

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("uuuu%u3duuu",x); (D) printf("uu%-8d",x); (G) printf("%-+10d",x);
 (B) printf("uuu%-u3duuuu",x); (E) printf("u%08d",x); (H) printf("%-+9d",x);
 (C) printf("uu%+-6d",x); (F) printf("%u10d",x); (I) printf("%0+9d",x);

inputs:	<u> 2</u>	<u> -6</u>	<u>1430580870</u>	<u>-1491851610</u>
1/2p.	uuuu2uuuuuu	uuu-6uuuuuu	uuuu1430580870uuuu	uuu-1491851610uuuu
2/2p.	+00000002 <u> </u>	-00000006 <u> </u>	+1430580870 <u> </u>	-1491851610 <u> </u>
3/2p.	u00000002 <u> </u>	u-0000006 <u> </u>	u1430580870 <u> </u>	u-1491851610 <u> </u>
4/2p.	+2uuuuuuuu	-6uuuuuuuu	+1430580870	-1491851610

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

- (A) printf("uuu%1s",x); (D) printf("uu%suuu",x); (G) printf("%-5s",x);
 (B) printf("uu%2s",x); (E) printf("u%-2s",x); (H) printf("%1suuuu",x);
 (C) printf("uu%3s",x); (F) printf("u%-4s",x); (I) printf("%5s",x);

inputs:	<u>""</u>	<u>"p"</u>	<u>"hg"</u>	<u>"hlb"</u>	<u>"jqybwz"</u>	<u>"bdqbpzpx"</u>
5/2p.	uuuuu	uPuuu	uhgu	uhlbu	ujqybwzu	ubdqbpzpxuu
6/2p.	uuuuu	uuPuuu	uuhg	uuhlbu	uujqybwzu	uubdqbpzpxuu
7/2p.	uuuuu	uuuuP	uuuhg	uhlbu	jqybwz	bdqbpzpx
8/2p.	uuuuu	uuuPu	uuuhgu	uuuhlbu	uuujqybwzu	uuubdqbpzpxu

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

- (A) printf("uuu%10.0f",x); (D) printf("uu%+12.6f",x); (G) printf("u%012.0f",x);
 (B) printf("uu%+010.0f",x); (E) printf("uu%012.4f",x); (H) printf("%+14.6f",x);
 (C) printf("uu%+010.2f",x); (F) printf("u%+013.2f",x); (I) printf("%014.2f",x);

inputs:	<u> 2</u>	<u> 4.45</u>	<u>-3.3567</u>	<u>-67546.104634</u>
9/2p.	uu+000000002uu	uu+000000004uu	uu-000000003uu	uu-000067546uu
10/2p.	uu+000002.00uu	uu+000004.45uu	uu-000003.36uu	uu-067546.10uu
11/2p.	uu0000002.0000	uu0000004.4500	uu-000003.3567	uu-067546.1046
12/2p.	u+000000002.00	u+000000004.45	u-000000003.36	u-000067546.10

