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]);
   k = n; while ( k > 1 ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   k -= 5; }
   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 ) {
      for ( i = 1 ; i < n ; i += 1 ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            }
         }
      }
   } else {
      if ( simpleCompare ) {
         d = n; do {
            simpleStatement;
         d -= 3; } while ( d > 1 );
      } 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]);
   if ( simpleCompare ) {
      k = 1; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            }
         } else {
            if ( simpleCompare ) {
               simpleStatement;
            }
         }
      k += 1; } while ( k < n );
   }
   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]);
   if ( simpleCompare ) {
      k = n; do {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  for ( d = n ; d > 1 ; d -= 5 ) {
                     g = 1; do {
                        simpleStatement;
                     g++; } while ( g < n );
                  }
               } else {
                  simpleStatement;
               }
            }
         }
      k--; } while ( k > 1 );
   }
   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 ) {
      a = n; do {
         for ( g = n ; g > 1 ; g -= 10 ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      a -= 2; } while ( a > 1 );
   } else {
      i = n; do {
         simpleStatement;
      i -= 5; } while ( i > 1 );
   }
   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]);
   for ( e = n ; e > 1 ; e -= 10 ) {
      c = 1; do {
         k = 1; while ( k < n ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            } else {
               simpleStatement;
            }
         k++; }
      c += 10; } while ( c < n );
   }
   return 0; }
```

7/2p. 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]);
   f = 1; while ( f < n ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            e = 1; do {
               simpleStatement;
            e += 1; } while ( e < n );
         } else {
            simpleStatement;
         }
      }
   f += 3; }
   return 0; }
```

8/2p. 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 ) {
      f = 1; do {
         if ( simpleCompare ) {
            d = n; do {
               j = 1; do {
                  a = 1; while ( a < n ) {
                     b = 1; do {
                        simpleStatement;
                     b++; } while ( b < n );
                  a += 1; }
               j += 3; } while ( j < n );
            d--; } while ( d > 1 );
         } else {
            i = 1; while ( i < n ) {
               if ( simpleCompare ) {
                  simpleStatement;
               }
            i += 5; }
         }
      f++; } while ( f < n );
   }
   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]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            if ( simpleCompare ) {
               for ( j = 1 ; j < n ; j += 10 ) {
                  if ( simpleCompare ) {
                     if ( simpleCompare ) {
                        if ( simpleCompare ) {
                           for ( k = n ; k > 1 ; k-- ) {
                              simpleStatement;
                           }
                        } else {
                           simpleStatement;
                        }
                     } else {
                        simpleStatement;
                     }
                  }
               }
            }
         } else {
            i = 1; while ( i < n ) {
               for ( g = 1 ; g < n ; g += 2 ) {
                  if ( simpleCompare ) {
                     for ( f = 1 ; f < n ; f += 2 ) {
                        e = n; do {
                           simpleStatement;
                        e--; } while ( e > 1 );
                     }
                  } else {
                     simpleStatement;
                  }
               }
            i += 5; }
         }
      }
   }
   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]);
   e = n; while ( e > 1 ) {
      if ( simpleCompare ) {
         for ( g = 1 ; g < n ; g += 3 ) {
            if ( simpleCompare ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     i = 1; do {
                        for ( a = 1 ; a < n ; a += 5 ) {
                           simpleStatement;
                        }
                     i++; } while ( i < n );
                  } else {
                     simpleStatement;
                  }
               } else {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               }
            }
         }
      } else {
         b = n; while ( b > 1 ) {
            j = n; do {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     for ( d = n ; d > 1 ; d-- ) {
                        simpleStatement;
                     }
                  } else {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            j -= 1; } while ( j > 1 );
         b--; }
      }
   e--; }
   return 0; }
```

Total points 20.

Answer Key (points per line)

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

Total points 20.