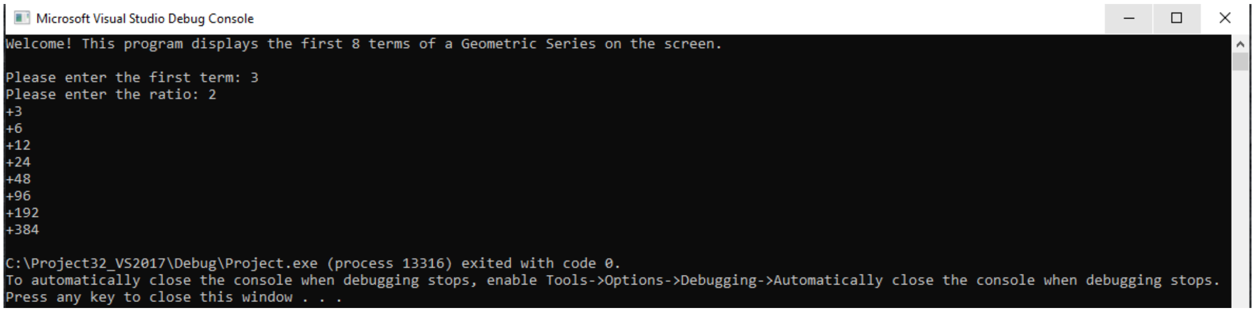
Comp 3350: Computer Organization & Assembly Language

HW # 9: Theme: Advanced Procedures, Stack Parameters, Locals and BCD

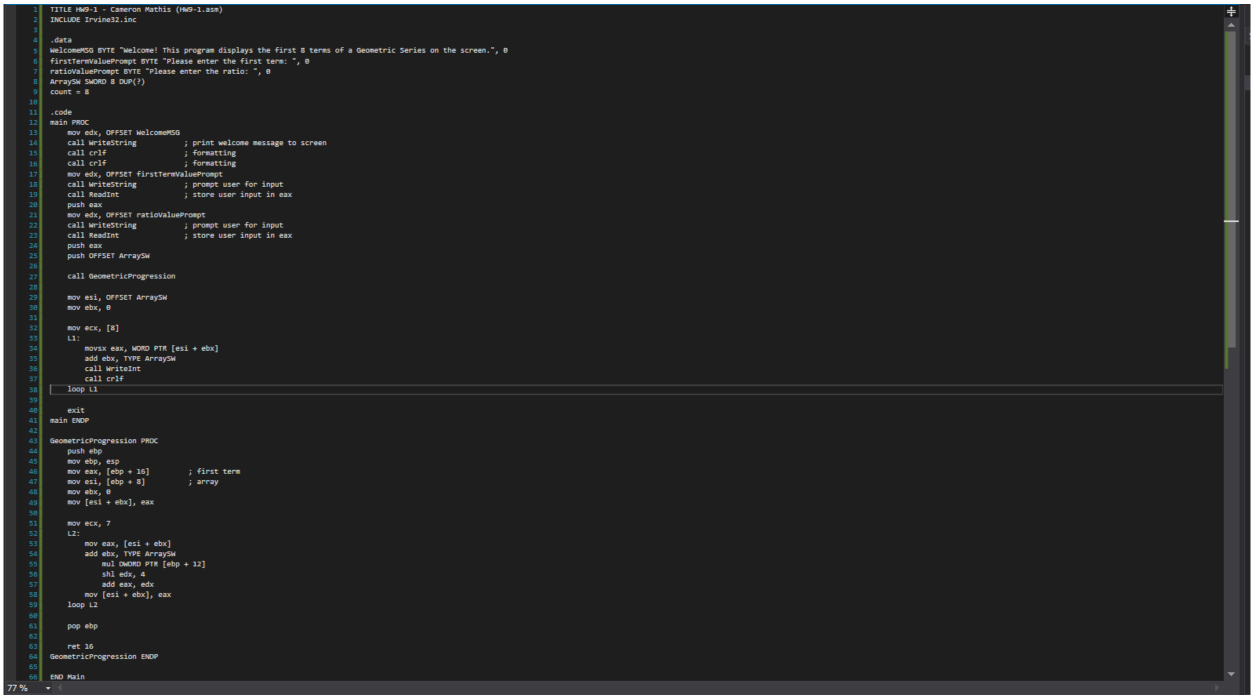
(All main questions carry equal weight. Credit awarded to only those answers for which work has been shown.)

