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) $n^2 \lg n$ (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) $\lg^2 n$ (J) 1

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

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

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

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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( e = 1 ; e < n ; e *= 3 ) {
         g = 1; while ( g * g < n ) {
            for ( i = n ; i > 1 ; i -= 2 ) {
               simpleStatement;
            }
         g += 5; }
      }
   } else {
      for ( d = 1 ; d * d < n ; d += 2 ) {
         simpleStatement;
      }
   }
   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^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) $\lg n$

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

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

14/2p. 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) $n^5 \lg n$ (C) $n^4 \lg^2 n$ (E) $n^3\sqrt{n}\lg^3 n$ (G) $n^2 \lg^4 n$ (I) n
(B) $n^4\sqrt{n}$ (D) $n^4 \lg n$ (F) $n^2\sqrt{n}\lg^3 n$ (H) $n\sqrt{n}\lg^5 n$ (J) $\sqrt{n}$

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

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

Total points 30.

Answer Key (points per line)

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

Total points 30.