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) $n^2\sqrt{n}$ (C) n^2 (E) $n\sqrt{n}$ (G) $\sqrt{n}\lg n$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}\lg n$ (F) n (H) $\sqrt{n}$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   g = 1; while ( g < n ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   g += 1; }
   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^3 (C) n^2 (E) $n\sqrt{n}$ (G) $n\lg n$ (I) $\sqrt{n}$
(B) $n^2 \lg n$ (D) $n\sqrt{n}\lg n$ (F) $n\lg^2 n$ (H) $\sqrt{n}\lg n$ (J) $\lg^3 n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      k = n; do {
         for ( a = 1 ; a < n ; a *= 5 ) {
            e = 1; do {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            e *= 2; } while ( e < n );
         }
      k--; } while ( k > 1 );
   } else {
      h = 1; while ( h * h < n ) {
         if ( simpleCompare ) {
            simpleStatement;
         }
      h += 1; }
   }
   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) $n^3\sqrt{n}$ (C) $n^2\sqrt{n}$ (E) $n^2 \lg n$ (G) n (I) $\lg n$
(B) $n^3 \lg n$ (D) $n^2 \lg^2 n$ (F) n^2 (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( g = 1 ; g < n ; g *= 2 ) {
      if ( simpleCompare ) {
         for ( i = 1 ; i < n ; i++ ) {
            for ( j = n ; j > 1 ; j /= 3 ) {
               d = n; while ( d > 1 ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               d -= 2; }
            }
         }
      }
   }
   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) $n^2\sqrt{n}$ (C) n^2 (E) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg^2 n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         e = 1; while ( e * e < n ) {
            g = 1; while ( g < n ) {
               simpleStatement;
            g += 2; }
         e++; }
      } else {
         simpleStatement;
      }
   } else {
      if ( simpleCompare ) {
         simpleStatement;
      } else {
         simpleStatement;
      }
   }
   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) $n^2\sqrt{n}$ (C) $n\sqrt{n}\lg n$ (E) $n\lg n$ (G) $\sqrt{n}\lg n$ (I) $\lg n$
(B) n^2 (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               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) $n^2\lg n$ (C) $n\sqrt{n}\lg n$ (E) n (G) $\sqrt{n}$ (I) $\lg n$
(B) n^2 (D) $n\sqrt{n}$ (F) $\sqrt{n}\lg n$ (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   k = n; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  f = 1; do {
                     simpleStatement;
                  f++; } while ( f * f < n );
               }
            } else {
               simpleStatement;
            }
         } else {
            if ( simpleCompare ) {
               a = 1; do {
                  simpleStatement;
               a++; } while ( a * a < n );
            } else {
               simpleStatement;
            }
         }
      }
   k /= 2; } while ( k > 1 );
   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) $n^2\sqrt{n}$ (C) $n\sqrt{n}$ (E) n (G) $\sqrt{n}$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n \lg n$ (F) $\sqrt{n} \lg n$ (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   g = 1; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   g++; } while ( g * g < n );
   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) $n^2\sqrt{n}$ (C) $n\sqrt{n} \lg n$ (E) n (G) $\sqrt{n}$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n \lg n$ (F) $\sqrt{n} \lg n$ (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         for ( k = 1 ; k * k < n ; k++ ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      } else {
         simpleStatement;
      }
   } else {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   }
   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) $n^2 \lg n$ (C) $n\sqrt{n}\lg n$ (E) n (G) $\sqrt{n}$ (I) $\lg n$
(B) n^2 (D) $n\sqrt{n}$ (F) $\sqrt{n}\lg n$ (H) $\lg^2 n$ (J) 1

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

10/2p. 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) $n^2\sqrt{n}$ (C) $n\sqrt{n}\lg n$ (E) $n\lg n$ (G) $\sqrt{n}\lg n$ (I) $\lg^2 n$
(B) n^2 (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   g = 1; while ( g < n ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      } else {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      }
   g *= 2; }
   return 0; }
```

11/2p. 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) $n^2\sqrt{n}$ (C) n^2 (E) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n\sqrt{n} \lg n$ (F) n (H) $\sqrt{n}$ (J) 1

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

12/2p. 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) $n^2\sqrt{n}$ (C) $n\sqrt{n} \lg n$ (E) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}$ (F) n (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( d = n ; d > 1 ; d-- ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               a = 1; while ( a < n ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               a *= 3; }
            } else {
               simpleStatement;
            }
         } else {
            h = 1; while ( h * h < n ) {
               simpleStatement;
            h += 10; }
         }
      }
   }
   return 0; }
```

