This test has 10 questions worth a total of 50 points. You have 120 minutes. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. Write out and sign the Honor Code pledge before turning in the test.

“I pledge my honor that I have not violated the Honor Code during this examination.”

Signature

Name:

NetID:

<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
<th>Problem</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Sub 1</td>
<td></td>
<td>Sub 2</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
0. Miscellaneous. (2 points)
(a) Write your name and Princeton NetID in the space provided on the front of the exam, and circle your precept number.
(b) Write and sign the honor code on the front of the exam.

1. Number systems. (4 points)
(a) What is the decimal representation of the binary integer $111001_2$? Circle your answer.
(b) Repeat (a), but assume $111001_2$ is a 6-bit two's complement binary integer.
(c) Convert the decimal integer $333_{10}$ to hexadecimal. Circle your answer.
(d) What is the largest positive value a Java int (32-bit two's complement signed integer) can store? Circle the correct answer.

\[ 2^{31} - 1 \quad 2^{31} \quad 2^{32} - 1 \quad 2^{32} \quad 2^{32} + 1 \]
2. Strings, booleans, and conditionals. (4 points)

Consider the following program.

```java
public class Conditionals {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        String s = "a";
        String t = "b";
        if (N >= 10 && N <= 20) s = s + t;
        else if (N <= 5 || N >= 25) s = t + s;
        else if (N >= 30) s = s + s;
        else s = t + t;

        if (N % 3 == 0) {
            t = s + t;
            if (N % 5 != 0) {
                t = t + s;
            }
        }

        System.out.println(s + t);
    }
}
```

(a) What gets printed when you execute the above program with

```bash
% java Conditionals 15
```

(b) Which (one or more) of the following correspond to the output of `Conditionals` for some positive input value of `N`? Circle all correct answers.

- `aaaab`
- `ababbab`
- `abb`
- `bbb`
- `bbbbbb`
3. Nested loops and conditionals. (6 points)

Consider the following program.

```java
public class Pattern {
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);

        for (int i = -N; i <= N; i++) {
            for (int j = -N; j <= N; j++) {
                if (i*i <= j*j) System.out.print("A ");
                else System.out.print("B ");
            }
            System.out.println();
        }
    }
}
```

(a) What is the result of executing the following command?

```bash
% java Pattern 2
```
(b) Suppose that you execute the following command

```
% java Pattern 1000 > output1000.txt
```

and observe that the resulting file `output1000.txt` is 8,010,003 bytes. Now suppose that you execute the following command.

```
% java Pattern 3000 > output3000.txt
```

Estimate (within 1%) the size of the file `output3000.txt` in bytes. Circle your answer.
4. Debugging and arrays. (6 points)

Consider the following (buggy) function mystery(). The line numbers are for reference.

```java
public class Mystery {
    public static void mystery(double[] a) {
        for (int i = 0; i < a.length; i++) {
            int max = i;
            for (int j = i+1; j <= a.length; j++) {
                if (a[j] > a[max])
                    max = i;
            }
            // swap a[i] with largest element in the rest of the array
            temp = a[i];
            a[i] = a[max];
            a[max] = temp;
        }
    }
}
```

(a) Describe the intended purpose of mystery(a) in the box below. Base your answer on the comments, not the code.
(b) When compiling Mystery, you receive the following error message. Indicate the line number you would modify to fix this error and write the corrected version below.

```
javac Mystery.java
Mystery.java:13: cannot find symbol
  symbol   : variable temp
  location: class Mystery
    temp = a[i];
```  

(c) After fixing the error in (b), your program compiles cleanly, but when calling `mystery(a)`, you receive the following error message. Indicate the line number you would modify to fix this error and write the corrected version below.

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException
  at Mystery.mystery(Mystery.java:8)
```  

(d) After fixing the errors in (b) and (c), calling `mystery(a)` does not produce any errors, but it does not work according to the intent specified in part (a) because of one tiny error. (Hint: look carefully at lines 6–10). Describe what the function actually does to the array `a[]` in the box below.

```
```

(e) Indicate the line number you would modify to fix the logical error described in (d) and write the corrected version below.
5. **Java basics and standard input. (7 points)**

Write a complete Java program `Streak.java` that reads in a sequence of positive integers from standard input and prints out the length of the longest streak (consecutive values in the sequence are the same). Assume that you have access to the library `StdIn.java`. Your answer will be graded for correctness and clarity.

```java
public class Streak {
    public static void main (String[] args) {
        int longest = 0; // length of longest streak so far
        int inarow = 0; // length of current streak

        System.out.println(longest);
    }
}
```

Here is a sample execution.

```
% more input.txt
1 1 2 2 1 3 3 3 3 3 1 1 1 4 4 4

% java Streak < input.txt
5
```
6. **Functions and binary numbers. (5 points)**

Write a function `bitCount()` that takes a positive integer argument `x` and returns the number of “1” bits in its binary representation. This quantity is useful in cryptography and information theory. For example, the bit count of 67\(_{10} = 1000011\)\(_{2}\) is 3. *Hint*: to extract the bits, use either integer division (\(\%\) and \(/\)) or bit-whacking (\& and >>).

