Do NOT write on this test. Record all answers on the bubble sheet. **Closed book. No notes.** Work strictly from memory. **No calculators. No time limit. Scratch paper okay.**

Precedence: What is the value of each expression? Mark (I) for error, (J) for none of the above.

1/1p.	`4+9%9*4+9`	(A) -83	(B) -80	(C) 4	(D) 13	(E) 22	(F) 25	(G) 68	(H) 83
2/1p.	`8-7*0*9+1`	(A) -46	(B) 0	(C) 1	(D) 8	(E) 9	(F) 73	(G) 76	(H) 80
3/1p.	`3%7-3%9*3`	(A) -64	(B) -61	(C) -26	(D) -6	(E) -1	(F) 0	(G) 3	(H) 9
4/1p.	`8/2*7-4%2`	(A) -89	(B) -65	(C) 0	(D) 1	(E) 5	(F) 8	(G) 28	(H) 75
5/1p.	`1%6/7+2+7`	(A) -84	(B) -65	(C) -54	(D) -26	(E) 0	(F) 1	(G) 8	(H) 61
6/1p.	`8+9-2-2%7`	(A) -1	(B) 3	(C) 6	(D) 10	(E) 13	(F) 17	(G) 78	(H) 94
7/1p.	`5+9-6-1+4`	(A) -81	(B) -43	(C) -23	(D) 5	(E) 11	(F) 31	(G) 61	(H) 63
8/1p.	`9+5/2-6+0`	(A) -87	(B) -81	(C) -78	(D) -47	(E) -4	(F) 5	(G) 7	(H) 85
9/1p.	`4%1-3/3+1`	(A) -96	(B) -77	(C) -67	(D) -55	(E) -51	(F) -2	(G) -1	(H) 1
10/1p.	`2-3-4*0*8`	(A) -70	(B) -22	(C) -1	(D) 0	(E) 2	(F) 16	(G) 54	(H) 82
11/1p.	`5-4>4<=4+1`	(A) 0	(B) 1	(C) 3	(D) 4	(E) 23	(F) 54	(G) 62	(H) 63
12/1p.	`7-6==7>5-3`	(A) -43	(B) 0	(C) 1	(D) 4	(E) 7	(F) 10	(G) 55	(H) 63
13/1p.	`3-9+6>=0<4`	(A) -71	(B) -61	(C) -7	(D) 0	(E) 1	(F) 3	(G) 35	(H) 65
14/1p.	`4-2>7||4-9`	(A) -90	(B) -43	(C) -8	(D) 1	(E) 3	(F) 12	(G) 28	(H) 60
15/1p.	`6-6%4+7*3`	(A) -17	(B) -12	(C) -9	(D) 0	(E) 6	(F) 25	(G) 33	(H) 58
16/1p.	`7/5&&2||7+9`	(A) -53	(B) -2	(C) 1	(D) 5	(E) 7	(F) 10	(G) 16	(H) 35
17/1p.	`2+7-0>=2>=9`	(A) -22	(B) 0	(C) 1	(D) 3	(E) 8	(F) 9	(G) 38	(H) 60
18/1p.	`2%1+9*5%9`	(A) -94	(B) 0	(C) 2	(D) 10	(E) 24	(F) 45	(G) 70	(H) 90
19/1p.	`9*6+4<=6!=8`	(A) -75	(B) 1	(C) 9	(D) 47	(E) 51	(F) 54	(G) 55	(H) 72
20/1p.	`5*0&&0<=1-3`	(A) -45	(B) -15	(C) -3	(D) -2	(E) 0	(F) 2	(G) 60	(H) 69
21/1p.	`1-9<9!=9-5`	(A) -5	(B) -4	(C) 0	(D) 1	(E) 5	(F) 6	(G) 62	(H) 88
22/1p.	`1-5%7-4%3`	(A) -41	(B) -24	(C) -5	(D) -4	(E) -1	(F) 1	(G) 2	(H) 31
23/1p.	`1%3-7+1-3`	(A) -10	(B) -8	(C) -7	(D) -5	(E) -1	(F) 49	(G) 63	(H) 64
24/1p.	`6/3-7+6%6`	(A) -2	(B) 1	(C) 3	(D) 5	(E) 19	(F) 44	(G) 47	(H) 51
25/1p.	`4-8>3>8*9`	(A) -92	(B) -32	(C) -5	(D) 0	(E) 3	(F) 27	(G) 36	(H) 70
26/1p.	`5-7<8>=6-6`	(A) -30	(B) -6	(C) -5	(D) 0	(E) 4	(F) 5	(G) 11	(H) 92
27/1p.	`5+9<=8>5-1`	(A) -79	(B) -71	(C) -1	(D) 1	(E) 4	(F) 5	(G) 19	(H) 96
28/1p.	`9/4!=8<=1-7`	(A) -67	(B) -6	(C) -2	(D) 0	(E) 1	(F) 2	(G) 9	(H) 84
29/1p.	`8/4%5-6/2`	(A) -34	(B) -8	(C) -2	(D) -1	(E) 0	(F) 8	(G) 35	(H) 51
30/1p.	`4-6%9/6+7`	(A) -3	(B) -2	(C) -1	(D) 0	(E) 6	(F) 7	(G) 10	(H) 11
31/1p.	`6/3-0/1-5`	(A) -77	(B) -6	(C) -3	(D) -1	(E) 0	(F) 2	(G) 4	(H) 7
32/1p.	`2/4%2-2%4`	(A) -73	(B) -69	(C) -64	(D) -35	(E) -1	(F) 2	(G) 3	(H) 43
33/1p.	`6>5>=4-9-2`	(A) -56	(B) -11	(C) -7	(D) -6	(E) 1	(F) 36	(G) 79	(H) 94
34/1p.	`5*3+3||6>=7`	(A) -84	(B) -76	(C) 5	(D) 15	(E) 16	(F) 20	(G) 38	(H) 90

