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("␣␣␣%␣3d␣␣␣",x);`
(B) `printf("␣␣␣%+5d␣",x);`
(C) `printf("␣␣%-+6d␣",x);`
(D) `printf("␣␣%0+5d␣␣",x);`
(E) `printf("␣%+4d␣␣␣␣",x);`
(F) `printf("%-␣9d",x);`
(G) `printf("%07d␣␣",x);`
(H) `printf("%09d",x);`
(I) `printf("%9d",x);`

inputs:	4	-9	1718294580	-1569486855
1/2p.	␣␣+4␣␣␣␣␣	␣␣-9␣␣␣␣␣	␣␣+1718294580␣	␣␣-1569486855␣
2/2p.	0000004␣␣	-000009␣␣	1718294580␣␣	-1569486855␣␣
3/2p.	000000004	-00000009	1718294580	-1569486855
4/2p.	␣␣␣+4␣␣␣␣	␣␣␣-9␣␣␣␣	␣+1718294580␣␣␣␣	␣-1569486855␣␣␣␣

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

(A) `printf("␣␣␣␣%s␣␣␣",x);`
(B) `printf("␣␣%3s␣␣",x);`
(C) `printf("␣%-2s␣␣␣␣",x);`
(D) `printf("␣%-4s␣␣",x);`
(E) `printf("␣%5s␣",x);`
(F) `printf("␣%s␣␣␣␣␣␣",x);`
(G) `printf("%-6s␣",x);`
(H) `printf("%-7s",x);`
(I) `printf("%7s",x);`

inputs:	“”	“b”	“kl”	“bvkw”	“kvzzzj”	“jkkqlgxy”
5/2p.	␣␣␣␣␣␣␣	␣␣␣␣␣b␣	␣␣␣␣kl␣	␣␣bvkw␣	␣kvzzzj␣	␣jkkqlgxy␣
6/2p.	␣␣␣␣␣␣␣	␣b␣␣␣␣␣␣	␣kl␣␣␣␣␣␣	␣bvkw␣␣␣␣␣␣	␣kvzzzj␣␣␣␣␣␣	␣jkkqlgxy␣␣␣␣␣␣
7/2p.	␣␣␣␣␣␣␣	b␣␣␣␣␣␣	kl␣␣␣␣␣	bvkw␣␣␣	kvzzzj␣	jkkqlgxy
8/2p.	␣␣␣␣␣␣␣	␣b␣␣␣␣␣	␣kl␣␣␣␣	␣bvkw␣␣	␣kvzzzj␣␣	␣jkkqlgxy␣␣

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

(A) `printf("␣␣␣␣%+10f",x);`
(B) `printf("␣␣%+12.0f",x);`
(C) `printf("␣␣%011.4f␣",x);`
(D) `printf("␣␣%12.6f",x);`
(E) `printf("␣%+011f␣␣",x);`
(F) `printf("␣%012.6f␣",x);`
(G) `printf("␣%013f",x);`
(H) `printf("␣%10.4f␣␣␣",x);`
(I) `printf("%12f␣␣",x);`

inputs:	1	1.01	1.4417	-57726.004384
9/2p.	␣00001.000000␣	␣00001.010000␣	␣00001.441700␣	␣-57726.004384␣
10/2p.	␣␣␣␣␣1.0000␣␣␣	␣␣␣␣␣1.0100␣␣␣	␣␣␣␣␣1.4417␣␣␣	␣-57726.0044␣␣␣
11/2p.	␣000001.000000	␣000001.010000	␣000001.441700	␣-57726.004384
12/2p.	␣␣␣␣␣␣1.000000	␣␣␣␣␣␣1.010000	␣␣␣␣␣␣1.441700	␣␣-57726.004384

