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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   b = n; while ( b > 1 ) {
      for ( k = 1 ; k < n ; k += 10 ) {
         d = 1; do {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         d += 5; } while ( d * d < n );
      }
   b--; }
   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) $n\lg n$ (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 ) {
      k = 1; while ( k < n ) {
         if ( simpleCompare ) {
            j = n; do {
               simpleStatement;
            j /= 2; } while ( j > 1 );
         }
      k += 3; }
   } else {
      b = 1; do {
         simpleStatement;
      b += 10; } while ( b * b < n );
   }
   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^5 (C) $n^3\sqrt{n}\lg n$ (E) n^3 (G) $n^2\lg^3 n$ (I) $\lg^5 n$
(B) $n^4\lg n$ (D) $n^3\lg n$ (F) $n^2\sqrt{n}\lg^2 n$ (H) $n\lg^4 n$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         for ( f = 1 ; f < n ; f += 1 ) {
            for ( k = n ; k > 1 ; k-- ) {
               if ( simpleCompare ) {
                  b = 1; do {
                     h = 1; do {
                        if ( simpleCompare ) {
                           a = 1; while ( a < n ) {
                              simpleStatement;
                           a++; }
                        }
                     h *= 5; } while ( h < n );
                  b += 10; } while ( b < 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) $n^2\sqrt{n}$ (C) $n\sqrt{n}\lg n$ (E) $n\lg n$ (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]);
   for ( f = 1 ; f < n ; f++ ) {
      if ( simpleCompare ) {
         h = n; while ( h > 1 ) {
            for ( b = 1 ; b * b < n ; b++ ) {
               simpleStatement;
            }
         h /= 5; }
      }
   }
   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) $n\lg n$ (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 ) {
      for ( j = 1 ; j < n ; j *= 2 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         }
      }
   } else {
      if ( simpleCompare ) {
         e = n; do {
            simpleStatement;
         e--; } while ( e > 1 );
      } 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) $n^2\sqrt{n}$ (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 ) {
      for ( d = 1 ; d * d < n ; d += 2 ) {
         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 (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 ) {
         d = n; do {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         d /= 5; } while ( d > 1 );
      } else {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      }
   } else {
      j = n; do {
         if ( simpleCompare ) {
            simpleStatement;
         }
      j -= 2; } while ( j > 1 );
   }
   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) $n \lg n$ (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 ) {
      if ( simpleCompare ) {
         a = n; while ( a > 1 ) {
            i = n; while ( i > 1 ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            i /= 3; }
         a--; }
      }
   }
   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) $n^2 \lg n$ (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 ) {
      for ( a = n ; a > 1 ; a /= 2 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   } else {
      if ( simpleCompare ) {
         e = 1; do {
            simpleStatement;
         e *= 2; } while ( e < n );
      } else {
         simpleStatement;
      }
   }
   return 0; }
```

11/2p. 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) $n^2 \lg n$ (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) $\lg n$

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

12/2p. 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^2 (E) $n\sqrt{n}$ (G) $\sqrt{n}$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}\lg n$ (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 ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      } else {
         c = 1; while ( c * c < n ) {
            simpleStatement;
         c++; }
      }
   } else {
      if ( simpleCompare ) {
         for ( b = 1 ; b * b < n ; b++ ) {
            for ( f = 1 ; f < n ; f++ ) {
               simpleStatement;
            }
         }
      }
   }
   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) $n\sqrt{n}\lg^2 n$ (G) $n \lg n$ (I) $\lg n$
(B) $n^2\sqrt{n}$ (D) n^2 (F) $n \lg^3 n$ (H) $\lg^2 n$ (J) 1

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      f = n; while ( f > 1 ) {
         for ( j = n ; j > 1 ; j /= 5 ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     if ( simpleCompare ) {
                        for ( i = 1 ; i < n ; i *= 3 ) {
                           simpleStatement;
                        }
                     }
                  } else {
                     if ( simpleCompare ) {
                        simpleStatement;
                     }
                  }
               }
            }
         }
      f -= 3; }
   } else {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            }
         }
      } else {
         k = n; while ( k > 1 ) {
            if ( simpleCompare ) {
               simpleStatement;
            }
         k /= 5; }
      }
   }
   return 0; }
```

15/2p. 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^3 (C) $n^2 \lg n$ (E) $n \lg^2 n$ (G) $\sqrt{n} \lg^2 n$ (I) $\lg^2 n$
(B) $n^2\sqrt{n}$ (D) $n\sqrt{n} \lg n$ (F) $n \lg n$ (H) $\lg^3 n$ (J) 1

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

Total points 30.

Answer Key (points per line)

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

Total points 30.