

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(" %+d ",x); | (D) printf(" %+6d ",x); | (G) printf(" %07d ",x); |
| (B) printf(" %+05d ",x); | (E) printf(" %05d ",x); | (H) printf(" %+9d ",x); |
| (C) printf(" %+-+6d ",x); | (F) printf(" %+-+7d ",x); | (I) printf(" %--7d ",x); |

| inputs: | <u>2</u> | <u>-9</u> | <u>1913519252</u> | <u>-1499719971</u> |
|---------|------------------------|-------------------------|----------------------------|----------------------------|
| 1/2p. | <u>+2 </u> | <u>-9 </u> | <u>+1913519252 </u> | <u>-1499719971 </u> |
| 2/2p. | <u>2 </u> | <u>-9 </u> | <u>1913519252 </u> | <u>-1499719971 </u> |
| 3/2p. | <u> +2 </u> | <u> -9 </u> | <u> +1913519252 </u> | <u> -1499719971 </u> |
| 4/2p. | <u> +2 </u> | <u> -9 </u> | <u> +1913519252 </u> | <u> -1499719971 </u> |

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

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

| inputs: | <u>" "</u> | <u>"f"</u> | <u>"wz"</u> | <u>"kjpv"</u> | <u>"zpgdxl"</u> | <u>"lcyqlcgb"</u> |
|---------|---------------|-----------------|------------------|--------------------|----------------------|------------------------|
| 5/2p. | <u> </u> | <u>f </u> | <u>wz </u> | <u>kjpv </u> | <u>zpgdxl </u> | <u>lcyqlcgb </u> |
| 6/2p. | <u> </u> | <u>f </u> | <u>wz </u> | <u>kjpv </u> | <u>zpgdxl </u> | <u>lcyqlcgb </u> |
| 7/2p. | <u> </u> | <u> f </u> | <u> wz </u> | <u> kjpv </u> | <u> zpgdxl </u> | <u> lcyqlcgb </u> |
| 8/2p. | <u> </u> | <u> f </u> | <u> wz </u> | <u> kjpv </u> | <u> zpgdxl </u> | <u> lcyqlcgb </u> |

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

- | | | |
|-----------------------------------|------------------------------|-------------------------------|
| (A) printf(" %+010.2f ",x); | (D) printf(" %+13.4f ",x); | (G) printf(" %+012.6f ",x); |
| (B) printf(" %+010.6f ",x); | (E) printf(" %011.2f ",x); | (H) printf(" %+014.2f ",x); |
| (C) printf(" %+12.4f ",x); | (F) printf(" %011f ",x); | (I) printf(" %+14.2f ",x); |

| inputs: | <u>7</u> | <u>-1.06</u> | <u>4.7799</u> | <u>-44873.577574</u> |
|---------|------------------------|------------------------|------------------------|----------------------------|
| 9/2p. | <u>00000007.00 </u> | <u>-0000001.06 </u> | <u>00000004.78 </u> | <u>-0044873.58 </u> |
| 10/2p. | <u>+07.000000 </u> | <u>-01.060000 </u> | <u>+04.779900 </u> | <u>-44873.577574 </u> |
| 11/2p. | <u> +7.0000 </u> | <u> -1.0600 </u> | <u> +4.7799 </u> | <u> -44873.5776 </u> |
| 12/2p. | <u> +7.00 </u> | <u> -1.06 </u> | <u> +4.78 </u> | <u> -44873.58 </u> |

