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) 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]);
   if ( simpleCompare ) {
      h = 1; do {
         if ( simpleCompare ) {
            simpleStatement;
         }
      h++; } while ( h < n );
   } 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) $n^2\sqrt{n}\lg n$ (C) $n^2 \lg^2 n$ (E) n^2 (G) $n\sqrt{n}$ (I) $\sqrt{n}\lg n$
(B) $n^2\sqrt{n}$ (D) $n^2 \lg n$ (F) $n\sqrt{n}\lg^2 n$ (H) $n \lg n$ (J) $\lg n$

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( i = n ; i > 1 ; i-- ) {
      f = n; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               d = n; while ( d > 1 ) {
                  simpleStatement;
               d--; }
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      f -= 3; } while ( f > 1 );
   }
   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) $n^2 \lg n$ (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) $\sqrt{n}$ (J) 1

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      a = n; while ( a > 1 ) {
         if ( simpleCompare ) {
            b = 1; while ( b < n ) {
               e = 1; do {
                  simpleStatement;
               e += 5; } while ( e * e < n );
            b += 10; }
         } else {
            simpleStatement;
         }
      a /= 3; }
   } else {
      h = n; do {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      h /= 2; } while ( h > 1 );
   }
   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) $n^2\sqrt{n}$ (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) $n\lg n$ (H) $\lg^2 n$ (J) 1

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

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) n (I) $\lg^2 n$
(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 ) {
         if ( simpleCompare ) {
            for ( f = 1 ; f < n ; f += 3 ) {
               if ( simpleCompare ) {
                  for ( b = 1 ; b < n ; b++ ) {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            }
         }
      } else {
         if ( simpleCompare ) {
            i = 1; while ( i * i < n ) {
               simpleStatement;
            i++; }
         } else {
            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) $n^3\sqrt{n}$ (C) $n^2\sqrt{n}$ (E) n^2 (G) $\lg^4 n$ (I) $\lg n$
(B) $n^3\lg n$ (D) $n^2\lg n$ (F) $\sqrt{n}\lg n$ (H) $\lg^2 n$ (J) 1

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

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

Total points 30.

Answer Key (points per line)

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

Total points 30.