1. Write a procedure named *Geometric Progression* that fills an array of eight (8) numbers with the Geometric series. The procedure receives three arguments: the first is the offset of an array, the second is the first term and the third is the ratio. The first argument is passed by value and the others by reference. In the main program, you should set the parameters and print the series. Please run your program with several different first term and ratios.

   Please embed your code into your homework solution along with a screen shot of the run of the program.



```
Welcome! This program displays the first 8 terms of a Geometric Series on the screen.

Please enter the first term: 3
Please enter the ratio: 2
+3
+6
+12
+24
+48
+96
+192
+384

C:\Project32_VS2017\Debug\Project.exe (process 13316) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```



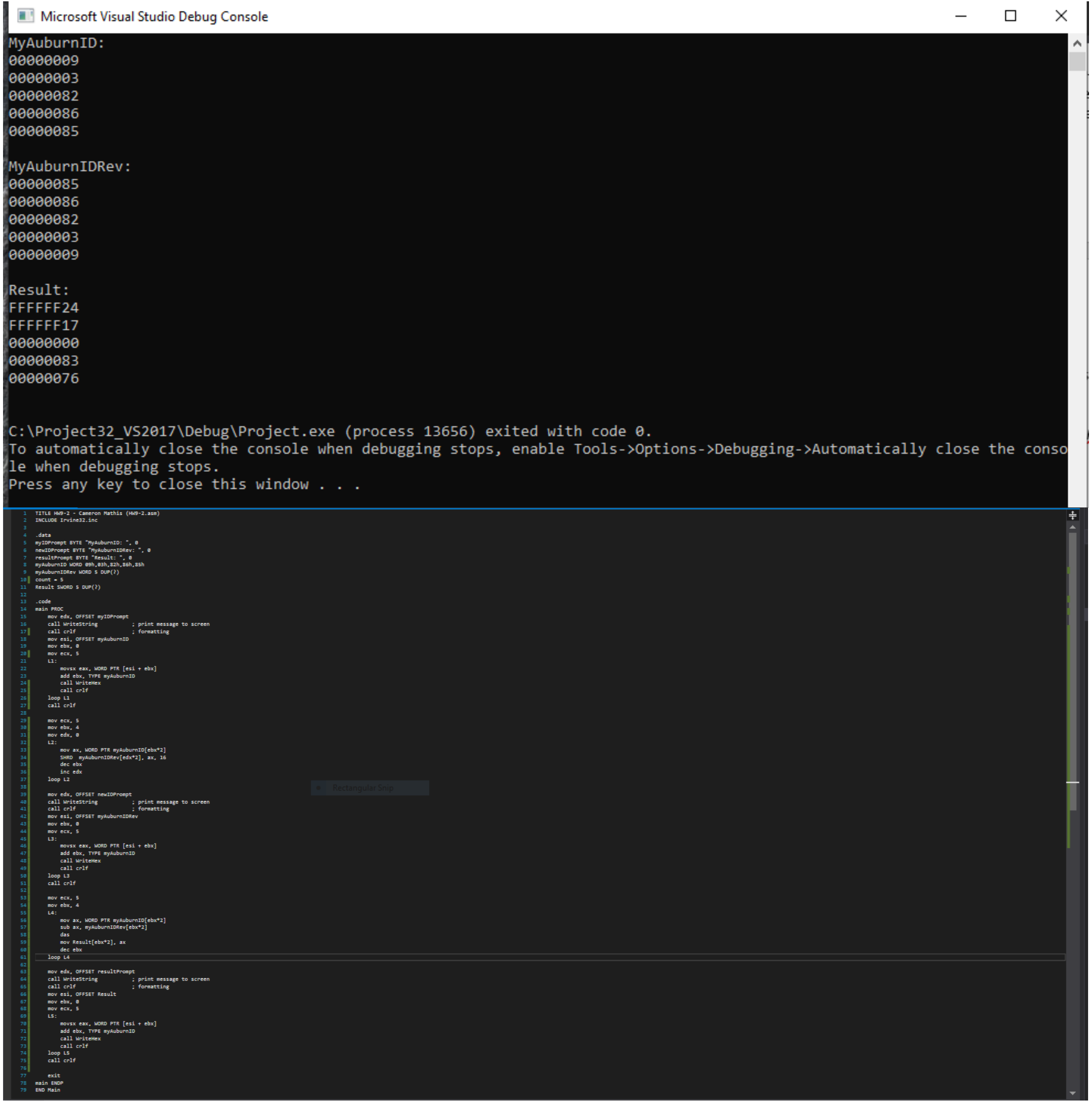
```
TITLE HW9-1 - Cameron Mathis (HW9-1.asm)
INCLUDE Irvine32.inc

