1. (10%  
(10 / 5) times as fast as M1. For program 2, M1 is 1.33 (4 / 3) times as fast as M2.

b. The execution rate is (200 x 10^6) / 10 = 20 x 10^6 for M1 and is (160 x 10^6) / 5 = 32 x 10^6 for M2 when running program1.

c. For program 1, CPI of M1 is 200 x 10^6 / 20 x 10^6 = 10; CPI of M2 is 300 x 10^6 / 32 x 10^6 = 9.4.

2. (10%)
Peak performance of M1 is 500 / 1 = 500 MIPS and of M2 is 750 / 2 = 375 MIPS.

3. (10%)
CPI for MFP = 0.1 x 6 + 0.15 x 4 + 0.05 x 20 + 0.7 x 2 = 3.6
MIPS for MFP = 1000 / 3.6 = 278
CPI for MNFP = 2
MIPS for MNFP = 1000 / 2 = 500

4. (10%)
a. (5%) Total execution time of A is 1001 seconds, of B is 110 seconds, and of C is 40 seconds. Computer C is fastest.
b. (5%) C is 25 (1001 / 40) times faster than A and is 2.75 (110 / 40) times faster than B.

5. (20%)
a. (10%) The code determines the most frequent word appearing in the array and returns it in $v1 and its multiplicity in $v0.
b. (10%) The cycles per iteration for “outer” is 1 + 2 + 1 + 1 = 5
The cycles per iteration for “skip” is 1 + 2 + 1 + 1 + 1 + 2 = 8
The total cycles for the outer loop is 5000 x (5+8)

The cycles per iteration for “inner” is 1 + 2 + 2 + 1 = 6
The cycles per iteration for “next” is 1 + 2 = 3
The total cycles for the inner loop is 5000 x 5000 x (6+3)
The overall execution time is \(\frac{5000 \times 13 + 5000 \times 5000 \times 9}{500 \times 10^6} = 0.45\) seconds

6. (10%)
The base address of \(x\) is 0000 0000 0011 1101 0000 1001 0000 0000
\[\text{lui } \$t1, \ 0000 \ 0000 \ 0011 \ 1101 \]
\[\text{ori } \$t1, \ \$t1, \ 0000 \ 1001 \ 0000 \ 0000 \]
\[\text{lw } \$t2, \ 44(\$t1) \]
\[\text{add } \$t2, \ \$t2, \ \$t0 \]
\[\text{sw } \$t2, \ 40(\$t1) \]

7. (20%)
a. (10%)
\[
\begin{align*}
\text{add } \& \ $t0, \ $zero, \ $zero & \quad \# \text{ Temp reg } \$t0 = 0 \\
\text{lw } & \ $t1, \ 0(\$s0) & \quad \# \text{ Temp reg } \$t1 = c \\
\text{lw } \& \ $t2, \ \text{AddressConstant4}(\$zero) & \quad \# \text{ Temp reg } \$t2 = 4 \\
\text{lw } \& \ $t3, \ \text{AddressConstant401}(\$zero) & \quad \# \text{ Temp reg } \$t3 = 401 \\
\text{Loop: add } \& \ $t4, \ \$a1, \ \$t0 & \quad \# \text{ Temp reg } \$t4 = \text{address of } b[i] \\
\text{lw } \& \ $t5, \ 0(\$t4) & \quad \# \text{ Temp reg } \$t5 = b[i] \\
\text{add } \& \ $t6, \ \$t5, \ \$t1 & \quad \# \text{ Temp reg } \$t6 = b[i] + c \\
\text{add } \& \ $t7, \ \$a0, \ \$t0 & \quad \# \text{ Temp reg } \$t7 = \text{address of } a[i] \\
\text{sw } \& \ $t6, \ 0(\$t7) & \quad \# a[i] = b[i] + c \\
\text{add } \& \ $t0, \ \$t0, \ \$t2 & \quad \# i = i + 4 \\
\text{slt } \& \ $t8, \ \$t0, \ \$t3 & \quad \# \$t8 = 1 \text{ if } \$t0 < 401 \ (i \leq 100) \\
\text{bne } \& \ $t8, \ $zero, \ \text{Loop} & \quad \# \text{ go to Loop if } i \leq 100 \\
\end{align*}
\]
b. (5%)
The number of instructions executed is \(4 + 101 \times 8 = 812\).

c. (5%)
The number of data references made is \(3 + 101 \times 2 = 205\).