1. (20%) Consider the following code segment when discussing short-circuit evaluation.
   ```c
   index = 0;
   while (index < listlen) && (LIST[index] != key)
       index = index + 1;
   ```
   Write a short program that includes this code segment and test whether your programming system allows or disallows short-circuit evaluation. Test especially when `LIST[]` has `listlen-1` elements and `index==listlen`. Print out the assembly code of your program and show where the code supports your conclusion. (Bonus 10%: Modify the assembly code so that if it does not allow short-circuit evaluation, it now does, and vice versa.)

2. (20%) Consider the following code segment when discussing multiple-way selection.
   ```c
   switch (x)
     default:
       if (prime(x))
         case 2: case 3: case 5: case 7:
             process_prime(x);
       else
         case 4: case 6: case 8: case 9: case 10:
             process_composite(x);
   ```
   List three advantages that this code may have, to compare with other “normal” code.

3. (30%) Write a short program to include the code segment in Problem 2. You may like to use a dumb loop to replace each of the two functions: `process_prime(x)` and `process_composite(x)`. Now write another program that uses a normal, naïve code to implement the above operations. Compare the two programs in terms of execution time and code size. (Hint: You may need to adjust the execution time of your dumb loops and your inputs to see the differences.)

4. (30%) Consider the following C program:
   ```c
   void swap_met(int i, int j) {
       int temp;
       temp = i;  i = j;  j = temp;
   }
   void main() {
       int val = 3, arrlist[5] = {4, 6, 3, 1, 2};
       swap_met(val, arrlist[0]);
       swap_met(arrlist[0], arrlist[1]);
       swap_met(val, arrlist[val]);
   }
   ```
For each of the following parameter-passing methods, what are all of the values of the variables \texttt{value} and \texttt{list} after each of the three calls to \texttt{swap}? Explain your answers.

(a) pass by value
(b) pass by reference
(c) pass by value-result