Keep this test booklet when you are done. **Closed book. No notes.** Work strictly from memory. **No calculators. Scratch paper okay.**

Section I: Record all answers on the bubble sheet.

On the following printf questions you are given a list of inputs. For each problem line determine which printf statement created the accompanying outputs. (␣ means space.)

Which of these printf statements created the outputs shown for each problem below? (`x` is `int x;`)

(A) `printf("␣␣␣␣␣%+4d␣",x);` (D) `printf("␣␣␣␣%5d␣",x);` (G) `printf("␣%-␣6d␣␣␣",x);`
(B) `printf("␣␣␣␣␣%03d␣␣",x);` (E) `printf("␣␣%-␣8d",x);` (H) `printf("%␣08d␣␣",x);`
(C) `printf("␣␣␣␣%-5d␣",x);` (F) `printf("␣␣%8d",x);` (I) `printf("%-␣10d",x);`

inputs:	1	-4	1442412769	-1758341286
1/2p.	␣␣␣␣1␣␣␣␣␣	␣␣␣␣-4␣␣␣␣	␣␣␣␣1442412769␣	␣␣␣␣-1758341286␣
2/2p.	␣␣␣␣␣␣␣+1␣	␣␣␣␣␣␣␣-4␣	␣␣␣␣␣+1442412769␣	␣␣␣␣␣-1758341286␣
3/2p.	␣␣1␣␣␣␣␣␣	␣-4␣␣␣␣␣␣	␣␣1442412769␣␣␣	␣-1758341286␣␣␣
4/2p.	␣1␣␣␣␣␣␣␣	-4␣␣␣␣␣␣␣␣	␣1442412769	-1758341286

Which of these printf statements created the outputs shown for each problem below? (`x` is `char * x;`)

(A) `printf("␣␣␣␣␣%s␣",x);` (D) `printf("␣␣␣%1s␣␣␣",x);` (G) `printf("%4s␣␣␣",x);`
(B) `printf("␣␣␣␣%1s␣",x);` (E) `printf("␣%-2s␣␣␣␣",x);` (H) `printf("%5s␣␣",x);`
(C) `printf("␣␣␣%-3s␣",x);` (F) `printf("%-7s",x);` (I) `printf("%7s",x);`

inputs:	""	"c"	"pq"	"gdkq"	"vfydqv"	"hjpdwzcl"
5/2p.	␣␣␣␣␣␣␣	␣␣␣c␣␣␣	␣␣␣pq␣␣	␣␣␣gdkq␣	␣␣␣vfydqv␣	␣␣␣hjpdwzcl␣
6/2p.	␣␣␣␣␣␣␣	␣␣␣␣␣␣␣c␣	␣␣␣␣␣␣␣pq␣	␣␣␣␣␣␣␣gdkq␣	␣␣␣␣␣␣␣vfydqv␣	␣␣␣␣␣␣␣hjpdwzcl␣
7/2p.	␣␣␣␣␣␣␣	␣␣␣c␣␣␣	␣␣␣pq␣␣␣	␣␣␣gdkq␣␣␣	␣␣␣vfydqv␣␣␣	␣␣␣hjpdwzcl␣␣␣
8/2p.	␣␣␣␣␣␣␣	␣␣␣␣␣␣␣c	␣␣␣␣␣pq	␣␣␣gdkq	␣vfydqv	hjpdwzcl

Which of these printf statements created the outputs shown for each problem below? (`x` is `double x;`)

(A) `printf("␣␣%011.6f␣",x);` (D) `printf("␣%0+10.6f␣␣␣",x);` (G) `printf("%+014.0f",x);`
(B) `printf("␣␣%10.6f␣␣",x);` (E) `printf("␣%12.0f␣",x);` (H) `printf("%+14.4f",x);`
(C) `printf("␣%+11.6f␣␣",x);` (F) `printf("␣%13.4f",x);` (I) `printf("%12.6f␣␣",x);`

inputs:	9	-7.98	3.4256	-341521.978540
9/2p.	+0000000000009	-0000000000008	+0000000000003	-0000000341522
10/2p.	␣␣␣␣␣␣␣␣9.0000	␣␣␣␣␣␣␣␣-7.9800	␣␣␣␣␣␣␣␣3.4256	␣␣-341521.9785
11/2p.	␣␣␣+9.000000␣␣	␣␣␣-7.980000␣␣	␣␣␣+3.425600␣␣	␣-341521.978540␣␣
12/2p.	␣␣␣␣␣␣␣+9.0000	␣␣␣␣␣␣␣␣-7.9800	␣␣␣␣␣␣␣␣+3.4256	␣␣-341521.9785

