COSC2406 F18 – Assembly Language Programming

Assignment 6 - Chapters 7 & 8

Due: November 17th, 2018 by 11:55pm

NOTE: Only single operand multiplication and division instructions are allowed.

Q1: [60] Create a program with an array of 10 SWORDS initialized with all -100. Then present the user with the following menu of options:

1 – Populate the array with random numbers
2 – Multiply the array with a user provided multiplier
3 – Divide the array with a user provided divisor
4 – Print the array
0 - Exit

For option #1, the random numbers will be in the range of -1500 to + 2500. Make a procedure called **populateRandomNum** which accepts **two unnamed stack parameters** of type dword – one for the offset of the sword array to be filled with random numbers; and one for the number of elements in the array. The procedure will return nothing. At least one local variable, created on the stack **without** the use of the LOCAL directive, must be used by the procedure. **You CANNOT use ENTER or LEAVE for this procedure.**

For options #2, ask the user to provide the multiplier for the required operation. Use a procedure which accepts the offset of the SWORD array to be multiplied and the multiplier as **NAMED stack parameters**. Push the parameters onto the stack and use CALL to execute the procedure. **The procedure MUST use ENTER and LEAVE.**

For option #3, ask the user to provide the divisor for the required operation. Use a procedure which accepts the a single SWORD value in the array and the divisor as **unnamed stack parameters**. The result of the division will be returned in EAX and used in a loop in the main procedure to update every value within the SWORD array. Push the parameters onto the stack and use CALL to execute the procedure. The stack parameters must be removed by the procedure when it returns.

For option #4, the printed values should be signed integers, comma separated, with square [ ] brackets around the list. A procedure must be used to complete this task and parameters must be passed using named parameters (with INVOKE).

Additional Requirements:

A. The menu must loop after each option (except for exit). Invalid choices should not break the program and should result in the menu being shown again.

B. ***Only option#4 will print the array values – but every other procedure should provide some appropriate message when complete.***

C. All procedures must preserve register and flag values whenever possible (32 bit versions).

D. Any procedure returning an answer must do so in either the EAX register or using an in-out stack parameter.

E. No procedure can directly reference an array (directly reference means using the name of the array). Only use indirect operands with the offset and size of the array being passed as a **parameter on the stack**.

F. Fully comment the code including header comments, comments at the start of each procedure, and line codes neatly in a right column. (Penalty of up to 25% for not doing this)

G. All INVOKE operations must have matching PROTO declarations.

Grading: Options 1 & 4 – 5 marks each; Options 2 & 3 – 15 marks each; menu and other structure 8 marks; demonstration and explanation 12 marks.

For this problem, a right-side column of comments must be included to explain the logic of your code. Writing the logic in Java or pseudo-code is acceptable. When I read your comments, I should clearly see your algorithm. Also, there must be proper comments at the start of every procedure as shown on pages 146-47, and comments with your name, course code, date, assignment/question number, and problem description at the top of every program. Marks will be deducted for missing commenting. ***Failure to comment the code properly or to format like the code in the textbook will result in a penalty of up to 25%.***

Demonstrate and explain your working programs to the TA, Glenn Driver (gdriver@algomau.ca) by Friday Nov 23th. The demonstration and explanation of your code will represent 25% of the assignment grade – the other 75% of the grade will come from grading the code and how well the programs meet the specification and requirements of the question.

IMPORTANT:

a) ***If you do not demonstrate your programs to the TA by the end of the demonstration date, you will receive a grade of zero.***

b) ***If you are unable to explain most or all of your code and the logic of your programs, then the maximum grade possible will be 50%.***