message "Character = x", using the following data definitions.

```
msg1    db      "Character =  ", 0
char1   db      ?
```

5. (4 points) Write the Pentium code to multiply the doubleword variable var3 by 1000 and store the result (the product) in the EBX register. You do not need to check for overflow (even though it is possible).

6. (4 points) What are two advantages of using registers as operands rather than memory (besides the fact that you have to use registers in the Pentium architecture)?

7. (address modes)

    1. (4 points) For the instruction

        ```
        mov     EAX, dword ptr [EBX][ESI]
        ```

        How many bits are needed in the instruction to specify all of the operands? (Note: assume that specifying a register requires 4 bits, specifying a memory address requires 32 bits, and specifying a constant requires 8 bits.)

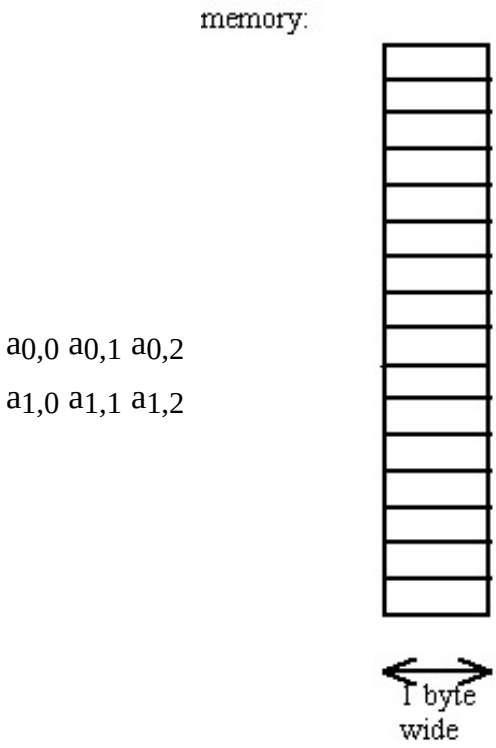
    2. (2 points) How many memory accesses (reads or writes) are needed by this instruction, not counting the fetch of the instruction?

8. (stacks)

    1. (2 points)Show the statement to define a stack with 256 bytes.

    2. (4 points)Suppose the address of the bottom-most byte (highest-numbered address) of this stack was y. What is the address of the top-most (lowest-numbered address) byte of the stack, and to what is the ESP initialized before your program begins execution?

9. (4 points)The elements of a 2x3 array are as follows (each element is one byte long). Show how these elements are stored in memory using row-major ordering.



10. (4 points)Show the Pentium code to check whether unsigned divide by the doubleword divisor var4 will result in overflow or not. (Note: assume the dividend has already been copied where it needs to be to use the "div" instruction.)

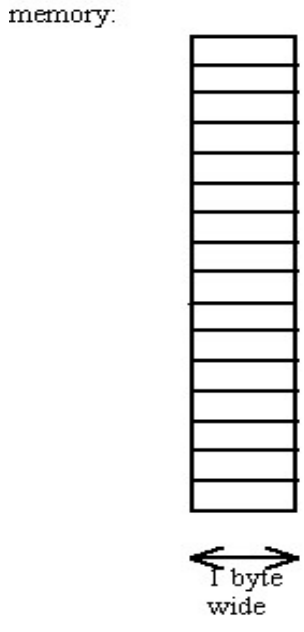
11. (4 points) EBX is used as a pointer to a doubleword operand in memory. We want to copy the value of the least significant byte of this operand to the AL register. Show the Pentium code to do so. (Note: You do not need to change the value in EBX; assume it already points to the doubleword.)

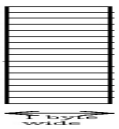
12. (8 points) Write the Pentium code to copy the value of the 17th element of a doubleword array "myarray" (i.e., myarray[16]) into the EAX register.

13. (8 points) "records" is 10x20 2-D array of records (structures) each of which is 7 bytes long. What is the offset to the 5th byte of the record in the 4th row and the 7th column of this array, relative to the start of the array?

14. (10 points) The ESP register is initialized to the value 0000101Fh. Show the value of the ESP register, and the contents of the stack, after executing the following code sequence. (Note: you only need to show the final result, not the intermediate values after each instruction.)

```
push    33FF33FFh
push    55445566h
push    22881100h
pop     EAX
```





15. (8 points) Write the Pentium code to check whether the stack is empty before popping a value from it, and jump to a label called "stackempty" if it is. Include any code needed to initialize variables, and assume all pushes and pops are of doublewords only.

16. (12 points) Write the Pentium code to copy the contents of the byte array str1 to to the byte array str2. Stop copying when a null character (equal to 0) is encountered in array str1. Show the data definition of any variables other than str1 and str2 that you use.

17. (12 points) Using only the data definition shown below and any registers you wish, write the code to sum the values of the first 100 bytes in array2. The result (the sum) should be in register EDX.

```
.data
array2  db      25, 49, -53, 68, -13, .... ; array is at least 100 bytes long
```

End of Exam