.data
WelcomeMSG BYTE "Welcome! This program displays the first 8 terms of a Geometric Series on the screen.", 0
firstTermValuePrompt BYTE "Please enter the first term: ", 0
ratioValuePrompt BYTE "Please enter the ratio: ", 0
ArraySW SWORD 8 DUP(?)
count = 8

.code
main PROC
    mov edx, OFFSET WelcomeMSG
    call WriteString            ; print welcome message to screen
    call crlf                   ; formatting
    call crlf                   ; formatting
    mov edx, OFFSET firstTermValuePrompt
    call WriteString            ; prompt user for input
    call ReadInt                ; store user input in eax
    push eax
    mov edx, OFFSET ratioValuePrompt
    call WriteString            ; prompt user for input
    call ReadInt                ; store user input in eax
    push eax
    push OFFSET ArraySW

    call GeometricProgression

    mov esi, OFFSET ArraySW
    mov ebx, 0

    mov ecx, [8]
    L1:
        movsx eax, WORD PTR [esi + ebx]
        add ebx, TYPE ArraySW
        call WriteInt
        call crlf
    loop L1

    exit
main ENDP

GeometricProgression PROC
    push ebp
    mov ebp, esp
    mov eax, [ebp + 16]         ; first term
    mov esi, [ebp + 8]          ; array
    mov ebx, 0
    mov [esi + ebx], eax

    mov ecx, 7
    L2:
        mov eax, [esi + ebx]
        add ebx, TYPE ArraySW
            mul DWORD PTR [ebp + 12]
            shl edx, 4
            add eax, edx
        mov [esi + ebx], eax
    loop L2

    pop ebp

    ret 16
GeometricProgression ENDP

END Main
```

2. Draft a program that subtracts one BCD number from another (10-digits each). The first BCD number is stored in an array named *myAuburnID*, and the second in an array named *myAurbunIdRev*. The first number is your actual Auburn ID (with a prefix single zero digit and the remaining digits as the 9-digits of your *Auburn ID*); the second is the value of *MyAuburnId*

written backwards. Your program should do the following:

1) Use shifts/rotates using *myAuburnID* to fill the array *myAuburnIdRev*
2) Display contents of the memory locations in question
3) Subtract *myAuburnIDRev from myAurbunId* using BCD arithmetic
4) Store the sum in a variable named *Result*, and
5) Display contents of memory post execution.

Please embed your code into your homework solution along with a screen shot post execution.

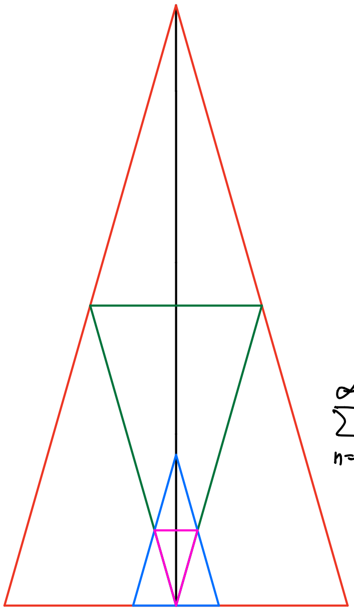


