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 Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

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

2/2p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   } else {
      if ( simpleCompare ) {
         simpleStatement;
      } else {
         simpleStatement;
      }
   }
   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^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( g = 1 ; g < n ; g++ ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      } else {
         for ( c = 1 ; c < n ; c++ ) {
            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^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   h = n; while ( h > 1 ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            }
         } else {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   h -= 10; }
   return 0; }
```

5/2p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

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

6/2p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               for ( k = n ; k > 1 ; k -= 10 ) {
                  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^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   e = 1; do {
      for ( b = n ; b > 1 ; b -= 2 ) {
         c = n; while ( c > 1 ) {
            if ( simpleCompare ) {
               for ( h = 1 ; h < n ; h++ ) {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         c -= 3; }
      }
   e++; } while ( e < 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^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

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

9/2p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in Θ(1) time.
(A) n^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( e = n ; e > 1 ; e -= 5 ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               for ( a = n ; a > 1 ; a-- ) {
                  j = 1; while ( j < n ) {
                     simpleStatement;
                  j++; }
               }
            } else {
               simpleStatement;
            }
         } else {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      } else {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               f = n; while ( f > 1 ) {
                  simpleStatement;
               f -= 3; }
            } else {
               simpleStatement;
            }
         } 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^9 (C) n^7 (E) n^5 (G) n^3 (I) n
(B) n^8 (D) n^6 (F) n^4 (H) n^2 (J) 1

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

Total points 20.

Answer Key (points per line)

1 (2).	I (n)
2 (2).	J (1)
3 (2).	H (n^2)
4 (2).	I (n)
5 (2).	I (n)
6 (2).	I (n)
7 (2).	F (n^4)
8 (2).	G (n^3)
9 (2).	G (n^3)
10 (2).	F (n^4)

Total points 20.