

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("uuuuu%02d",x); (D) printf("uuu%+6d",x); (G) printf("u%+8d",x);
 (B) printf("uuuu%5d",x); (E) printf("uu%0u5d",x); (H) printf("%-10d",x);
 (C) printf("uuu%+05d",x); (F) printf("uu%0+7d",x); (I) printf("%4d",x);

inputs:	<u>5</u>	<u>-4</u>	<u>1362528547</u>	<u>-1951769934</u>
1/2p.	uuu0005uuu	uu-0004uuu	uuu1362528547uuu	uu-1951769934uuu
2/2p.	uuuuu05uuu	uuuuu-4uuu	uuuuu1362528547uuu	uuuuu-1951769934uuu
3/2p.	uuu5uuuuuu	uu-4uuuuuu	1362528547uuuuuu	-1951769934uuuuuu
4/2p.	5uuuuuuuuu	-4uuuuuuuu	1362528547	-1951769934

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

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

inputs:	<u>""</u>	<u>"j"</u>	<u>"yp"</u>	<u>"flwc"</u>	<u>"bgvqch"</u>	<u>"qyccjdkl"</u>
5/2p.	uuuuuu	uujuuuu	uuypuuuu	uflwcuuuu	uubgvqchuuuu	uuqyccjdkluuuu
6/2p.	uuuuuu	uujuuu	uuypu	uflwc	uubgvqch	uuuqyccjdkl
7/2p.	uuuuuu	uujuuu	uypuuu	uflwcuuu	ubgvqchuuu	uqyccjdkluuu
8/2p.	uuuuuu	uuuuju	uuypu	uflwcu	bgvqchu	qyccjdklu

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

- (A) printf("uu%+010f",x); (D) printf("u%012.6f",x); (G) printf("%0+12f",x);
 (B) printf("u%+11.0f",x); (E) printf("u%013.4f",x); (H) printf("%13f",x);
 (C) printf("u%+12.0f",x); (F) printf("%+13f",x); (I) printf("%14.4f",x);

inputs:	<u>8</u>	<u>-7.40</u>	<u>3.3827</u>	<u>-16304.985766</u>
9/2p.	uu+08.000000uu	uu-07.400000uu	uu+03.382700uu	uu-16304.985766uu
10/2p.	uuuu+8.000000u	uuuu-7.400000u	uuuu+3.382700u	-16304.985766u
11/2p.	uuuuuuuuuu+8uu	uuuuuuuuuu-7uu	uuuuuuuuuu+3uu	uuuuuu-16305uu
12/2p.	uuuuuuuuuu+8u	uuuuuuuuuu-7u	uuuuuuuuuu+3u	uuuuuu-16305u

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

- 13/1p. $6-6||6<=6+9$ (A) -34 (B) -15 (C) -4 (D) 1 (E) 5 (F) 10 (G) 14 (H) 21
 14/1p. $6+7\%1/7-5$ (A) -95 (B) -64 (C) -5 (D) -2 (E) 0 (F) 6 (G) 57 (H) 92
 15/1p. $9/3<5||1*5$ (A) -44 (B) 0 (C) 1 (D) 5 (E) 9 (F) 45 (G) 65 (H) 94
 16/1p. $0-3-6+5-6$ (A) -90 (B) -43 (C) -20 (D) -10 (E) -8 (F) 2 (G) 4 (H) 14
 17/1p. $2/3/9-3*4$ (A) -12 (B) -4 (C) -2 (D) 0 (E) 28 (F) 40 (G) 65 (H) 89
 18/1p. $7/4-2*5+7$ (A) -16 (B) -12 (C) -1 (D) 0 (E) 2 (F) 5 (G) 22 (H) 36
 19/1p. $2+6/6+4-7$ (A) -98 (B) -39 (C) -20 (D) -7 (E) -5 (F) -2 (G) 0 (H) 4
 20/1p. $5-8-3/2*5$ (A) -30 (B) -20 (C) -15 (D) -10 (E) -5 (F) -3 (G) -1 (H) 2
 21/1p. $6/1||7>4+9$ (A) -1 (B) 0 (C) 6 (D) 9 (E) 10 (F) 15 (G) 33 (H) 54
 22/1p. $8-9\%0-6/8$ (A) -14 (B) -1 (C) 0 (D) 3 (E) 8 (F) 9 (G) 24 (H) 81
 23/1p. $9-2-5>3>=5$ (A) -99 (B) 0 (C) 1 (D) 6 (E) 8 (F) 9 (G) 34 (H) 48
 24/1p. $4+3/4/3-9$ (A) -25 (B) -9 (C) -8 (D) -5 (E) -2 (F) 1 (G) 3 (H) 4
 25/1p. $6/4-2*3+1$ (A) -7 (B) -6 (C) -2 (D) 0 (E) 2 (F) 10 (G) 40 (H) 55
 26/1p. $9/7*4*6-9$ (A) -39 (B) -36 (C) -12 (D) -9 (E) -1 (F) 0 (G) 15 (H) 94
 27/1p. $3*9>=8||9+4$ (A) -75 (B) -64 (C) 3 (D) 5 (E) 7 (F) 15 (G) 49 (H) 78
 28/1p. $2*7/2*5/8$ (A) -36 (B) -18 (C) 0 (D) 2 (E) 3 (F) 4 (G) 14 (H) 72
 29/1p. $9-4/7\%1*2$ (A) -54 (B) 1 (C) 5 (D) 18 (E) 36 (F) 60 (G) 81 (H) 98
 30/1p. $8*3\%2/2+6$ (A) -69 (B) 0 (C) 6 (D) 10 (E) 11 (F) 24 (G) 31 (H) 48
 31/1p. $9-1+0\&\&3<=2$ (A) -49 (B) 0 (C) 1 (D) 7 (E) 8 (F) 9 (G) 22 (H) 24