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

13/1p.	$8/8/8*8+2$	(A) -90	(B) 3	(C) 4	(D) 12	(E) 22	(F) 29	(G) 66	(H) 80
14/1p.	$0-7\%9/5\%9$	(A) -58	(B) -1	(C) 0	(D) 7	(E) 19	(F) 61	(G) 79	(H) 96
15/1p.	$6+7-2\%6*6$	(A) -60	(B) 1	(C) 9	(D) 17	(E) 30	(F) 36	(G) 66	(H) 92
16/1p.	$3*3+8\%5\%3$	(A) -89	(B) -28	(C) -11	(D) 0	(E) 1	(F) 2	(G) 3	(H) 9
17/1p.	$6-3*9*2\%9$	(A) -66	(B) -48	(C) -43	(D) -42	(E) -30	(F) 0	(G) 6	(H) 83
18/1p.	$6\%4!=8 8-9$	(A) -93	(B) -38	(C) -2	(D) 0	(E) 1	(F) 5	(G) 10	(H) 82
19/1p.	$2\%6-5+6*6$	(A) -54	(B) -33	(C) -18	(D) 2	(E) 12	(F) 18	(G) 33	(H) 36
20/1p.	$2\%7/6-4\%5$	(A) -87	(B) -1	(C) 0	(D) 1	(E) 2	(F) 4	(G) 28	(H) 64
21/1p.	$6-1\%9*2*2$	(A) -93	(B) -52	(C) 2	(D) 5	(E) 10	(F) 20	(G) 73	(H) 82
22/1p.	$2\%2-6/3+2$	(A) -72	(B) -51	(C) -45	(D) -41	(E) -4	(F) 0	(G) 1	(H) 2
23/1p.	$8*8-4/9/9$	(A) -86	(B) -71	(C) -59	(D) 0	(E) 7	(F) 60	(G) 64	(H) 87
24/1p.	$7-6>3>3*8$	(A) -54	(B) -1	(C) 0	(D) 6	(E) 7	(F) 8	(G) 48	(H) 56
25/1p.	$6*7>=3<3+5$	(A) -83	(B) -57	(C) -55	(D) 5	(E) 6	(F) 34	(G) 36	(H) 45
26/1p.	$6+6>=1>3+2$	(A) -83	(B) 1	(C) 2	(D) 3	(E) 6	(F) 7	(G) 8	(H) 70
27/1p.	$7\%5>0>=3-3$	(A) -61	(B) -59	(C) -52	(D) -31	(E) -2	(F) -1	(G) 0	(H) 1
28/1p.	$1==3<=8*6+9$	(A) -99	(B) -70	(C) -32	(D) 0	(E) 1	(F) 9	(G) 10	(H) 77
29/1p.	$7-9\%8/9+3$	(A) 0	(B) 1	(C) 3	(D) 7	(E) 10	(F) 19	(G) 33	(H) 37
30/1p.	$8\%3\%4-9+2$	(A) -45	(B) -9	(C) -5	(D) -2	(E) -1	(F) 0	(G) 2	(H) 75
31/1p.	$5*9+0/7\%5$	(A) -97	(B) -86	(C) -63	(D) -2	(E) 0	(F) 1	(G) 20	(H) 45

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

32/2p.	<code>int j; for(j=-8; j!=-7; j++) body;</code>
33/2p.	<code>int c; for(c=-6; c>=-13; c--) body;</code>
34/2p.	<code>int x; for(x=0; x<1; ++x) body;</code>
35/2p.	<code>int m=8; while(m-- >= 4) body;</code>
36/2p.	<code>int b=7; while(b-- != 4) body;</code>
37/2p.	<code>int z=1; do body; while(--z > -1);</code>
38/2p.	<code>int d=-5; while(++d < 2) body;</code>
39/2p.	<code>int e; for(e=8; e>5; e--) body;</code>

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^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 ) {
        for ( h = 1 ; h * h < n ; h += 3 ) {
            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) $n \lg n$ (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 ) {
        i = n; while ( i > 1 ) {
            c = 1; while ( c * c < n ) {
                simpleStatement;
                c += 5; }
            i--; }
    } 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) $n \lg n$ (G) $\sqrt{n} \lg n$ (I) $\lg n$
 (B) $n^2 \lg n$ (D) $n\sqrt{n}$ (F) n (H) $\lg^2 n$ (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    for ( h = 1 ; h * h < n ; h++ ) {
        if ( simpleCompare ) {
            for ( k = 1 ; k * k < n ; k += 10 ) {
                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^8 (C) $n^4\sqrt{n}$ (E) $n^2\sqrt{n}\lg^5 n$ (G) $n^2\sqrt{n}\lg^3 n$ (I) $\sqrt{n}\lg n$
 (B) $n^7\lg n$ (D) $n^4\lg^2 n$ (F) $n^2\sqrt{n}\lg^4 n$ (H) $n\sqrt{n}\lg n$ (J) $\lg^2 n$

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  b = 1; do {
    e = n; do {
      for ( a = 1 ; a < n ; a *= 3 ) {
        i = 1; while ( i < n ) {
          for ( k = n ; k > 1 ; k -= 5 ) {
            g = n; do {
              h = 1; do {
                for ( f = 1 ; f < n ; f += 1 ) {
                  simpleStatement;
                }
                h += 3; } while ( h < n );
            } while ( g > 1 );
          }
          i += 2; }
        }
      e--; } while ( e > 1 );
    } while ( b < n );
  } while ( b < n );
  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) $n^4\sqrt{n}$ (C) n^4 (E) $n^2\sqrt{n}\lg n$ (G) $n\sqrt{n}\lg^3 n$ (I) $\lg^2 n$
 (B) $n^4\lg^2 n$ (D) $n^3\lg^2 n$ (F) $n^2\sqrt{n}$ (H) $n\lg^4 n$ (J) 1

