

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

Total points 20.

Answer Key (points per line)

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

Total points 20.