Do NOT write on this test. Record all answers on the bubble sheet. **Closed book. No notes.** Work strictly from memory. **No calculators. No time limit. Scratch paper okay.**

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("␣␣␣␣␣%␣d␣␣␣␣",x);   (D) printf("␣%␣9d",x);         (G) printf("%-␣10d",x);
(B) printf("␣␣␣%+-4d␣␣␣",x);    (E) printf("␣%+9d",x);         (H) printf("%-+10d",x);
(C) printf("␣␣␣%7d",x);         (F) printf("␣%0␣6d␣␣␣",x);     (I) printf("%0+10d",x);
```

```
 inputs:  5            -1           1215293624         -1695290415
 1/2p.    ␣5␣␣␣␣␣␣␣␣   -1␣␣␣␣␣␣␣␣   ␣1215293624        -1695290415
 2/2p.    +5␣␣␣␣␣␣␣␣   -1␣␣␣␣␣␣␣␣   +1215293624        -1695290415
 3/2p.    ␣␣␣+5␣␣␣␣␣   ␣␣␣-1␣␣␣␣␣   ␣␣␣+1215293624␣␣␣  ␣␣␣-1695290415␣␣␣
 4/2p.    ␣␣␣␣␣␣5␣␣␣␣  ␣␣␣␣␣-1␣␣␣␣  ␣␣␣␣␣␣1215293624␣␣␣␣  ␣␣␣␣␣-1695290415␣␣␣␣
```

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

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

```
 inputs:  “”     “c”     “wd”     “cpwx”     “fzcdfd”     “xfqccwyv”
 5/2p.    ␣␣␣␣   ␣c␣␣␣   ␣wd␣␣␣   ␣cpwx␣␣␣   ␣fzcdfd␣␣␣   ␣xfqccwyv␣␣␣
 6/2p.    ␣␣␣␣   c␣␣␣    wd␣␣     cpwx       fzcdfd       xfqccwyv
 7/2p.    ␣␣␣␣   ␣␣c␣    ␣wd␣     ␣cpwx␣     ␣fzcdfd␣     ␣xfqccwyv␣
 8/2p.    ␣␣␣␣   ␣␣␣c␣   ␣␣␣wd␣   ␣␣␣cpwx␣   ␣␣␣fzcdfd␣   ␣␣␣xfqccwyv␣
```

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

```
(A) printf("␣␣␣%+10.4f␣",x);   (D) printf("␣%+11.4f␣␣",x);   (G) printf("%+14.4f",x);
(B) printf("␣␣%+010f␣␣",x);    (E) printf("␣%10.6f␣␣␣",x);   (H) printf("%14.4f",x);
(C) printf("␣␣%+011.6f␣",x);   (F) printf("␣%11.6f␣␣",x);    (I) printf("%14.6f",x);
```