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  c = n; while ( c > 1 ) {
    f = 1; while ( f * f < n ) {
      k = n; while ( k > 1 ) {
        for ( i = 1 ; i < n ; i += 2 ) {
          j = 1; do {
            if ( simpleCompare ) {
              h = 1; do {
                simpleStatement;
              } while ( h * h < n );
            }
          } while ( j < n );
          j *= 3; } while ( j < n );
        }
        k -= 2; }
      f += 10; }
    c /= 5; }
  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^3 (C) $n^2 \lg n$ (E) $n\sqrt{n} \lg n$ (G) $n \lg n$ (I) \sqrt{n}
 (B) $n^2\sqrt{n}$ (D) n^2 (F) $n\sqrt{n}$ (H) $\sqrt{n} \lg^2 n$ (J) $\lg^2 n$

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  if ( simpleCompare ) {
    if ( simpleCompare ) {
      if ( simpleCompare ) {
        if ( simpleCompare ) {
          if ( simpleCompare ) {
            simpleStatement;
          } else {
            simpleStatement;
          }
        } else {
          simpleStatement;
        }
      } else {
        f = 1; while ( f < n ) {
          simpleStatement;
          f += 1; }
      }
    } else {
      for ( d = n ; d > 1 ; d -= 2 ) {
        for ( g = 1 ; g < n ; g *= 2 ) {
          simpleStatement;
        }
      }
    }
  } else {
    c = 1; do {
      h = n; do {
        b = n; do {
          if ( simpleCompare ) {
            if ( simpleCompare ) {
              simpleStatement;
            } else {
              simpleStatement;
            }
          }
          b /= 5; } while ( b > 1 );
        h -= 1; } while ( h > 1 );
      c++; } while ( c < 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 “`aba(b*bba)+`”. List the shortest five (or all) strings in L.

47/4p. Let L be the language accepted by the regular expression “`aba(abb|ba)?`”. List the shortest five (or all) strings in L.

48/4p. Let L be the language accepted by the regular expression “`(abbbba)*abb`”. List the shortest five (or all) strings in L.

49/4p. Let L be the language accepted by the regular expression “`((abaab)*b*|abbba+b*)?aababaa+`”. 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 their product (multiply).

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 its letters in reverse order.

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). | B |
| 2 (2). | I |
| 3 (2). | E |
| 4 (2). | G |
| 5 (2). | E |
| 6 (2). | D |
| 7 (2). | I |
| 8 (2). | A |
| 9 (2). | B |
| 10 (2). | C |
| 11 (2). | E |
| 12 (2). | F |
| 13 (1). | J (2) |
| 14 (1). | B (-1) |
| 15 (1). | B (1) |
| 16 (1). | H (9) |
| 17 (1). | G (6) |
| 18 (1). | E (1) |
| 19 (1). | G (33) |
| 20 (1). | J (-4) |
| 21 (1). | C (2) |
| 22 (1). | F (0) |
| 23 (1). | G (64) |
| 24 (1). | C (0) |
| 25 (1). | J (1) |
| 26 (1). | J (0) |
| 27 (1). | H (1) |
| 28 (1). | E (1) |
| 29 (1). | E (10) |
| 30 (1). | C (-5) |
| 31 (1). | H (45) |
| 32 (2). | 1 |
| 33 (2). | 8 |
| 34 (2). | 1 |
| 35 (2). | 5 |
| 36 (2). | 3 |
| 37 (2). | 2 |
| 38 (2). | 6 |
| 39 (2). | 3 |
| 40 (3). | H (\sqrt{n}) |
| 41 (3). | D ($n\sqrt{n}$) |
| 42 (3). | F (n) |
| 43 (5). | B ($n^7 \lg n$) |
| 44 (5). | D ($n^3 \lg^2 n$) |
| 45 (10). | C ($n^2 \lg n$) |

Total points 150.