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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   d = n; while ( d > 1 ) {
      e = 1; do {
         for ( c = 1 ; c * c < n ; c++ ) {
            h = 1; while ( h < n ) {
               simpleStatement;
            h *= 2; }
         }
      e += 3; } while ( e * e < n );
   d -= 2; }
   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^2 (C) $n\sqrt{n}$ (E) n (G) $\sqrt{n}$ (I) $\lg n$
(B) $n\sqrt{n} \lg n$ (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 ) {
         b = 1; do {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     if ( simpleCompare ) {
                        simpleStatement;
                     } else {
                        simpleStatement;
                     }
                  }
               }
            } else {
               for ( h = 1 ; h < n ; h += 10 ) {
                  simpleStatement;
               }
            }
         b *= 5; } while ( b < n );
      }
   }
   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) $n^2\sqrt{n}$ (C) n^2 (E) $n\sqrt{n}$ (G) n (I) $\lg^2 n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}\lg n$ (F) $n \lg n$ (H) $\sqrt{n}\lg n$ (J) 1

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( i = 1 ; i * i < n ; i += 3 ) {
      a = 1; do {
         e = 1; while ( e < n ) {
            if ( simpleCompare ) {
               f = n; do {
                  simpleStatement;
               f -= 10; } while ( f > 1 );
            }
         e *= 2; }
      a += 10; } while ( a < n );
   }
   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^3 (C) n^2 (E) $n\sqrt{n}$ (G) $\sqrt{n} \lg^2 n$ (I) $\lg^2 n$
(B) $n^2 \lg n$ (D) $n\sqrt{n} \lg n$ (F) $n \lg n$ (H) $\sqrt{n}$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( f = 1 ; f * f < n ; f += 1 ) {
      for ( i = 1 ; i * i < n ; i += 2 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  for ( h = 1 ; h < n ; h *= 3 ) {
                     simpleStatement;
                  }
               }
            } else {
               simpleStatement;
            }
         } else {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  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) $n^2 \lg n$ (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) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( b = 1 ; b < n ; b++ ) {
      if ( simpleCompare ) {
         i = 1; do {
            simpleStatement;
         i *= 3; } while ( i < n );
      } 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) $n^2 \lg n$ (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) 1

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

9/2p. 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) $n^2\sqrt{n}$	(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]);
   e = 1; do {
      if ( simpleCompare ) {
         a = n; while ( a > 1 ) {
            simpleStatement;
         a -= 5; }
      } else {
         simpleStatement;
      }
   e *= 5; } while ( e < n );
   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) $n^2\sqrt{n}$	(C) $n\sqrt{n}$	(E) n	(G) $\sqrt{n}$	(I) $\lg n$
(B) n^2	(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 ) {
         if ( simpleCompare ) {
            for ( f = 1 ; f * f < n ; f += 10 ) {
               for ( e = n ; e > 1 ; e /= 2 ) {
                  simpleStatement;
               }
            }
         } else {
            simpleStatement;
         }
      } else {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      }
   }
   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) $n\sqrt{n}$ (G) $n\lg n$ (I) $\sqrt{n}\lg n$
(B) $n^2\lg n$ (D) $n\sqrt{n}\lg n$ (F) $n\lg^2 n$ (H) $\sqrt{n}\lg^2 n$ (J) $\lg^3 n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( b = n ; b > 1 ; b-- ) {
      for ( e = 1 ; e < n ; e *= 2 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               k = 1; do {
                  simpleStatement;
               k *= 2; } while ( k < n );
            } 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) $\sqrt{n}$ (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 ) {
      c = n; while ( c > 1 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      c--; }
   } else {
      h = 1; do {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      h *= 2; } while ( h < n );
   }
   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^4 (C) $n^3 \lg n$ (E) n^2 (G) $n \lg^2 n$ (I) $\sqrt{n} \lg n$
(B) $n^3 \sqrt{n}$ (D) $n^2 \sqrt{n} \lg n$ (F) $n \sqrt{n}$ (H) $n \lg n$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   k = n; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            c = 1; do {
               if ( simpleCompare ) {
                  for ( h = n ; h > 1 ; h-- ) {
                     i = 1; do {
                        simpleStatement;
                     i += 10; } while ( i * i < n );
                  }
               } else {
                  simpleStatement;
               }
            c *= 5; } while ( c < n );
         } else {
            for ( g = n ; g > 1 ; g /= 5 ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            }
         }
      } else {
         e = 1; do {
            if ( simpleCompare ) {
               b = 1; while ( b * b < n ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               b += 3; }
            } else {
               a = 1; do {
                  simpleStatement;
               a++; } while ( a * a < n );
            }
         e += 10; } while ( e * e < n );
      }
   k -= 10; } while ( k > 1 );
   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) $n^3 \lg n$ (C) $n^2\sqrt{n}$ (E) n^2 (G) $n\sqrt{n}\lg n$ (I) $\lg^2 n$
(B) n^3 (D) $n^2 \lg n$ (F) $n\sqrt{n}\lg^2 n$ (H) $n \lg^2 n$ (J) 1

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

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

Total points 30.

Answer Key (points per line)

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

Total points 30.