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

| | | | | | | | | | |
|--------|-------------|---------|---------|---------|---------|---------|---------|--------|--------|
| 13/1p. | 5/2-7-6/5 | (A) -78 | (B) -6 | (C) -3 | (D) -1 | (E) 0 | (F) 1 | (G) 2 | (H) 14 |
| 14/1p. | 4%5>=3<=0-6 | (A) -58 | (B) -6 | (C) -5 | (D) -2 | (E) -1 | (F) 1 | (G) 24 | (H) 27 |
| 15/1p. | 4*4-2*4%7 | (A) -16 | (B) 1 | (C) 4 | (D) 5 | (E) 8 | (F) 12 | (G) 15 | (H) 56 |
| 16/1p. | 4>5>6*8-5 | (A) -49 | (B) -5 | (C) -4 | (D) 0 | (E) 1 | (F) 9 | (G) 32 | (H) 73 |
| 17/1p. | 9*0-5-1-4 | (A) -90 | (B) -58 | (C) -50 | (D) -42 | (E) -18 | (F) -10 | (G) -8 | (H) 0 |
| 18/1p. | 5+9+5%9+8 | (A) -90 | (B) -22 | (C) 2 | (D) 9 | (E) 18 | (F) 19 | (G) 27 | (H) 40 |
| 19/1p. | 0-8-2%5+1 | (A) -11 | (B) -10 | (C) -9 | (D) -5 | (E) 0 | (F) 1 | (G) 2 | (H) 5 |
| 20/1p. | 9*8/7-5+6 | (A) -90 | (B) -30 | (C) -18 | (D) -2 | (E) -1 | (F) 10 | (G) 11 | (H) 42 |
| 21/1p. | 0-9>=9>2-9 | (A) -96 | (B) -88 | (C) -64 | (D) -10 | (E) 0 | (F) 1 | (G) 8 | (H) 9 |
| 22/1p. | 3/2*7-6+8 | (A) -14 | (B) -7 | (C) -1 | (D) 0 | (E) 2 | (F) 8 | (G) 9 | (H) 23 |
| 23/1p. | 0/9-7%4*9 | (A) -94 | (B) -27 | (C) -7 | (D) 21 | (E) 29 | (F) 35 | (G) 71 | (H) 75 |
| 24/1p. | 5+7*3/5/5 | (A) -88 | (B) -56 | (C) 0 | (D) 5 | (E) 7 | (F) 26 | (G) 36 | (H) 60 |
| 25/1p. | 2+0<=3<9*3 | (A) -87 | (B) -72 | (C) -4 | (D) 0 | (E) 1 | (F) 3 | (G) 9 | (H) 32 |
| 26/1p. | 6/7+4%6+6 | (A) -97 | (B) 0 | (C) 4 | (D) 6 | (E) 7 | (F) 20 | (G) 40 | (H) 62 |
| 27/1p. | 4/4/3-8+7 | (A) -42 | (B) -15 | (C) -11 | (D) -4 | (E) 0 | (F) 2 | (G) 3 | (H) 6 |
| 28/1p. | 7+2+8==4 5 | (A) -93 | (B) 0 | (C) 7 | (D) 8 | (E) 10 | (F) 38 | (G) 86 | (H) 94 |
| 29/1p. | 4-7/5/6+9 | (A) -11 | (B) -5 | (C) -1 | (D) 0 | (E) 4 | (F) 8 | (G) 9 | (H) 37 |
| 30/1p. | 4+4!=6<=4+7 | (A) -53 | (B) -40 | (C) -29 | (D) -10 | (E) 5 | (F) 8 | (G) 12 | (H) 70 |
| 31/1p. | 3%4-6%3*4 | (A) -36 | (B) -15 | (C) -3 | (D) -1 | (E) 0 | (F) 8 | (G) 9 | (H) 12 |

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

- 32/2p. int z=-3; do body; while(z++ != 0);
- 33/2p. int f; for(f=-2; f>-9; f++) body;
- 34/2p. int i; for(i=10; i>=7; i--) body;
- 35/2p. int i; for(i=-10; i!=-13; ++i) body;
- 36/2p. int x=3; while(++x != 6) body;
- 37/2p. int g=-4; do body; while(++g <= 1);
- 38/2p. int x; for(x=-6; x!=-1; ++x) body;
- 39/2p. int u; for(u=0; u!=-6; u--) 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 (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 ( k = n ; k > 1 ; k -= 1 ) {
            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) $n\sqrt{n}\lg n$ (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 ) {
        a = 1; while ( a * a < n ) {
            for ( c = n ; c > 1 ; c /= 5 ) {
                simpleStatement;
            }
            a += 10;
        } 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^4 (C) $n^3\lg n$ (E) $n^2\sqrt{n}$ (G) $n\sqrt{n}\lg^2 n$ (I) $\sqrt{n}\lg n$
 (B) $n^3\sqrt{n}$ (D) n^3 (F) $n^2\lg n$ (H) $n\lg n$ (J) $\lg^4 n$