Matching: Which Perl regular expression commands have what meaning? (If no match mark J.)

(A) `<` (B) `[` (C) `\*` (D) `\D` (E) `\W` (F) `\cX` (G) `\d` (H) `\w` (I) `{`

35/1p. non-digit
36/1p. start of character class
37/1p. the null character
38/1p. not a word character
39/1p. start of multiplier

Matching: Which Perl regular expression commands have what meaning? (If no match mark J.)

(A) `(` (B) `>` (C) `\` (D) `\A` (E) `\a` (F) `sog` (G) `{n,m}` (H) `{n..m}` (I) `}`

40/1p. alarm (alert)
41/1p. at least n times, at most m
42/1p. start of group
43/1p. end of multiplier
44/1p. escape the next character

True or False: does the string match the regular expression?

45/1p. Does the empty string match the regular expression "`t*nn`"?

46/1p. Does the string "`dy`" match the regular expression "`dy+|df`"?

47/1p. Does the string "`sd`" match the regular expression "`sd`"?

48/1p. Does the string "`huhuza`" match the regular expression "`(hu)+za`"?

49/1p. Does the empty string match the regular expression "`hf|a?`"?

50/1p. Does the string "`hxxgg`" match the regular expression "`hx*g|g`"?

51/1p. Does the string "`fspp`" match the regular expression "`fs|p?`"?

52/1p. Does the string "`qf`" match the regular expression "`q+|f*`"?

53/1p. Does the string "`ggabab`" match the regular expression "`g?|ab*`"?

54/1p. Does the string "`xxbb`" match the regular expression "`(x?(bk)?)+`"?

55/1p. Does the string "`yk`" match the regular expression "`((wb)*|yp)?yk`"?

56/1p. Does the string "`thxdxd`" match the regular expression "`th|(xd*)?`"?

57/1p. Does the string "`zhgsgs`" match the regular expression "`tq*|zh(gs)*`"?

58/1p. Does the string "`ry`" match the regular expression "`gz|ry|ru?`"?

59/1p. Does the string "`bff`" match the regular expression "`(fh|bf)+`"?

60/1p. Does the empty string match the regular expression "`(zc)*tr|(n*)*`"?

61/1p. Does the string "`fzfzayy`" match the regular expression "`(fz)*ay+|zb`"?

62/1p. Does the string "`xx`" match the regular expression "`bk+|xx|(rr*|g)*`"?

63/1p. Does the string "`yaarp`" match the regular expression "`(u?|ya)*rp?`"?

64/1p. Does the string "`hsdasusuc`" match the regular expression "`(hs)?da|su?|(ck)*`"?

65/1p. Does the string "`cqyss`" match the regular expression "`(cq)+ys|sp*`"?

66/1p. Does the string "`pz`" match the regular expression "`(ns)?pz(ub)?|(hg)*`"?

67/1p. Does the string "`tfsysyqxx`" match the regular expression "`k*tf(sy)*qx+`"?

68/1p. Does the string "`rbdbd`" match the regular expression "`r+|yz(zq)?gs|(bd)?`"?