```java
public static int bitCount(int x) {
    // Implementation goes here
}
```
7. Functions and recursion. (5 points)

Consider the following two Java functions.

```java
public static void stars(int N) {
    for (int i = 0; i < N; i++)
        System.out.print("*");
    System.out.println();
}

public static void mystery(int N) {
    if (N == 0) return;
    stars(N);
    mystery(N-1);
    stars(N);
}
```

(a) What gets printed when you call `mystery(3)`?
(b) What gets printed when you call mystery(4) ?

(c) What happens when you call mystery(-1) ? Circle the best answer.
   
   i. Prints one empty line.
   ii. Prints one asterisk.
   iii. Goes into an infinite loop, printing nothing.
   iv. Goes into an infinite loop, printing asterisks.
   v. Prints some blank lines, then produces a StackOverflowError.
8. Recursion, analysis of algorithms, and number representation. (6 points)

The binomial coefficient $C(N, k)$ is the number of ways of choosing $k$ unordered outcomes from $N$ possibilities. A recursive formula is given by

$$
C(N, k) = \begin{cases} 
1 & \text{if } k = 0 \\
0 & \text{if } N = 0 \text{ and } k \geq 1 \\
C(N-1, k) + C(N-1, k-1) & \text{otherwise}
\end{cases}
$$

The function $C()$ is a natural Java implementation of the recursive definition.

```java
public static int C(int N, int k) {
    if (k == 0) return 1;
    if (N == 0) return 0;
    return C(N-1, k) + C(N-1, k-1);
}
```

(a) Complete the tree of recursive calls for $C(4, 2)$.

```
C(4, 2)
   /  \
  /    \ 
C(3, 2)  C(3, 1)
```
(b) What value does $C(4, 2)$ return?

(c) On Kevin’s office computer, it takes about 15 minutes to compute $C(36, 18)$. Moreover, the value returned is 485200708 (as opposed to the correct answer of 9,075,135,300). Briefly explain each defect in a box below.

- **Excessively slow execution time:**

- **Wrong answer:**

(d) Estimate how long it would take to evaluate $C(37, 18)$ on Kevin’s office computer. Circle the best answer.

- Instantaneous
- 15 minutes
- 30 minutes
- 1 hour
- Infinite loop
9. **TOY. (5 points)**

An integer array is stored starting at memory location 51. The length of the array is stored in memory location 50. The following (partial) TOY program reverses the array by swapping the first element and the last element, then swapping the second element and the next-to-last element, and so on until the lo and hi pointers cross.

```
15:     ____  15-16 breaks out loop if hi < lo
16:    D41E  if(R[4] > 0) pc <- 1E
17:     ____  R[5] <- mem[R[2]]  17-1A perform the swap
1A:    B602  mem[R[2]] <- R[6]
1D:     ____  jump to addr 15
1E:    0000  halt
```

Fill in the 5 missing instructions in the space provided above. You **must** select instructions from the following list.

```
2423  2432  7251  7350  8250  8251  8350  A502  B502  D015  F015
```
TOY REFERENCE CARD

INSTRUCTION FORMATS

| . . . . | . . . . | . . . . | . . . . |
Format 1: | opcode | d | s | t | (0-6, A-B)
Format 2: | opcode | d | addr | (7-9, C-F)

ARITHMETIC and LOGICAL operations
1: add \( R[d] \leftarrow R[s] + R[t] \)
2: subtract \( R[d] \leftarrow R[s] - R[t] \)
3: and \( R[d] \leftarrow R[s] \& R[t] \)
4: xor \( R[d] \leftarrow R[s] \oplus R[t] \)
5: shift left \( R[d] \leftarrow R[s] \ll R[t] \)
6: shift right \( R[d] \leftarrow R[s] \gg R[t] \)

TRANSFER between registers and memory
7: load address \( R[d] \leftarrow addr \)
8: load \( R[d] \leftarrow \text{mem}[addr] \)
9: store \( \text{mem}[addr] \leftarrow R[d] \)
A: load indirect \( R[d] \leftarrow \text{mem}[R[t]] \)
B: store indirect \( \text{mem}[R[t]] \leftarrow R[d] \)

CONTROL
0: halt halt
C: branch zero if \( (R[d] == 0) \) \( \text{pc} \leftarrow \text{addr} \)
D: branch positive if \( (R[d] > 0) \) \( \text{pc} \leftarrow \text{addr} \)
E: jump register \( \text{pc} \leftarrow R[d] \)
F: jump and link \( R[d] \leftarrow \text{pc}; \text{pc} \leftarrow \text{addr} \)

Register 0 always reads 0.
Loads from \( \text{mem}[FF] \) come from stdin.
Stores to \( \text{mem}[FF] \) go to stdout.