type 5 hazard發生在上圖的紅框與藍框部分。

下圖是x5已經大於等於x11時所截的圖，這時候的processor仍然繼續將init block裡藍框的code pipeline進processor的IF,ID stage，但事實上因為init block的條件(x5 < x11)已不再成立，所以processor就得如圖二所示，將已經不符合條件的add,mul指令flush掉，再將init\_exit的addi pipeline到processor裡。

一張含有 文字, 電子用品, 電腦, 計分板 的圖片

自動產生的描述

(圖一)

一張含有 文字, 電子用品 的圖片

自動產生的描述

(圖二)