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

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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   k = 1; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            e = 1; do {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            e += 5; } while ( e < n );
         } else {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   k++; } while ( k < 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) $n^2\sqrt{n}$ (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 ) {
      j = 1; do {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      j += 2; } while ( j * j < 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) $n^4\sqrt{n}\lg^2 n$ (C) $n^3\lg^3 n$ (E) $n\sqrt{n}\lg^5 n$ (G) $n\lg^5 n$ (I) $\lg^7 n$
(B) $n^3\sqrt{n}\lg n$ (D) $n^2\sqrt{n}\lg^4 n$ (F) $n\sqrt{n}\lg^4 n$ (H) $\sqrt{n}$ (J) 1

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

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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( g = 1 ; g * g < n ; g += 10 ) {
         d = 1; while ( d * d < n ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  i = n; do {
                     c = 1; do {
                        simpleStatement;
                     c += 1; } while ( c < n );
                  i--; } while ( i > 1 );
               }
            } else {
               for ( h = n ; h > 1 ; h /= 2 ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               }
            }
         d += 3; }
      }
   } else {
      if ( simpleCompare ) {
         f = n; while ( f > 1 ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  for ( a = n ; a > 1 ; a /= 5 ) {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            }
         f -= 1; }
      } else {
         k = 1; while ( k < n ) {
            if ( simpleCompare ) {
               simpleStatement;
            }
         k += 10; }
      }
   }
   return 0; }
```

Total points 30.

Answer Key (points per line)

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

Total points 30.