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) $n \lg n$ (G) $\sqrt{n}$ (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 ) {
         simpleStatement;
      } else {
         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^3 (C) n^2 (E) $n\sqrt{n}$ (G) 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) $\lg n$

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

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

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

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

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

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            for ( f = 1 ; f * f < n ; f += 2 ) {
               g = 1; do {
                  if ( simpleCompare ) {
                     e = 1; do {
                        if ( simpleCompare ) {
                           simpleStatement;
                        } else {
                           simpleStatement;
                        }
                     e++; } while ( e * e < n );
                  } else {
                     simpleStatement;
                  }
               g *= 3; } while ( g < n );
            }
         }
      } else {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         } else {
            d = 1; do {
               simpleStatement;
            d++; } while ( d * d < n );
         }
      }
   }
   return 0; }
```

Total points 30.

Answer Key (points per line)

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

Total points 30.