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

- 32/2p. `int m=7; do body; while(--m > 2);`
 33/2p. `int x=-10; do body; while(x++ < -6);`
 34/2p. `int w=-10; while(--w >= -12) body;`
 35/2p. `int e=1; while(e-- != -2) body;`
 36/2p. `int k=0; while(k++ != 6) body;`
 37/2p. `int k=0; do body; while(++k < 1);`
 38/2p. `int r=10; while(++r <= 17) body;`
 39/2p. `int e=-8; while(e++ < -6) 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 ( b = 1 ; b < n ; b *= 5 ) {
            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^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 ) {
        for ( a = 1 ; a < n ; a++ ) {
            i = 1; while ( i * i < n ) {
                simpleStatement;
                i += 1; }
        }
    } 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^3 (C) $n^2 \lg n$ (E) $n \lg n$ (G) $\sqrt{n} \lg^2 n$ (I) $\lg^2 n$
(B) $n^2\sqrt{n}$ (D) n^2 (F) n (H) $\sqrt{n} \lg n$ (J) $\lg n$

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    for ( c = 1 ; c < n ; c *= 3 ) {
        if ( simpleCompare ) {
            e = 1; while ( e < n ) {
                h = 1; while ( h * h < n ) {
                    simpleStatement;
                    h++; }
                e *= 2; }
        }
    }
    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) $n\sqrt{n} \lg n$ (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 ) {
    c = n; do {
      if ( simpleCompare ) {
        simpleStatement;
      } else {
        simpleStatement;
      }
      c /= 3; } while ( c > 1 );
  } 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) $\lg^3 n$
 (B) $n^2 \lg n$ (D) $n \lg^2 n$ (F) $\sqrt{n} \lg^2 n$ (H) \sqrt{n} (J) $\lg n$

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  j = n; do {
    if ( simpleCompare ) {
      if ( simpleCompare ) {
        i = 1; while ( i < n ) {
          b = n; do {
            simpleStatement;
          } while ( b > 1 );
          i *= 2; }
      } else {
        simpleStatement;
      }
    } else {
      if ( simpleCompare ) {
        if ( simpleCompare ) {
          simpleStatement;
        } else {
          simpleStatement;
        }
      }
    }
    j -= 10; } while ( j > 1 );
  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^4 (C) $n^2\sqrt{n}$ (E) $n\sqrt{n}\lg^2 n$ (G) $n\sqrt{n}$ (I) $\lg n$
 (B) $n^2\sqrt{n}\lg n$ (D) $n^2\lg n$ (F) $n\sqrt{n}\lg n$ (H) $\lg^4 n$ (J) 1

```
int main ( int argc, char * * argv ) {
  int n = atoi(argv[1]);
  if ( simpleCompare ) {
    if ( simpleCompare ) {
      if ( simpleCompare ) {
        if ( simpleCompare ) {
          d = 1; do {
            simpleStatement;
            d += 3; } while ( d * d < n );
        } else {
          simpleStatement;
        }
      } else {
        simpleStatement;
      }
    } else {
      if ( simpleCompare ) {
        if ( simpleCompare ) {
          simpleStatement;
        } else {
          simpleStatement;
        }
      } else {
        simpleStatement;
      }
    }
  } else {
    for ( j = 1 ; j * j < n ; j += 5 ) {
      for ( k = 1 ; k < n ; k *= 2 ) {
        i = n; while ( i > 1 ) {
          if ( simpleCompare ) {
            for ( b = n ; b > 1 ; b-- ) {
              simpleStatement;
            }
          } else {
            simpleStatement;
          }
        }
        i--; }
      }
    }
  }
  return 0; }
```

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

(A) \$ (B) \1 (C) \A (D) \W (E) \a (F) \n (G) \r (H) \w (I) ^

46/1p. carriage return

47/1p. first back-reference

48/1p. end of string

49/1p. word character

50/1p. alarm (alert)

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

51/1p. Does the string "rfh" match the regular expression "r*|fh"?

52/1p. Does the string "ryryp" match the regular expression "ry?p"?

53/1p. Does the string "xwxwww" match the regular expression "(xw+|w)?"?

54/1p. Does the string "syn" match the regular expression "s+(yn)?"?

55/1p. Does the string "gup" match the regular expression "g?|ak|up"?

56/1p. Does the string "gxf" match the regular expression "(tp|g|xf)*"?

57/1p. Does the string "cu" match the regular expression "f+|cu((fk)+)?"?

58/1p. Does the string "w" match the regular expression "w+|yw|k"?

59/1p. Does the string "drsatt" match the regular expression "d+rs|(at+)?"?

60/1p. Does the string "tqqr" match the regular expression "(ph)?tq+r?"?

61/1p. Does the string "ax" match the regular expression "r?(gq)?(dk)*|ax"?

62/1p. Does the string "tuk" match the regular expression "t+uq?kt?"?

Total points 100.

Answer Key (points per line)

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

Total points 100.