```
MyAuburnID:
00000009
00000003
00000082
00000086
00000085

MyAuburnIDRev:
00000085
00000086
00000082
00000003
00000009

Result:
FFFFFF24
FFFFFF17
00000000
00000083
00000076


C:\Project32_VS2017\Debug\Project.exe (process 13656) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

```
TITLE HW9-2 - Cameron Mathis (HW9-2.asm)
INCLUDE Irvine32.inc

.data
myIDPrompt BYTE "MyAuburnID: ", 0
newIDPrompt BYTE "MyAuburnIDRev: ", 0
resultPrompt BYTE "Result: ", 0
myAuburnID WORD 09h,03h,82h,86h,85h
myAuburnIDRev WORD 5 DUP(?)
count = 5
Result SWORD 5 DUP(?)

.code
main PROC
    mov edx, OFFSET myIDPrompt
    call WriteString            ; print message to screen
    call crlf                   ; formatting
    mov esi, OFFSET myAuburnID
    mov ebx, 0
    mov ecx, 5
    L1:
        movsx eax, WORD PTR [esi + ebx]
        add ebx, TYPE myAuburnID
        call WriteHex
        call crlf
    loop L1
    call crlf

    mov ecx, 5
    mov ebx, 4
    mov edx, 0
    L2:
        mov ax, WORD PTR myAuburnID[ebx*2]
        SHRD  myAuburnIDRev[edx*2], ax, 16
        dec ebx
        inc edx
    loop L2

    mov edx, OFFSET newIDPrompt
    call WriteString            ; print message to screen
    call crlf                   ; formatting
    mov esi, OFFSET myAuburnIDRev
    mov ebx, 0
    mov ecx, 5
    L3:
        movsx eax, WORD PTR [esi + ebx]
        add ebx, TYPE myAuburnID
        call WriteHex
        call crlf
    loop L3
    call crlf

    mov ecx, 5
    mov ebx, 4
    L4:
        mov ax, WORD PTR myAuburnID[ebx*2]
        sub ax, myAuburnIDRev[ebx*2]
        das
        mov Result[ebx*2], ax
        dec ebx
    loop L4

    mov edx, OFFSET resultPrompt
    call WriteString            ; print message to screen
    call crlf                   ; formatting
    mov esi, OFFSET Result
    mov ebx, 0
    mov ecx, 5
    L5:
        movsx eax, WORD PTR [esi + ebx]
        add ebx, TYPE myAuburnID
        call WriteHex
        call crlf
    loop L5
    call crlf

    exit
main ENDP
END Main
```

3. Consider an isosceles triangle A with base 8 and height 14. Consider another triangle B formed using vertices which are the center of the sides of triangle A. Consider another triangle C whose

vertices are similarly formed from B. Repeat this process ad infinitum. Express the sum of the areas of all such triangles using a series and its closed form sum. Compute the areas *(a)* by using only the first two terms of the series and *(b)* by using the closed form of the series sum. Write a program to find the sums and use shifts to compute. What is the difference in the two computed sums?



$A_1 = \frac{1}{2}(8)(14) = 56 = \frac{112}{2}$

$A_2 = \frac{1}{2}(4)(7) = 14 = \frac{112}{8}$

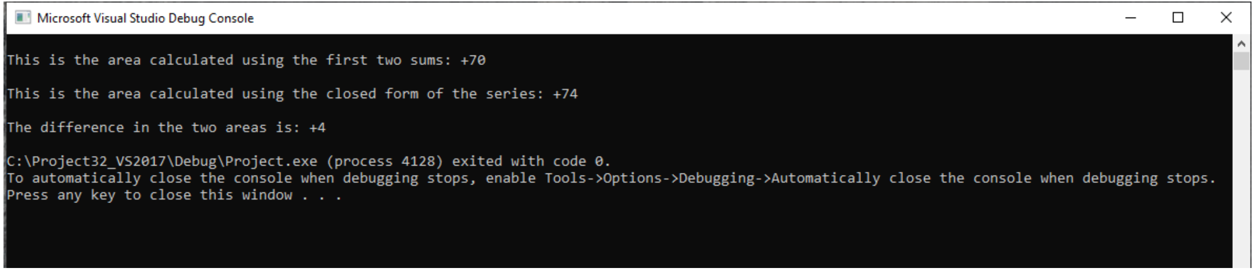
$A_3 = \frac{1}{2}(2)(3.5) = 3.5 = \frac{112}{32}$

$A_4 = \frac{1}{2}(1)(1.75) = .875 = \frac{112}{64}$

$A = \frac{112}{\frac{2^{2n}}{2}} = \frac{224}{2^{2n}} = \frac{224}{(2^2)^n} = \frac{224}{4^n}$

