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 (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 ) {
         for ( e = 1 ; e * e < n ; e += 10 ) {
            c = n; while ( c > 1 ) {
               simpleStatement;
            c--; }
         }
      } else {
         simpleStatement;
      }
   } 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^3 (C) n^2 (E) $n\sqrt{n}$ (G) $\sqrt{n}$ (I) $\lg n$
(B) $n^2\sqrt{n}$ (D) $n\sqrt{n}\lg n$ (F) $\sqrt{n}\lg^2 n$ (H) $\lg^3 n$ (J) 1

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

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

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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( c = 1 ; c * c < n ; c++ ) {
      a = 1; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               k = 1; while ( k < n ) {
                  d = n; do {
                     if ( simpleCompare ) {
                        if ( simpleCompare ) {
                           if ( simpleCompare ) {
                              simpleStatement;
                           } else {
                              simpleStatement;
                           }
                        }
                     } else {
                        simpleStatement;
                     }
                  d--; } while ( d > 1 );
               k++; }
            }
         } else {
            for ( f = 1 ; f < n ; f++ ) {
               for ( e = n ; e > 1 ; e /= 2 ) {
                  if ( simpleCompare ) {
                     for ( j = 1 ; j * j < n ; j += 3 ) {
                        if ( simpleCompare ) {
                           simpleStatement;
                        }
                     }
                  } else {
                     simpleStatement;
                  }
               }
            }
         }
      a += 2; } while ( a < n );
   }
   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^6 (C) n^5 (E) $n^2 \lg^5 n$ (G) $n \lg^6 n$ (I) $\lg^2 n$
(B) $n^5 \lg n$ (D) $n^2 \sqrt{n} \lg^2 n$ (F) $n \sqrt{n} \lg^5 n$ (H) $\lg^7 n$ (J) 1

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

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

Total points 30.

Answer Key (points per line)

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

Total points 30.