1. Use virtual register notation for each of the following.

Change the following into four instructions which are likely a time improvement over the current instructions when it comes to actual code generation. Assume register \( r_7 \) is free and registers \( r_2, r_3, \) and \( r_4 \) are used later. [6 pts]

\[
\begin{align*}
& r_2 = r_1 \times 8 \\
& r_2 = r_2 + r_3 \\
& r_3 = r_1 \times 8 \\
& r_4 = r_4 + r_2
\end{align*}
\]

What two terms describe these particular peephole optimizations?
1) ____________________________
2) ____________________________

Change the following into another pair of instructions which are likely a time improvement over the current instructions when it comes to actual code generation. Assume registers \( r_5 \) and \( r_6 \) are used later. [5 pts]

\[
\begin{align*}
& r_5 = 3 + 29 \\
& r_6 = r_1 \times r_5
\end{align*}
\]

What three terms describe these particular peephole optimizations?
1) ____________________________
2) ____________________________
3) ____________________________

2. Given the following variable definitions

```c
static int x = 858;    // initialized static variable
float y = 858;         // initialized global variable
```

write the corresponding SPARC assembly code that should be generated to properly allocate space for each variable, along with their initial values and alignment, and to ensure proper access/visibility to these variables if another file is linked to this code’s object file. [7 pts]

```assembly
.section __________
.align 4

x: __________ __________
   __________ __________

y: __________ __________
```

3. Variables declared with the keyword _______________________ won’t be optimized by the compiler. [2 pts]
Given the following assembly code, draw a box around each basic block and label each box with a consecutive basic block number (e.g. #1, #2, #3, etc.):

```
.section    "text"
.align      4
.global     fib

fib:
set    -(92 + 32) & -8, %g1
save   %sp, %g1, %sp
st     %i0, [%fp+68]
set    68, %l7
add    %fp, %l7, %l7
ld     [%l7], %o0
set    2, %o1
cmp    %o0, %g0
bge    .$$..cmp.1
mov    %g0, %o0
inc    %o0

.$$..cmp.1:
set    -4, %o1
add    %fp, %o1, %o1
st     %o0, [%o1]
set    -4, %l7
add    %fp, %l7, %l7
ld     [%l7], %o0
cmp    %o0, %g0
be     .$$..endif.1
nop    set    68, %l7
add    %fp, %l7, %l7
ld     [%l7], %i0
ret    restore

.$$..endif.1:
set    68, %l7
add    %fp, %l7, %l7
ld     [%l7], %o0
sub    %o0, %o1, %o0
set    -16, %o1
add    %fp, %o1, %o1
st     %o0, [%o1]
set    -16, %l7
add    %fp, %l7, %l7
ld     [%l7], %o0
cmp    %o0, %g0
bge    .$$..cmp.2
nop    inc    %o0

.$$..cmp.2:
set    -28, %o1
add    %fp, %o1, %o1
st     %o0, [%o1]
set    -20, %l7
add    %fp, %l7, %l7
ld     [%l7], %o0
set    -28, %l7
add    %fp, %l7, %l7
ld     [%l7], %o1
add    %o0, %o1, %o0
set    -32, %o1
add    %fp, %o1, %o1
st     %o0, [%o1]
set    -32, %l7
add    %fp, %l7, %l7
ld     [%l7], %i0
ret    restore
```

5. What are the values of \(a\), \(b\), and \(c\) after the following Reduced-C statements? [3 pts]

```
int a = 20;
int b = ++a;
int c = b++ + ++a;
a = b++;
```

Value of \(a\) is __________

Value of \(b\) is __________

Value of \(c\) is __________