1. Consider the following Prolog program.

\texttt{takes(jane\_doo, his201).}
\texttt{takes(jane\_doo, cs254).}
\texttt{takes(ajit\_chandra, art302).}
\texttt{takes(ajit\_chandra, cs254).}
\texttt{classmates(X,Y) :- takes(X,Z), takes(Y,Z).}

What does the query \texttt{classmates(jane\_doo,X)} return? Give details of how the search procedure produces this result.

2. Consider the following C-like program.

\texttt{int w = 3;}
\texttt{int x = 10;}
\texttt{int incw() \{ return ++w; \}}
\texttt{int incx() \{ return ++x; \}}
\texttt{void foo(y, z){
  printf("%d\n", y + y);
  x = 1;
  printf("%d\n", z);
}}
\texttt{int main() {
  foo(incw(), incx());
  return 0;
}}

What does it print if the language uses
(a) Applicative-order evaluation?
(b) Normal-order evaluation?

3. In an assembly-language-like notation (e.g., use MIPS or a pseudocode of your own choosing), write what a good optimizing compiler would produce for the following two switch statements:

\texttt{switch (a) {
  case 1: x = 3; break;
  case 2: x = 5; break;
  case 3: x = 15; break;
  case 4: x = 20; break;
  case 5: x = 23; break;
  default: x = 28; break;
}}
\texttt{switch (b) {
  case 1: x = 3; break;
  case 10: x = 5; break;
  case 100: x = 15; break;
  case 1000: x = 20; break;
  default: x = 25; break;
}}

4. For a 32-bit little-endian processor with the usual alignment rules, show the memory layout and size in bytes of the following C types.

\texttt{union {
  struct {
    int a; /* 32-bit */
    char b; /* 8-bit */
  } s;
  int c;
} ul1;}
\texttt{struct {
  char a;
  short b;
  int c;
  char d;
} s1;}
\texttt{struct {
  char a;
  char d;
  short b;
  int c;
} s2;}

Submit solution on paper (no email). Please write your name clearly on the paper.

Do this assignment alone. You may consult the instructor and the TAs, but not other students.