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^4 (C) $n^3 \lg n$ (E) $n^2 \lg n$ (G) $n \lg^2 n$ (I) $\sqrt{n} \lg^3 n$
(B) $n^3\sqrt{n}$ (D) $n^2\sqrt{n}$ (F) $n\sqrt{n} \lg^2 n$ (H) $n \lg n$ (J) 1

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( i = 1 ; i < n ; i += 5 ) {
      b = 1; while ( b * b < n ) {
         for ( h = 1 ; h * h < n ; h++ ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         }
      b++; }
   }
   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) $\lg^2 n$
(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 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   } else {
      for ( j = 1 ; j < n ; j += 5 ) {
         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) $\lg^2 n$ (J) 1

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( b = n ; b > 1 ; b /= 3 ) {
      for ( f = 1 ; f * f < n ; f++ ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               e = 1; while ( e < n ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               e++; }
            } else {
               k = 1; while ( k < n ) {
                  simpleStatement;
               k++; }
            }
         }
      }
   }
   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) $n\sqrt{n}$ (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) $\lg n$

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

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

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

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

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

Total points 30.

Answer Key (points per line)

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

Total points 30.