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

13/1p.	`7-2||6||9-2`	(A) -88	(B) -1	(C) 1	(D) 4	(E) 6	(F) 8	(G) 26	(H) 92
14/1p.	`5*8%9+5*1`	(A) 9	(B) 12	(C) 40	(D) 45	(E) 50	(F) 65	(G) 89	(H) 98
15/1p.	`2+4-5*0-8`	(A) -11	(B) -8	(C) -6	(D) -2	(E) 10	(F) 46	(G) 61	(H) 99
16/1p.	`1+2&&7==4-4`	(A) -9	(B) -6	(C) -4	(D) -3	(E) 0	(F) 1	(G) 48	(H) 88
17/1p.	`5/6-0%5+9`	(A) -80	(B) -28	(C) -9	(D) 0	(E) 9	(F) 14	(G) 40	(H) 77
18/1p.	`3-3%2/2+8`	(A) -53	(B) -45	(C) -5	(D) 0	(E) 8	(F) 11	(G) 92	(H) 98
19/1p.	`8*9||1<9-8`	(A) -7	(B) 0	(C) 3	(D) 8	(E) 47	(F) 81	(G) 83	(H) 96
20/1p.	`7<=4>=2-6-6`	(A) -77	(B) -12	(C) -6	(D) -5	(E) -1	(F) 0	(G) 1	(H) 73
21/1p.	`3+5*3+7+2`	(A) -50	(B) -28	(C) -13	(D) 33	(E) 49	(F) 55	(G) 63	(H) 82
22/1p.	`0-6-2/1+5`	(A) -13	(B) -9	(C) -3	(D) -2	(E) -1	(F) 0	(G) 3	(H) 89
23/1p.	`7*6%5-6-5`	(A) -40	(B) -35	(C) -9	(D) -4	(E) 0	(F) 2	(G) 6	(H) 14
24/1p.	`9/6/6%2+8`	(A) -81	(B) 0	(C) 1	(D) 8	(E) 9	(F) 17	(G) 53	(H) 87
25/1p.	`4-6/3*5%8`	(A) -15	(B) -6	(C) -5	(D) -1	(E) 2	(F) 3	(G) 7	(H) 10
26/1p.	`2/9>0<=8-4`	(A) -17	(B) -4	(C) -3	(D) -2	(E) -1	(F) 0	(G) 2	(H) 68
27/1p.	`2%1-3%6*6`	(A) -18	(B) -15	(C) -10	(D) 0	(E) 12	(F) 18	(G) 33	(H) 95
28/1p.	`4*7/2/6-5`	(A) -91	(B) -21	(C) -20	(D) -6	(E) -5	(F) -3	(G) 12	(H) 14
29/1p.	`6-1+3%2+9`	(A) -94	(B) -16	(C) -5	(D) -3	(E) 1	(F) 2	(G) 8	(H) 9
30/1p.	`9+8-3||8>=3`	(A) -65	(B) -60	(C) -28	(D) 9	(E) 10	(F) 16	(G) 17	(H) 54
31/1p.	`8+5/9-8-8`	(A) -83	(B) -55	(C) -15	(D) -8	(E) -2	(F) 1	(G) 5	(H) 8

