

EEAP 282

EXAM #3

November 17, 1997

NAME: _____

CWRUnet ID: _____

IMPORTANT INFORMATION:

Exam is closed book, closed notes. Only the M68000 Programmer's Reference Manual and / or Programming Reference Card are allowed to be used. NOT ALL PROBLEMS COUNT THE SAME.

Problem	Score	Possible
1	<input type="text"/>	15
2	<input type="text"/>	10
3	<input type="text"/>	10
4	<input type="text"/>	20
5	<input type="text"/>	25
6	<input type="text"/>	20

TOTAL SCORE 100

Hint:

divisor quotient
 dividend

Name: _____

CWRUnet ID: _____

1. The following code is executed:

```
LSR.W      #1, D0
ROR.W      #1, D1
MOVEM.L    D0-D1, - (A0)
```

Memory is initially as given below

```
$1100      [ $00 ]
$1101      [ $A4 ]
$1102      [ $FF ]
$1103      [ $A2 ]
$1104      [ $FE ]
$1105      [ $00 ]
$1106      [ $A4 ]
$1107      [ $A2 ]
$1108      [ $EC ]
```

You may further assume

(A0)=\$00001108

(D0)=\$CD235A3E

(D1)=\$BF802928

before the above code is executed. What are D0, D1, A0 and the memory contents after the code is executed?

(D0) = _____

(D1) = _____

(A0) = _____

Name: _____

CWRUnet ID: _____

2. What is in D0 and D1 after the following program is executed?

```
        MOVE.L    #$0A00FFFF, D0
        MOVEQ.L   #31, D1
NB:     ASL.L     #1, D0
        DECS     D1, NB
        TRAP     #0
```

(D0) = _____

(D1) = _____

Name: _____

CWRU net ID: _____

3. What is in D5 after executing the following program fragment ?

```
ORG      $3000
MOVE.W   #$FFB3, D4
MOVE.L   #$109, D5
DIVS     D4, D5
SWAP     D5
```

(D5) = _____

Name: _____

CWRUnet ID: _____

4. What are the contents of the 10 words in memory beginning at \$4600? The emphasis upon you to understand the algorithm. A simple flow chart or pseudocode would be very useful, especially for partial credit.

```
FIB EQU      $4600

      ORG      $4000
      MOVE.L   #$11,D0      ;SET THE COUNTER TO 17
      LEA     FIB,A1
      CLR.W   D1            ;D1=0
      MOVEQ.W #1,D2        ;D2=1
      MOVE.W  D1,(A1)+
      MOVE.W  D2,(A1)+

NXT: JSR     SBR
      MOVE.W  D2,D1
      MOVE.W  D3,D2
      MOVE.W  D3,(A1)+
      DBF     D0,NXT

      BRA     DONE

SBR:  ADD.W   D2,D1
      MOVE.W  D1,D3
      RTS

DONE END
```

Name: _____

CWRUnet ID: _____

5. A student has decided to use in line coding of data to pass parameters to a subroutine and the stack to return a single word length result. The main program shown below calls the subroutine SUBR. The stack pointer is initially at \$8000. Answer the following questions:

(a) What is on the stack when the PC is at the label INST? Explicitly show all stack contents AND addresses.

```

                                [      ]
                                [      ]
                                [      ]
                                [      ]
original SP-->                 [      ]
                                [      ]
                                [      ]
                                [      ]
main   ORG                      $6000
        MOVE.W                  #6,D1
        MOVE.W                  #5,D2
        ADD                     D1,D2
JSR    SUBR
A      DC.L                      4
B      DC.W                      2
C      DS.W                      1
* Your subroutine should return to the following instruction.
* and pop a word length result off the stack
DOIT   MOVE.W                   (SP)+,C
        END                     main
```

```
SUBR   MOVEM.L                  D1/D3,-(SP)
```

*(b) Write instructions to put A into D1 and B into D2.
*You are NOT allowed to use the symbols A and B in your code.

```
INST   MOVE.L                   #1,D3
        MULS                     D1,D3           ;answer in D3
```

*(c) Write instructions to put word length result in D3 onto stack such that it can be popped off stack after subroutine return at DOIT.

*

```
RTS
```

Name: _____

CWRUnet ID: _____

6. A subroutine SUB6 is called with parameters passed and returned on the stack.

```

        ORG          $6000
        MOVE.W       ARG1,-(SP)    ;push ARG onto stack
        MOVE.W       ARG2,-(SP)
        JSR          SUB6          ;call subroutine SUB2
        MOVE.W       (SP)+,RSLT    ;pop answer from stack
END6

ARG1    DC.W         4             ;base
ARG2    DC.W         2             ;exponent
RSLT    DS.W         1             ;result

SUB6    MOVE.W       xx(SP),D1     ;put ARG1 into D1
        MOVE.W       yy(SP),D2     ;put ARG2 into D2
        MOVE.L       #1,D3         ;put starting 1 into D3
LOOP6   SUBQ        #1,D2         ;decrement power
        BMI          EXIT          ;if D2-1<0 then quit SUB2
        MULS        D1,D3         ;multiply out
        BRA         LOOP6         ;and repeat as necessary

EXIT    MOVE.W       D3,zz(SP)
        MOVE.L       (d)          ;move return address to
                                ;correct location for
                                ;return
        ADDQ.L      (e)          ;increment SP to final
                                ;value

        RTS
```

(a) What should be the value of xx to correctly retrieve ARG1 from the stack?

xx=_____

(b) What should be the value of yy to correctly retrieve ARG2 from the stack?

yy=_____

(c) Specify the value of zz to properly put D3 on the stack so that it can be POPed from the stack and put into ARG3 AFTER the subroutine return.

zz=_____

Specify the missing operand fields to make the subroutine work as described.

(d) _____

(e) _____

Name: _____ CWRUnet ID: _____

I have since discovered that some people used a long word argument instead of a word length argument. This makes the stack look like this:
Using this picture the answers are: