Do not write on this test. Record all answers on the bubble sheet. **Closed book. No notes.** Work strictly from memory. **No calculators. Scratch paper okay.**

1/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume atoi, simpleStatement, and simpleCompare each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   d = 1; do {
      e = n; while ( e > 1 ) {
         h = n; do {
            if ( simpleCompare ) {
               for ( g = 1 ; g < n ; g++ ) {
                  for ( i = n ; i > 1 ; i-- ) {
                     simpleStatement;
                  }
               }
            } else {
               simpleStatement;
            }
         h--; } while ( h > 1 );
      e--; }
   d += 3; } while ( d < n );
   return 0; }
```

2/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume atoi, simpleStatement, and simpleCompare each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   }
   return 0; }
```

3/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         simpleStatement;
      } else {
         simpleStatement;
      }
   } else {
      simpleStatement;
   }
   return 0; }
```

4/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   e = 1; while ( e < n ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               f = 1; do {
                  if ( simpleCompare ) {
                     if ( simpleCompare ) {
                        for ( b = 1 ; b < n ; b++ ) {
                           simpleStatement;
                        }
                     } else {
                        simpleStatement;
                     }
                  } else {
                     for ( d = 1 ; d < n ; d++ ) {
                        simpleStatement;
                     }
                  }
               f++; } while ( f < n );
            }
         }
      }
   e += 3; }
   return 0; }
```

5/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         simpleStatement;
      } else {
         simpleStatement;
      }
   } else {
      simpleStatement;
   }
   return 0; }
```

6/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         c = n; while ( c > 1 ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         c -= 5; }
      } else {
         for ( b = 1 ; b < n ; b += 1 ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   }
   return 0; }
```

7/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( d = n ; d > 1 ; d-- ) {
         e = n; while ( e > 1 ) {
            simpleStatement;
         e -= 2; }
      }
   } else {
      simpleStatement;
   }
   return 0; }
```

8/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( k = n ; k > 1 ; k-- ) {
      if ( simpleCompare ) {
         for ( b = n ; b > 1 ; b-- ) {
            i = 1; do {
               h = n; do {
                  for ( a = 1 ; a < n ; a++ ) {
                     e = n; do {
                        if ( simpleCompare ) {
                           if ( simpleCompare ) {
                              simpleStatement;
                           } else {
                              simpleStatement;
                           }
                        } else {
                           simpleStatement;
                        }
                     e -= 10; } while ( e > 1 );
                  }
               h -= 2; } while ( h > 1 );
            i++; } while ( i < n );
         }
      }
   }
   return 0; }
```

9/2p. Give a tight big-oh $\Theta()$ bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      f = 1; while ( f < n ) {
         b = n; do {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               }
            } else {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            }
         b--; } while ( b > 1 );
      f += 10; }
   } else {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               i = 1; while ( i < n ) {
                  simpleStatement;
               i++; }
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      } else {
         g = 1; while ( g < n ) {
            simpleStatement;
         g++; }
      }
   }
   return 0; }
```

10/2p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( c = 1 ; c < n ; c++ ) {
      g = 1; while ( g < n ) {
         if ( simpleCompare ) {
            for ( d = n ; d > 1 ; d -= 3 ) {
               for ( b = n ; b > 1 ; b -= 5 ) {
                  e = 1; while ( e < n ) {
                     if ( simpleCompare ) {
                        f = n; while ( f > 1 ) {
                           if ( simpleCompare ) {
                              h = n; while ( h > 1 ) {
                                 for ( i = 1 ; i < n ; i += 10 ) {
                                    if ( simpleCompare ) {
                                       if ( simpleCompare ) {
                                          simpleStatement;
                                       }
                                    }
                                 }
                              h -= 3; }
                           }
                        f--; }
                     } else {
                        if ( simpleCompare ) {
                           if ( simpleCompare ) {
                              if ( simpleCompare ) {
                                 simpleStatement;
                              }
                           } else {
                              simpleStatement;
                           }
                        } else {
                           k = 1; while ( k < n ) {
                              simpleStatement;
                           k += 10; }
                        }
                     }
                  e += 3; }
               }
            }
         }
      g += 3; }
   }
   return 0; }
```

Total points 20.

Answer Key (points per line)

1 (2).	E (n^5)
2 (2).	J (1)
3 (2).	J (1)
4 (2).	G (n^3)
5 (2).	J (1)
6 (2).	I (n)
7 (2).	H (n^2)
8 (2).	D (n^6)
9 (2).	H (n^2)
10 (2).	B (n^8)

Total points 20.