```
 inputs:  8                2.51             4.4789           -256876.348229
 9/2p.    ␣␣␣␣␣␣+8.0000␣   ␣␣␣␣␣␣+2.5100␣   ␣␣␣␣␣␣+4.4789␣   ␣␣␣-256876.3482␣
 10/2p.   ␣␣␣␣␣␣␣+8.0000   ␣␣␣␣␣␣␣+2.5100   ␣␣␣␣␣␣␣+4.4789   ␣␣-256876.3482
 11/2p.   ␣␣␣␣8.000000␣␣   ␣␣␣␣2.510000␣␣   ␣␣␣␣4.478900␣␣   ␣-256876.348229␣␣
 12/2p.   ␣␣␣␣␣␣8.000000   ␣␣␣␣␣␣2.510000   ␣␣␣␣␣␣4.478900   -256876.348229
```

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

13/1p.	3%7%9-9*7	(A) -60	(B) -53	(C) -51	(D) -44	(E) -23	(F) -18	(G) -7	(H) 72
14/1p.	3-6-7!=9>=1	(A) -70	(B) -57	(C) -4	(D) -2	(E) 0	(F) 1	(G) 2	(H) 81
15/1p.	8+7||2<2*2	(A) 0	(B) 1	(C) 2	(D) 9	(E) 10	(F) 19	(G) 52	(H) 57
16/1p.	0-5/3%1*4	(A) -85	(B) -62	(C) -36	(D) -2	(E) -1	(F) 0	(G) 2	(H) 50
17/1p.	6-4/1%7-6	(A) -53	(B) -39	(C) -13	(D) -1	(E) 0	(F) 6	(G) 7	(H) 8
18/1p.	6/4/1-7+4	(A) -10	(B) -6	(C) -2	(D) -1	(E) 2	(F) 3	(G) 6	(H) 51
19/1p.	8-8-3+8-5	(A) -69	(B) -37	(C) -16	(D) -10	(E) -5	(F) 0	(G) 6	(H) 16
20/1p.	7+6<=4>5-2	(A) -88	(B) -3	(C) -2	(D) -1	(E) 1	(F) 5	(G) 52	(H) 63
21/1p.	3+3%4*5/2	(A) -81	(B) 3	(C) 4	(D) 5	(E) 6	(F) 9	(G) 10	(H) 12
22/1p.	5-2%1/5-8	(A) -54	(B) -8	(C) -7	(D) -3	(E) 0	(F) 5	(G) 11	(H) 13
23/1p.	5/6<5<=3-5	(A) -22	(B) -19	(C) -4	(D) -1	(E) 0	(F) 20	(G) 21	(H) 31
24/1p.	8+5%4+5/7	(A) -78	(B) -70	(C) -42	(D) 1	(E) 2	(F) 8	(G) 9	(H) 12
25/1p.	6+5/8%2*5	(A) -16	(B) 0	(C) 1	(D) 5	(E) 30	(F) 52	(G) 54	(H) 81
26/1p.	7-4-5/8+1	(A) -1	(B) 0	(C) 3	(D) 4	(E) 7	(F) 8	(G) 9	(H) 13
27/1p.	4%9+4/3*5	(A) -57	(B) -48	(C) 0	(D) 4	(E) 5	(F) 9	(G) 24	(H) 25
28/1p.	9>4&&8+3*8	(A) -84	(B) -36	(C) -26	(D) -13	(E) 0	(F) 1	(G) 25	(H) 32
29/1p.	3%9+7+4*7	(A) 13	(B) 21	(C) 31	(D) 38	(E) 49	(F) 80	(G) 84	(H) 98
30/1p.	5-1||9!=0-9	(A) -94	(B) -8	(C) -5	(D) 4	(E) 13	(F) 26	(G) 65	(H) 85
31/1p.	5-9%3%8*7	(A) 2	(B) 3	(C) 5	(D) 7	(E) 14	(F) 17	(G) 35	(H) 54

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

32/2p. `int f; for( f=10; f>=7; f-- ) body;`

33/2p. `int u=-4; do body; while( u++ != 2 );`

34/2p. `int c=8; do body; while( c-- >= 6 );`

35/2p. `int i=-2; do body; while( --i != -10 );`

36/2p. `int h=-3; while( h-- < 0 ) body;`

37/2p. `int a; for( a=-6; a<=-7; a++ ) body;`

38/2p. `int z=8; do body; while( z-- > 5 );`

39/2p. `int s=8; while( s-- != 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) $n^2 \lg n$ (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 ( c = 1 ; c < n ; c *= 2 ) {
         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^2 (E) $n\sqrt{n}$ (G) n (I) $\lg^2 n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}\lg n$ (F) $n \lg n$ (H) $\sqrt{n}\lg n$ (J) 1

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      e = n; while ( e > 1 ) {
         i = n; while ( i > 1 ) {
            simpleStatement;
         i -= 5; }
      e--; }
   } 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) $\lg n$
(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]);
   f = n; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            simpleStatement;
         } else {
            simpleStatement;
         }
      }
   f--; } while ( f > 1 );
   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) $n^2 \lg n$ (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 ) {
      j = n; do {
         e = 1; while ( e < n ) {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         e += 2; }
      j /= 3; } while ( j > 1 );
   } else {
      d = n; while ( d > 1 ) {
         for ( b = n ; b > 1 ; b /= 3 ) {
            simpleStatement;
         }
      d--; }
   }
   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) $n^3 \lg n$ (C) $n^2 \lg n$ (E) $n \lg^3 n$ (G) n (I) $\sqrt{n}$
(B) $n^2 \lg^2 n$ (D) $n\sqrt{n}$ (F) $n \lg n$ (H) $\sqrt{n}\lg^3 n$ (J) $\lg^2 n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( c = n ; c > 1 ; c-- ) {
      g = n; while ( g > 1 ) {
         for ( a = 1 ; a < n ; a *= 3 ) {
            for ( i = n ; i > 1 ; i -= 3 ) {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            }
         }
      g--; }
   }
   return 0; }
```

45/5p. Give a tight big-oh Θ() bound on the running time T(n) of this program.
Assume `atoi`, `simpleStatement`, and `simpleCompare` each run in $\Theta(1)$ time.