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

32/2p. `int p=1; while( p++ < 3 ) body;`

33/2p. `int b=-5; do body; while( ++b <= -2 );`

34/2p. `int c; for( c=7; c>=3; --c ) body;`

35/2p. `int v=7; while( --v >= -1 ) body;`

36/2p. `int x=-1; do body; while( x-- > -6 );`

37/2p. `int q=-7; while( --q <= 0 ) body;`

38/2p. `int f=-1; while( ++f <= 4 ) body;`

39/2p. `int d=-8; do body; while( d-- != -8 );`

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^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 ) {
      b = n; do {
         simpleStatement;
      b /= 3; } while ( b > 1 );
   } 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) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg^2 n$
(B) $n^2 \lg n$ (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      k = 1; while ( k * k < n ) {
         j = n; do {
            simpleStatement;
         j--; } while ( j > 1 );
      k++; }
   } 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^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) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   for ( c = n ; c > 1 ; c /= 2 ) {
      if ( simpleCompare ) {
         for ( e = 1 ; e * e < n ; e++ ) {
            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) $n\lg n$ (G) $\sqrt{n}\lg n$ (I) $\lg^2 n$
(B) $n^2\lg n$ (D) $n\sqrt{n}$ (F) n (H) $\sqrt{n}$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   if ( simpleCompare ) {
      for ( e = 1 ; e < n ; e *= 2 ) {
         for ( c = 1 ; c < n ; c *= 3 ) {
            if ( simpleCompare ) {
               simpleStatement;
            }
         }
      }
   } else {
      if ( simpleCompare ) {
         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) $n^2\lg n$ (C) $n\sqrt{n}\lg n$ (E) $n\lg^2 n$ (G) n (I) $\lg^3 n$
(B) n^2 (D) $n\sqrt{n}$ (F) $n\lg n$ (H) $\sqrt{n}\lg n$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
   int n = atoi(argv[1]);
   c = 1; while ( c * c < n ) {
      if ( simpleCompare ) {
         for ( h = n ; h > 1 ; h-- ) {
            for ( i = 1 ; i * i < n ; i++ ) {
               if ( simpleCompare ) {
                  simpleStatement;
               } else {
                  simpleStatement;
               }
            }
         }
      } else {
         if ( simpleCompare ) {
            simpleStatement;
         }
      }
   c += 10; }
   return 0; }
```

45/5p. 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^2 (E) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg n$
(B) $n^2 \lg n$ (D) $n\sqrt{n} \lg n$ (F) n (H) $\sqrt{n}$ (J) 1

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

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

(A) `>` (B) `C-X` (C) `\0` (D) `\B` (E) `\b` (F) `\cX` (G) `\f` (H) `ff` (I) `}`

46/1p. first back-reference
47/1p. form feed
48/1p. control X
49/1p. word boundary
50/1p. end of multiplier

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

51/1p. Does the string "`ww`" match the regular expression "`w+|xs`"?

52/1p. Does the string "`qqug`" match the regular expression "`q?ug`"?

53/1p. Does the empty string match the regular expression "`wn?`"?

54/1p. Does the string "`pbnn`" match the regular expression "`pb|nn`"?

55/1p. Does the string "`ggx`" match the regular expression "`(g?x|pg)+`"?

56/1p. Does the string "`zh`" match the regular expression "`z+hy+`"?

57/1p. Does the string "`wrr`" match the regular expression "`(sg*ak|wr)?`"?

58/1p. Does the string "`wwx`" match the regular expression "`w?(xq)+`"?

59/1p. Does the string "`fbfbqgg`" match the regular expression "`(fb)?(u?)?(qg)+`"?

60/1p. Does the string "`nfpkpkk`" match the regular expression "`(nf|pk*)?`"?

61/1p. Does the string "`naaaztzbn`" match the regular expression "`na+|az|tz|(t|bn)*`"?

62/1p. Does the string "`dzhhy`" match the regular expression "`(d+|zh)+(w*yk)*`"?

Section II: Non-Bubble Answers: Write neatly. Use the paper provided. Answer on the front ONLY of each page. On the upper-right corner of the back, lightly write your 7-digit BYUH student id number. Do NOT write your NAME anywhere. (This aids fairness in grading.) Leave margins the same size as on this page. Sloppy/non-compliant work will be penalized.

Write each program on a separate sheet of paper. Write the program number in the upper left corner and circle it.

63/10p. Write this program: Prompt for and accept two integers, a and b. Print their sum (add).

64/10p. Write this program: Given a list of strings on the command line, prompt for and accept another string. Then tell whether it also appears on the command line.

65/10p. Write this program: Given n on the command line, print its prime factorization, smallest to largest. Factors divide evenly into n. Primes have no divisors except 1 and themself. For 12, print 2x2x3.

66/10p. Write this program: Subtracting time. Prompt for and accept a start time and an ending time in HH:MM:SS format. Print the difference. Example: start 1:39:44, end 3:51:15, difference 2:11:31.

67/10p. Write this program: out

Total points 150.

Answer Key (points per line)

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

Total points 150.