69/2p. 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 ) {
      for ( i = 1 ; i * i < n ; i += 1 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               g = n; while ( g > 1 ) {
                  simpleStatement;
               g -= 5; }
            } else {
               simpleStatement;
            }
         }
      }
   }
   return 0; }
```

70/2p. 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) $n\lg^2 n$ (G) n (I) $\sqrt{n}\lg n$
(B) $n^2\sqrt{n}$ (D) $n\sqrt{n}\lg n$ (F) $n\lg n$ (H) $\sqrt{n}\lg^2 n$ (J) 1

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

71/2p. 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) $\lg^3 n$ (I) $\lg n$
(B) $n^2\lg n$ (D) $n\sqrt{n}$ (F) $n\lg n$ (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( f = n ; f > 1 ; f /= 2 ) {
      if ( simpleCompare ) {
         for ( g = 1 ; g < n ; g *= 5 ) {
            if ( simpleCompare ) {
               for ( j = 1 ; j < n ; j++ ) {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         }
      }
   }
   return 0; }
```

72/2p. 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 (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 ) {
      h = n; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               for ( k = n ; k > 1 ; k-- ) {
                  simpleStatement;
               }
            }
         } else {
            simpleStatement;
         }
      h--; } while ( h > 1 );
   } else {
      b = 1; while ( b * b < n ) {
         d = 1; do {
            simpleStatement;
         d += 5; } while ( d * d < n );
      b += 10; }
   }
   return 0; }
```

73/2p. 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^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 ) {
      g = 1; do {
         for ( j = 1 ; j * j < n ; j++ ) {
            simpleStatement;
         }
      g++; } while ( g * g < n );
   } else {
      simpleStatement;
   }
   return 0; }
```

74/2p. 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) $n^5\sqrt{n}$ (C) $n^4\sqrt{n}$ (E) $n^2\sqrt{n}\lg^2 n$ (G) $n^2 \lg^4 n$ (I) n
(B) n^5 (D) $n^2\sqrt{n}\lg^3 n$ (F) $n^2\sqrt{n}\lg n$ (H) $n^2 \lg^2 n$ (J) $\lg^6 n$

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

75/2p. 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]);
   b = 1; while ( b * b < n ) {
      for ( e = n ; e > 1 ; e -= 3 ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      }
   b++; }
   return 0; }
```

76/2p. 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 ) {
      c = 1; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         } else {
            simpleStatement;
         }
      c *= 2; } while ( c < n );
   } else {
      if ( simpleCompare ) {
         b = n; while ( b > 1 ) {
            simpleStatement;
         b -= 2; }
      }
   }
   return 0; }
```

77/2p. 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) $\sqrt{n}$ (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]);
   for ( e = n ; e > 1 ; e /= 2 ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   }
   return 0; }
```

78/2p. 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^7 (C) $n^5 \lg^2 n$ (E) $n^2 \lg^3 n$ (G) n^2 (I) $n \lg^6 n$
(B) $n^6 \lg n$ (D) $n^2\sqrt{n}\lg^3 n$ (F) $n^2 \lg n$ (H) $n\sqrt{n}$ (J) $\sqrt{n}\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   e = 1; while ( e < n ) {
      f = n; while ( f > 1 ) {
         for ( h = n ; h > 1 ; h /= 2 ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  for ( i = 1 ; i < n ; i++ ) {
                     if ( simpleCompare ) {
                        for ( d = n ; d > 1 ; d-- ) {
                           if ( simpleCompare ) {
                              for ( b = 1 ; b < n ; b *= 5 ) {
                                 for ( j = 1 ; j < n ; j++ ) {
                                    simpleStatement;
                                 }
                              }
                           }
                        }
                     }
                  }
               }
            }
         }
      f -= 3; }
   e += 1; }
   return 0; }
```

79/2p. 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) $n^5 \lg n$ (C) $n^3\sqrt{n}\lg^2 n$ (E) $n^3 \lg^3 n$ (G) $n \lg^5 n$ (I) $\lg n$
(B) $n^4 \lg n$ (D) $n^3\sqrt{n}\lg n$ (F) $n\sqrt{n}$ (H) $\sqrt{n}\lg^5 n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( i = 1 ; i * i < n ; i++ ) {
         g = 1; while ( g < n ) {
            h = 1; do {
               for ( b = 1 ; b * b < n ; b += 3 ) {
                  for ( e = 1 ; e < n ; e += 1 ) {
                     j = 1; while ( j < n ) {
                        simpleStatement;
                     j *= 5; }
                  }
               }
            h++; } while ( h * h < n );
         g += 1; }
      }
   }
   return 0; }
```

80/2p. 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^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 ) {
      for ( c = 1 ; c < n ; c *= 2 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   } else {
      a = n; do {
         e = 1; do {
            simpleStatement;
         e += 2; } while ( e * e < n );
      a /= 2; } while ( a > 1 );
   }
   return 0; }
```