(A) n^3	(C) $n\sqrt{n}\lg n$	(E) $n\lg^2 n$	(G) n	(I) $\lg^2 n$
(B) $n^2\lg n$	(D) $n\sqrt{n}$	(F) $n\lg n$	(H) $\sqrt{n}\lg^2 n$	(J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   k = 1; do {
      if ( simpleCompare ) {
         if ( simpleCompare ) {
            e = n; while ( e > 1 ) {
               f = 1; do {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               f++; } while ( f * f < n );
            e /= 2; }
         } else {
            for ( g = 1 ; g * g < n ; g += 2 ) {
               i = n; while ( i > 1 ) {
                  simpleStatement;
               i--; }
            }
         }
      } else {
         if ( simpleCompare ) {
            h = n; do {
               if ( simpleCompare ) {
                  if ( simpleCompare ) {
                     simpleStatement;
                  } else {
                     simpleStatement;
                  }
               } else {
                  simpleStatement;
               }
            h--; } while ( h > 1 );
         } else {
            if ( simpleCompare ) {
               simpleStatement;
            } else {
               simpleStatement;
            }
         }
      }
   k *= 5; } while ( k < n );
   return 0; }
```

Matching: Which Perl regular expression commands have what meaning? (If no match mark J.)

(A) `[` (B) `\D` (C) `\S` (D) `\W` (E) `\d` (F) `\i` (G) `\s` (H) `\t` (I) `{`

46/1p. digit
47/1p. whitespace
48/1p. tab
49/1p. non a word character
50/1p. start of character class

True or False: does the string match the regular expression?

51/1p. Does the empty string match the regular expression “`b|c`”?

52/1p. Does the empty string match the regular expression “`n+|zt`”?

53/1p. Does the string “`zz`” match the regular expression “`q+|z?`”?

54/1p. Does the string “`ppngz`” match the regular expression “`p?|n|gz`”?

55/1p. Does the string “`qqqddd`” match the regular expression “`q+(qd?)?`”?

56/1p. Does the string “`s`” match the regular expression “`x?s|w*`”?

57/1p. Does the string “`ffwhbb`” match the regular expression “`f?|w(hb)*|z`”?

58/1p. Does the string “`azkkk`” match the regular expression “`a|z(kk)+|ct`”?

59/1p. Does the string “`ncccddrdr`” match the regular expression “`((nc)*cd|dr*)*`”?

60/1p. Does the string “`zzz`” match the regular expression “`((pw)*z+)+(yz)*`”?

61/1p. Does the string “`xdxdgrsss`” match the regular expression “`(xd)+|g|((rs)+)*|c+`”?

62/1p. Does the string “`wcbwugr`” match the regular expression “`w+cb(wu)?t?gr`”?

Total points 100.

Answer Key (points per line)

1 (2).	G	32 (2).	4
2 (2).	H	33 (2).	7
3 (2).	B	34 (2).	4
4 (2).	A	35 (2).	8
5 (2).	F	36 (2).	9
6 (2).	G	37 (2).	0
7 (2).	E	38 (2).	4
8 (2).	A	39 (2).	4
9 (2).	A	40 (3).	I ($\lg n$)
10 (2).	G	41 (3).	C (n^2)
11 (2).	F	42 (3).	F (n)
12 (2).	I	43 (5).	D ($n \lg n$)
13 (1).	A (-60)	44 (5).	A ($n^3 \lg n$)
14 (1).	F (1)	45 (5).	C ($n\sqrt{n} \lg n$)
15 (1).	B (1)	46 (1).	E
16 (1).	F (0)	47 (1).	G
17 (1).	J (-4)	48 (1).	H
18 (1).	C (-2)	49 (1).	D
19 (1).	F (0)	50 (1).	A
20 (1).	J (0)	51 (1).	false
21 (1).	G (10)	52 (1).	false
22 (1).	D (-3)	53 (1).	false
23 (1).	E (0)	54 (1).	false
24 (1).	G (9)	55 (1).	false
25 (1).	J (6)	56 (1).	true
26 (1).	D (4)	57 (1).	false
27 (1).	F (9)	58 (1).	false
28 (1).	F (1)	59 (1).	false
29 (1).	D (38)	60 (1).	true
30 (1).	J (1)	61 (1).	false
31 (1).	C (5)	62 (1).	true

Total points 100.