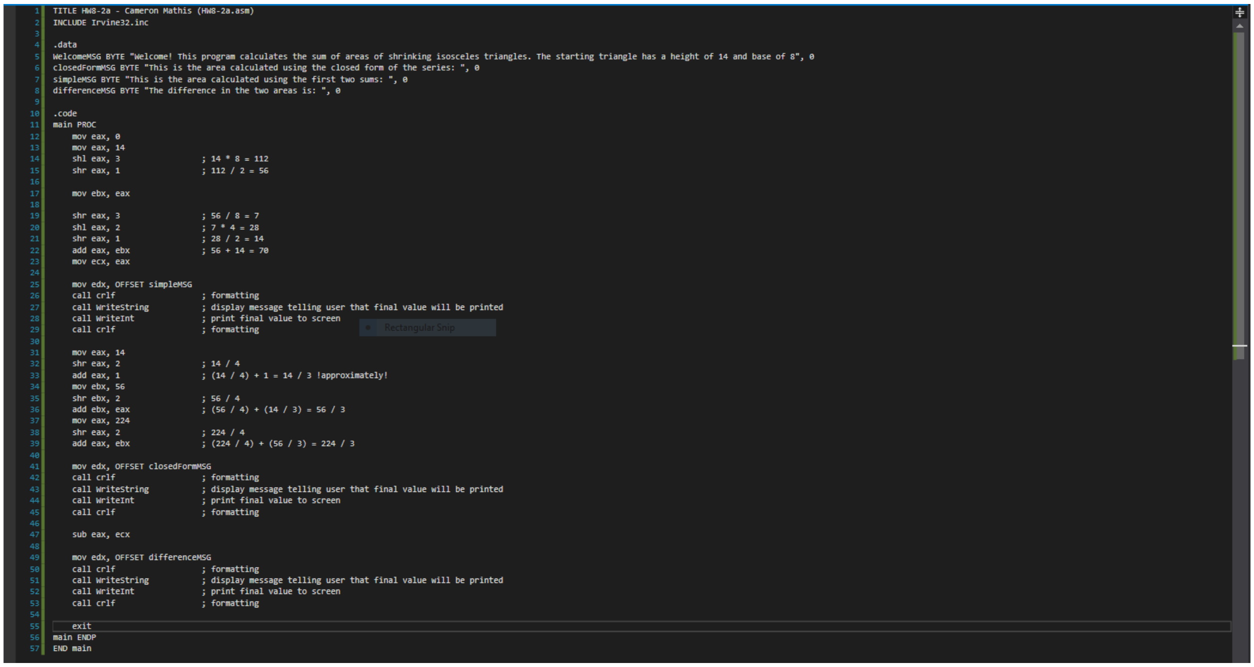
$$\sum_{n=1}^{\infty} \frac{224}{4^n} = \sum_{n=1}^{\infty} 224\left(\frac{1}{4}\right)^n = 224\sum_{n=0}^{\infty}\left(\frac{1}{4}\right)^n - \left(\frac{1}{4}\right)^0 = 224\left(\frac{1}{1-\frac{1}{4}} - 1\right) = 224\left(\frac{4}{3} - 1\right) = \frac{224}{3}$$

Please embed your code into your homework solution along with a screen shot post execution.



```
This is the area calculated using the first two sums: +70

This is the area calculated using the closed form of the series: +74

The difference in the two areas is: +4

C:\Project32_VS2017\Debug\Project.exe (process 4128) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

The difference in the two sums is $4.\overline{666}$, but we have not discussed how to deal with values between 0 and 1 so it was rounded to 4 in my program. The reason the closed form is slightly greater is because it is more accurate.



```
TITLE HW8-2a - Cameron Mathis (HW8-2a.asm)
INCLUDE Irvine32.inc

.data
WelcomeMSG BYTE "Welcome! This program calculates the sum of areas of shrinking isosceles triangles. The starting triangle has a height of 14 and base of 8", 0
closedFormMSG BYTE "This is the area calculated using the closed form of the series: ", 0
simpleMSG BYTE "This is the area calculated using the first two sums: ", 0
differenceMSG BYTE "The difference in the two areas is: ", 0

.code
main PROC
    mov eax, 0
    mov eax, 14
    shl eax, 3                 ; 14 * 8 = 112
    shr eax, 1                 ; 112 / 2 = 56

    mov ebx, eax

    shr eax, 3                 ; 56 / 8 = 7
    shl eax, 2                 ; 7 * 4 = 28
    shr eax, 1                 ; 28 / 2 = 14
    add eax, ebx               ; 56 + 14 = 70
    mov ecx, eax

    mov edx, OFFSET simpleMSG
    call crlf                  ; formatting
    call WriteString           ; display message telling user that final value will be printed
    call WriteInt              ; print final value to screen
    call crlf                  ; formatting

    mov eax, 14
    shr eax, 2                 ; 14 / 4
    add eax, 1                 ; (14 / 4) + 1 = 14 / 3 !approximately!
    mov ebx, 56
    shr ebx, 2                 ; 56 / 4
    add ebx, eax               ; (56 / 4) + (14 / 3) = 56 / 3
    mov eax, 224
    shr eax, 2                 ; 224 / 4
    add eax, ebx               ; (224 / 4) + (56 / 3) = 224 / 3

    mov edx, OFFSET closedFormMSG
    call crlf                  ; formatting
    call WriteString           ; display message telling user that final value will be printed
    call WriteInt              ; print final value to screen
    call crlf                  ; formatting

    sub eax, ecx

    mov edx, OFFSET differenceMSG
    call crlf                  ; formatting
    call WriteString           ; display message telling user that final value will be printed
    call WriteInt              ; print final value to screen
    call crlf                  ; formatting

    exit
main ENDP
END main
```