13/2p. 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^3 (C) $n^2 \lg n$ (E) $n \lg^2 n$ (G) $\sqrt{n} \lg^3 n$ (I) $\lg^4 n$
(B) $n^2\sqrt{n}$ (D) n^2 (F) $n \lg n$ (H) $\sqrt{n} \lg n$ (J) $\lg^2 n$

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

14/2p. 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) $n^7 \lg n$ (C) $n^4\sqrt{n}\lg^2 n$ (E) $n^2\sqrt{n}$ (G) n^2 (I) $n\lg^6 n$
(B) $n^6\sqrt{n}\lg n$ (D) $n^3\sqrt{n}\lg n$ (F) $n^2\lg n$ (H) $n\sqrt{n}\lg^6 n$ (J) $\sqrt{n}\lg^7 n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( f = n ; f > 1 ; f /= 3 ) {
      c = 1; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  b = 1; while ( b < n ) {
                     e = 1; while ( e < n ) {
                        h = 1; do {
                           i = n; do {
                              a = 1; while ( a < n ) {
                                 d = 1; while ( d * d < n ) {
                                    simpleStatement;
                                 d += 5; }
                              a++; }
                           i -= 2; } while ( i > 1 );
                        h++; } while ( h * h < n );
                     e *= 2; }
                  b++; }
               }
            } else {
               g = n; while ( g > 1 ) {
                  j = 1; do {
                     if ( simpleCompare ) {
                        k = n; do {
                           if ( simpleCompare ) {
                              simpleStatement;
                           } else {
                              simpleStatement;
                           }
                        k -= 3; } while ( k > 1 );
                     } else {
                        simpleStatement;
                     }
                  j *= 5; } while ( j < n );
               g--; }
            }
         }
      c++; } while ( c * c < n );
   }
   return 0; }
```

15/2p. 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) $n^5 \lg n$ (C) $n^3 \lg^3 n$ (E) n^2 (G) $n\sqrt{n} \lg^3 n$ (I) $n \lg n$
(B) $n^3\sqrt{n} \lg^2 n$ (D) $n^2\sqrt{n} \lg^3 n$ (F) $n\sqrt{n} \lg^4 n$ (H) $n\sqrt{n} \lg n$ (J) n

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         g = n; do {
            b = 1; while ( b * b < n ) {
               c = n; do {
                  if ( simpleCompare ) {
                     for ( e = n ; e > 1 ; e /= 2 ) {
                        d = 1; do {
                           k = n; do {
                              if ( simpleCompare ) {
                                 simpleStatement;
                              } else {
                                 simpleStatement;
                              }
                           k /= 3; } while ( k > 1 );
                        d *= 2; } while ( d < n );
                     }
                  }
               c /= 5; } while ( c > 1 );
            b += 2; }
         g--; } while ( g > 1 );
      } else {
         if ( simpleCompare ) {
            a = n; do {
               for ( h = 1 ; h < n ; h++ ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               }
            a /= 2; } while ( a > 1 );
         } else {
            f = 1; do {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            f++; } while ( f * f < n );
         }
      }
   }
   return 0; }
```

Total points 30.

Answer Key (points per line)

1 (2).	`F`	(n)
2 (2).	`F`	$(n \lg^2 n)$
3 (2).	`D`	$(n^2 \lg^2 n)$
4 (2).	`D`	$(n\sqrt{n})$
5 (2).	`J`	(1)
6 (2).	`F`	$(\sqrt{n} \lg n)$
7 (2).	`G`	$(\sqrt{n})$
8 (2).	`G`	$(\sqrt{n})$
9 (2).	`G`	$(\sqrt{n})$
10 (2).	`J`	$(\lg n)$
11 (2).	`F`	(n)
12 (2).	`D`	$(n\sqrt{n})$
13 (2).	`G`	$(\sqrt{n} \lg^3 n)$
14 (2).	`C`	$(n^4\sqrt{n} \lg^2 n)$
15 (2).	`F`	$(n\sqrt{n} \lg^4 n)$

Total points 30.