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

13/1p.	`6*1-2%1+2`	(A) -6	(B) 0	(C) 1	(D) 2	(E) 8	(F) 12	(G) 18	(H) 42
14/1p.	`9%6/2+8-3`	(A) -42	(B) -36	(C) -17	(D) -3	(E) 1	(F) 5	(G) 6	(H) 69
15/1p.	`4+1==9==0*8`	(A) 0	(B) 4	(C) 5	(D) 8	(E) 12	(F) 32	(G) 37	(H) 40
16/1p.	`6%5+3*5/2`	(A) -91	(B) -35	(C) 3	(D) 6	(E) 8	(F) 10	(G) 12	(H) 15
17/1p.	`8*3-8/5*4`	(A) -32	(B) -19	(C) -2	(D) 0	(E) 12	(F) 20	(G) 24	(H) 92
18/1p.	`9-6-6%7+4`	(A) -7	(B) -3	(C) 1	(D) 5	(E) 6	(F) 8	(G) 13	(H) 24
19/1p.	`8-9*2/3-3`	(A) -7	(B) -4	(C) -1	(D) 3	(E) 5	(F) 11	(G) 35	(H) 39
20/1p.	`9-6*3+6-3`	(A) -48	(B) -27	(C) -25	(D) -18	(E) -12	(F) -6	(G) 18	(H) 24
21/1p.	`3-4&&3<6*6`	(A) -3	(B) 1	(C) 2	(D) 6	(E) 12	(F) 16	(G) 33	(H) 83
22/1p.	`8-5%9%2-3`	(A) -3	(B) 0	(C) 4	(D) 5	(E) 8	(F) 9	(G) 10	(H) 11
23/1p.	`8/1!=6&&3+9`	(A) -58	(B) 0	(C) 1	(D) 8	(E) 17	(F) 27	(G) 53	(H) 91
24/1p.	`7+8==1>0-7`	(A) -96	(B) -94	(C) -38	(D) -7	(E) -6	(F) 0	(G) 1	(H) 7
25/1p.	`6+0-9%1*2`	(A) -88	(B) -18	(C) 0	(D) 1	(E) 5	(F) 6	(G) 7	(H) 12
26/1p.	`8/5+1-5%7`	(A) -81	(B) -67	(C) -4	(D) -3	(E) 1	(F) 3	(G) 4	(H) 20
27/1p.	`5*2!=1>=5+6`	(A) 0	(B) 3	(C) 5	(D) 6	(E) 7	(F) 11	(G) 30	(H) 35
28/1p.	`7<=2<=4+6-1`	(A) -91	(B) -40	(C) -1	(D) 1	(E) 5	(F) 6	(G) 26	(H) 93
29/1p.	`6/5%4-7*8`	(A) -55	(B) -51	(C) -50	(D) -48	(E) -8	(F) -1	(G) 4	(H) 91
30/1p.	`6+6*1%5/5`	(A) -74	(B) -53	(C) -6	(D) 0	(E) 2	(F) 6	(G) 7	(H) 14
31/1p.	`5/5*9%6-4`	(A) -91	(B) -4	(C) -3	(D) -1	(E) 0	(F) 1	(G) 5	(H) 68

How many times does the body of the loop execute? (Mark 9 if 9 or more.)