81/2p. 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) $n^3\sqrt{n}\lg n$ (C) $n^2\lg n$ (E) $n\sqrt{n}$ (G) $n\lg n$ (I) $\lg^2 n$
(B) $n^2\sqrt{n}\lg n$ (D) $n\sqrt{n}\lg^3 n$ (F) $n\lg^4 n$ (H) $\lg^5 n$ (J) $\lg n$

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

82/2p. 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) $n \lg n$ (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) $\lg n$

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

83/2p. 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^8 (C) $n^6\sqrt{n}\lg n$ (E) $n^6\lg^2 n$ (G) $n^4\sqrt{n}$ (I) $n^2\sqrt{n}\lg^5 n$
(B) n^7 (D) $n^6\sqrt{n}$ (F) $n^4\sqrt{n}\lg^2 n$ (H) $n^3\sqrt{n}\lg n$ (J) n^2

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

84/2p. 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 ) {
            if ( simpleCompare ) {
               k = 1; while ( k * k < n ) {
                  if ( simpleCompare ) {
                     if ( simpleCompare ) {
                        simpleStatement;
                     } else {
                        simpleStatement;
                     }
                  }
               k++; }
            } else {
               c = 1; do {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               c++; } while ( c * c < n );
            }
         } else {
            if ( simpleCompare ) {
               b = 1; do {
                  e = 1; do {
                     if ( simpleCompare ) {
                        if ( simpleCompare ) {
                           simpleStatement;
                        } else {
                           simpleStatement;
                        }
                     } else {
                        simpleStatement;
                     }
                  e += 5; } while ( e * e < n );
               b += 3; } while ( b * b < n );
            }
         }
      }
   }
   return 0; }
```

Total points 100.

Answer Key (points per line)

1 (1).	D (13)	43 (1).	I
2 (1).	E (9)	44 (1).	C
3 (1).	D (-6)	45 (1).	false
4 (1).	G (28)	46 (1).	true
5 (1).	J (9)	47 (1).	true
6 (1).	E (13)	48 (1).	true
7 (1).	E (11)	49 (1).	true
8 (1).	F (5)	50 (1).	false
9 (1).	J (0)	51 (1).	false
10 (1).	C (-1)	52 (1).	false
11 (1).	B (1)	53 (1).	false
12 (1).	C (1)	54 (1).	false
13 (1).	E (1)	55 (1).	true
14 (1).	D (1)	56 (1).	false
15 (1).	F (25)	57 (1).	true
16 (1).	C (1)	58 (1).	true
17 (1).	B (0)	59 (1).	false
18 (1).	B (0)	60 (1).	true
19 (1).	B (1)	61 (1).	true
20 (1).	E (0)	62 (1).	true
21 (1).	D (1)	63 (1).	false
22 (1).	C (-5)	64 (1).	false
23 (1).	B (-8)	65 (1).	false
24 (1).	J (-5)	66 (1).	true
25 (1).	D (0)	67 (1).	true
26 (1).	J (1)	68 (1).	false
27 (1).	J (0)	69 (2).	D ($n\sqrt{n}$)
28 (1).	E (1)	70 (2).	E ($n \lg^2 n$)
29 (1).	D (-1)	71 (2).	E ($n \lg^2 n$)
30 (1).	G (10)	72 (2).	B (n^2)
31 (1).	C (-3)	73 (2).	E (n)
32 (1).	J (-2)	74 (2).	G ($n^2 \lg^4 n$)
33 (1).	E (1)	75 (2).	D ($n\sqrt{n}$)
34 (1).	J (1)	76 (2).	F (n)
35 (1).	D	77 (2).	I ($\lg n$)
36 (1).	B	78 (2).	C ($n^5 \lg^2 n$)
37 (1).	J	79 (2).	D ($n^3\sqrt{n} \lg n$)
38 (1).	E	80 (2).	F ($\sqrt{n} \lg n$)
39 (1).	I	81 (2).	C ($n^2 \lg n$)
40 (1).	E	82 (2).	A (n^3)
41 (1).	G	83 (2).	H ($n^3\sqrt{n} \lg n$)
42 (1).	A	84 (2).	F (n)

Total points 100.