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

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

Total points 20.

Answer Key (points per line)

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

Total points 20.