32/2p. `int t; for( t=9; t>2; ++t ) body;`

33/2p. `int s=-10; while( --s > -16 ) body;`

34/2p. `int f=7; while( f-- >= 4 ) body;`

35/2p. `int z; for( z=-5; z!=0; z++ ) body;`

36/2p. `int g=8; do body; while( ++g != 15 );`

37/2p. `int m=8; do body; while( m++ < 15 );`

38/2p. `int a=9; while( a-- != 6 ) body;`

39/2p. `int p=-2; while( ++p <= 4 ) body;`

40/3p. 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) $n\sqrt{n}\lg n$ (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 ( f = 1 ; f * f < n ; f++ ) {
         simpleStatement;
      }
   } else {
      simpleStatement;
   }
   return 0; }
```

41/3p. 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) $n\sqrt{n}\lg n$ (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 ) {
      d = 1; while ( d < n ) {
         e = n; while ( e > 1 ) {
            simpleStatement;
         e--; }
      d *= 5; }
   } else {
      simpleStatement;
   }
   return 0; }
```

42/3p. 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) 1

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

43/5p. 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 (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 ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   } else {
      simpleStatement;
   }
   return 0; }
```

44/5p. 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) $\lg n$
(B) $n^2\lg n$ (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            i = 1; do {
               simpleStatement;
            i += 5; } while ( i < n );
         } else {
            simpleStatement;
         }
      } else {
         simpleStatement;
      }
   } else {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      }
   }
   return 0; }
```

45/10p. 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) $\lg n$
(B) $n^2\lg n$ (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) 1

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

Section II: List regex answers on a single well-labeled sheet of paper.

46/4p. Let L be the language accepted by the regular expression "`(bbb)+b?`". List the shortest five (or all) strings in L.

47/4p. Let L be the language accepted by the regular expression "`(abaabb)+bb`". List the shortest five (or all) strings in L.

48/4p. Let L be the language accepted by the regular expression "`(bab|bba)b*?`". List the shortest five (or all) strings in L.

49/4p. Let L be the language accepted by the regular expression "`(abb|a*)a*bba*abaaab(abbaa)+baa*`". List the shortest five (or all) strings in L.

Section III: Write each program on a separate well-labeled sheet of paper.

50/8p. Write this program on a separate sheet of paper. Use one side of one sheet. Label it clearly. Accept two integers, a and b, from the command line (argv). Print their difference (a minus b).

51/8p. Write this program on a separate sheet of paper. Use one side of one sheet. Label it clearly. Prompt for and accept two integers, a and b. Print the smaller (min).

52/15p. Write this program on a separate sheet of paper. Use one side of one sheet. Label it clearly. Prompt for and accept a string. Print out all its substrings, duplicates okay. For cat print (in any order) c, a, t, ca, at, cat.

53/15p. Write this program on a separate sheet of paper. Use one side of one sheet. Label it clearly. Prompt for and accept a start time and a running time in HH:MM:SS format. Print the ending time. Example: 1:39:44 + 2:11:31 = 3:51:15.

Total points 150.

Answer Key (points per line)

1	(2).	C
2	(2).	A
3	(2).	G
4	(2).	I
5	(2).	C
6	(2).	A
7	(2).	D
8	(2).	I
9	(2).	G
10	(2).	F
11	(2).	C
12	(2).	H
13	(1).	E (8)
14	(1).	G (6)
15	(1).	J (1)
16	(1).	E (8)
17	(1).	F (20)
18	(1).	C (1)
19	(1).	C (-1)
20	(1).	F (-6)
21	(1).	B (1)
22	(1).	C (4)
23	(1).	C (1)
24	(1).	F (0)
25	(1).	F (6)
26	(1).	D (-3)
27	(1).	J (1)
28	(1).	D (1)
29	(1).	A (-55)
30	(1).	F (6)
31	(1).	D (-1)
32	(2).	9
33	(2).	5
34	(2).	4
35	(2).	5
36	(2).	7
37	(2).	8
38	(2).	3
39	(2).	6
40	(3).	G ($\sqrt{n}$)
41	(3).	D ($n \lg n$)
42	(3).	J (1)
43	(5).	J (1)
44	(5).	F (n)
45	(10).	H ($\sqrt{n}$)

Total points 150.