1. **(30 points)** Show the state of the stack and registers SP and A6 during the execution of the following program.

```c
void main(void) {
    int a, b, c;
    a = 3;
    b = -4;
    c = iswap(a, &b);
}
```

```c
int iswap(int x, int * y) {
    int t = *y;
    * y = x;
    x = t;
    return (x* *y);
}
```

Code generated by the cross compiler is given below:
void main(void) {

Variable a is at -2(A6)
Variable b is at -4(A6)
Variable c is at -6(A6)

main

ORG $1000
LINK A6,#-6

int a, b, c;

a = 3;
MOVE #3,-2(A6)
b = -4;
MOVE #-4,-4(A6)
c = iswap(a, &b);
PEA.L -4(A6)
MOVE #3,-(A7)
JSR iswap
MOVE D0,-6(A6)

}

UNLK A6
RTS

Function size = 38

int iswap(int x, int * y) {
Parameter x is at 8(A6)
Parameter y is at 10(A6)
Variable t is at -2(A6)
iswap

LINK A6,#-2

int t = *y;
MOVEA.L 10(A6),A4
MOVE (A4),-2(A6)

* y = x;
MOVE 8(A6),(A4)
x = t;
MOVE -2(A6),8(A6)
return (x* *y);
MOVEA.L 10(A6),A4
MOVE (A4),D1
MULS 8(A6),D1
MOVE D1,D0

}

UNLK A6
RTS
END
2. (30 points) The assembler generates the following listing file for a C program.

```assembly
0000003C                     45  *10  int sma(int x[], int n) {
0000003C                     46  * Parameter x is at 8(A6)
0000003C                     47  * Parameter n is at 12(A6)
0000003C                     48  * Variable i is at -2(A6)
0000003C                     49  * Variable sum is at -4(A6)
0000003C                     51  sma
0000003C  4E56 FFFC          52   LINK A6,#-4
00000040  2F07               53   MOVE.L D7,-(A7)
00000042  426E FFFE          56   CLR -2(A6)
00000046  426E FFFC          58   CLR -4(A6)
0000004A  6000 0024          60   BRA L1
0000004E                     61  L2
0000004E  322E FFFC          63   MOVE -4(A6),D1
00000052  302E FFFE          64   MOVE -2(A6),D0
00000056  3400               65   MOVE D0,D2
00000058 E34A                66   LSL #1,D2
0000005A  286E 0008          67   MOVEA.L 8(A6),A4
0000005E  3E34 2000          68   MOVE (A4,D2.W),D7
00000062 CFF4 2000          69   MULS (A4,D2.W),D7
00000066 DE41                70   ADD D1,D7
00000068 3D47 FFFC          71   MOVE D7,-4(A6)
0000006C  526E FFFC          74   ADDQ #1,-2(A6)
00000070 3E2E FFFE          75   MOVE -2(A6),D7
00000074 BE6E 000C          76   CMP 12(A6),D7
00000078 6DD4                77   BLT.S L2
0000007A 302E FFFC          79   MOVE -4(A6),D0
0000007E 2E1F                81   MOVE.L (A7)+,D7
00000080 4E56                82   UNLK A6
00000082 4E75                83   RTS
```

A. (10 points) What does this subroutine do?

B. (10 points) Find the total execution time of the given subroutine on the M68000 as a function of parameter \( n \) and the *clock cycle time*. What is the total execution time when \( n=5 \) and clock frequency is 8 MHz.
C. (5 points) Calculate the average CPI (number of clocks per instructions) for n=100, f=8MHz.

D. (5 points) Calculate the MIPS rate (n = 100, f=8 MHz).
3. (25 points)

A. (5 points) Show assembly code produced by compiling the following C program and discuss it.

```c
void main(void) {
    int x = 4;
    x = (x>0) ? x : -x;
}
```

B. (5 points) Show assembly code produced by compiling the following C program and discuss it.

```c
void main(void) {
    int i;
    int square[20]
    for(i=0; i<20; i++) square[i] = i*i;
}
```
C. (10 points) Show assembly code produced by compiling the following C program and discuss it.

```c
void main(void) {
    int x = 5;
    int *p_x;
    int y1;
    long int y2, y3;
    long int *p_y2;
    int mya[20];
    char *p_mya;
    p_x = &x;
    y1 = 10 + x;
    p_y2 = &y2;
    y3 = 10 + p_y2;
    p_mya = mya;
    mya = mya + 3;
}
```

D. (5 points) Describe the process of passing parameters for the following procedure:

```c
void pqr(int a, int b, int * c)
```

What is done in the main program calling this procedure? How do we access these parameters in the procedure? Explain your answers.
4. (20 points) Write a subroutine using 68K assembly that counts the number of 1’s in an integer n. Note: Start with writing corresponding C code.

```c
int count_bits(int n) /* count the number of 1's in integer n */
{
    int ones_count = 0; /* create and initialize a counter */
    /* your C code goes here */

    return ones_count;
}
```

*1  int count_bits(int n)
* Parameter n is at 8(A6)
* Variable ones_count is at -2(A6)
* Write your assembly code below