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

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    i = 1; while ( i * i < n ) {
        for ( d = n ; d > 1 ; d /= 2 ) {
            for ( a = n ; a > 1 ; a /= 3 ) {
                j = 1; do {
                    e = 1; while ( e * e < n ) {
                        if ( simpleCompare ) {
                            simpleStatement;
                        } else {
                            simpleStatement;
                        }
                    }
                    e++; }
                j += 10; } while ( j * j < n );
            }
        }
    i += 1; }
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) $n^2\lg^3 n$ (E) $n\sqrt{n}\lg^2 n$ (G) $n\lg^4 n$ (I) $\lg^5 n$
 (B) $n^2\sqrt{n}\lg n$ (D) $n^2\lg n$ (F) $n\sqrt{n}\lg n$ (H) n (J) 1

```
int main ( int argc, char * * argv ) {
    int n = atoi(argv[1]);
    i = 1; do {
        j = 1; do {
            if ( simpleCompare ) {
                g = 1; do {
                    for ( f = 1 ; f < n ; f *= 2 ) {
                        h = 1; do {
                            simpleStatement;
                        }
                        h++; } while ( h * h < n );
                    }
                g++; } while ( g * g < n );
            } else {
                if ( simpleCompare ) {
                    simpleStatement;
                }
            }
        }
    j += 10; } while ( j * j < n );
    i *= 2; } while ( i < n );
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) $n^3\sqrt{n}$ (C) n^2 (E) $n\sqrt{n}\lg n$ (G) $\sqrt{n}\lg^3 n$ (I) $\lg^4 n$
- (B) $n^2\sqrt{n}$ (D) $n\sqrt{n}\lg^2 n$ (F) $n\lg^3 n$ (H) $\sqrt{n}\lg n$ (J) $\lg^2 n$

```

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

```

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

- (A) \1 (B) \2 (C) \A (D) \B (E) \a (F) \s (G)] (H) {n+} (I) {n,}

- 46/1p. n or more times
47/1p. end of character class
48/1p. word boundary
49/1p. alarm (alert)
50/1p. second back-reference

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

- 51/1p. Does the string "ssz" match the regular expression "s+z"?
52/1p. Does the string "wtt" match the regular expression "w+tt"?
53/1p. Does the string "dgg" match the regular expression "dg"??
54/1p. Does the string "z" match the regular expression "z+u"??
55/1p. Does the string "fts" match the regular expression "q*|ft|s"?
56/1p. Does the string "paguaf" match the regular expression "(pa)*gu(af)?"?
57/1p. Does the string "hhuyy" match the regular expression "((hd)?h?uy)?"?
58/1p. Does the string "wrwrwrwr" match the regular expression "(r+wr+|br)+"?
59/1p. Does the string "unkh" match the regular expression "ab|un(sr)*kh"?
60/1p. Does the string "k" match the regular expression "(fp|k)*(hh)*|hy"?
61/1p. Does the string "pth" match the regular expression "(pc)+|(gy)?|(bp)+th"?
62/1p. Does the string "zwwayayzwwayaykg" match the regular expression "(((zw)+|(ay)+)?(kg)+)?"??

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: Accept a list of one or more integers from the command line (argv). Print the largest (max).

65/10p. Write this program: Accept two integers, a and b, from the command line (argv). Print an a-by-b hollow rectangular box of stars (*).

66/10p. Write this program: Write a subroutine that, given a word, returns 'yes' if its vowels are in alphabetical order, otherwise returns 'no'. Vowels are a, e, i, o, and u.

67/10p. Write this program: 3.0

Total points 150.

Answer Key (points per line)

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

Total points 150.