

Final CS100J 09:00–11:30 Barton Hall East 8 May 2003

The results of this Final will be posted on CMS as soon as it is graded, hopefully this evening. Grades for the course may be available Friday, but it may be Monday.

Please submit any regrade requests using CMS where possible (or by emailing dgries@charter.net where it is not possible) BEFORE 6PM TONIGHT.

THIS FINAL EXAMINATION

You have 2.5 hours to complete the questions in this exam. The questions are numbered 0..8. Please glance through the whole exam before starting. The exam is worth a total of 100 points.

Question 0 (1 point). Print your name and net id at the top of each page. Please make them legible.

Question 1 (10 points). Algorithms. Write the logarithmic algorithm to store a^b (a multiplied by itself b times) in **int** variable z. Assume all variables are **int** variables and that $b \geq 0$.

DON'T write a method. Just write the statements that perform the task. You are expected to write the loop invariant first and then to develop the loop from the invariant.

Question 2 (10 points). Strings and methods. A palindrome is a string that reads the same backward and forward. Here are examples of palindromes: "a", "aa", "aba", "abba". This string is not a palindrome: "abb". Note that if a palindrome is not empty, its first and last characters are the same.

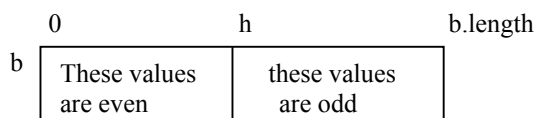
Write a function that has a String parameter c and returns the (**boolean**) value of the sentence "c is a palindrome". Do not write a function that reverses a string (or anything like that). If you write a loop, it will help you (and us) if you write the invariant of the loop, but you don't have to.

Question 3 (10 points). Arrays and methods. An array b of type **int**[][] is *rectangular* if all its rows have the same number of elements and *square* if the number of elements in each row is the same as the number of rows. Thus, a square array is also rectangular. Note that an array with 0 rows is square.

Write a method that has b as a parameter and returns 2 if b is square, 1 if b is rectangular but not square, and 0 otherwise. If you write a loop, it will help you (and us) if you write the invariant of the loop, but you don't have to.

Question 4 (09 points). Loops and invariants. Consider an **int** array b. We want a single loop (with initialization) to place all the even numbers in the beginning and all the odd numbers at the end. For example, the array

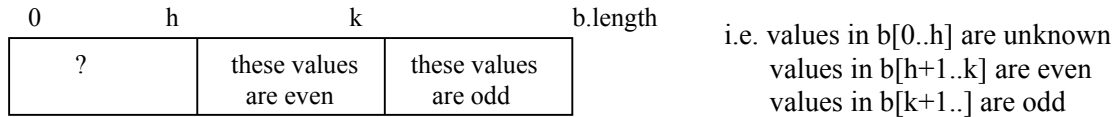
$b = (2, 4, -1, 0, 3, 0, -2)$ could be changed into $b = (-2, 4, 0, 2, 0, 3, -1)$



Question 0.	_____	(out of 1)
Question 1.	_____	(out of 10)
Question 2.	_____	(out of 10)
Question 3.	_____	(out of 10)
Question 4.	_____	(out of 09)
Question 5.	_____	(out of 10)
Question 6.	_____	(out of 15)
Question 7.	_____	(out of 20)
Question 8.	_____	(out of 15)
Total		(out of 100)

The postcondition is:

Write a single loop with initialization to perform this task. The loop must be constructed using the following invariant, or you get 0 points.



Question 5 (10 points). Matlab. Write a Matlab function that, given an argument n , produces the sum of the first n terms of this series:

$$1/3 - 2/4 + 3/5 - 4/6 + 5/7 - 6/8 + \dots$$

Question 6 (15 points). Miscellaneous class stuff

(a) Consider the classes on the right. Suppose the following variables have been assigned proper values.

Cx x; Cy y; Object z;

Tell whether each of the 5 following statements is legal or not. By legal, we mean that it will be compiled; by illegal, we mean that the program will not be compiled because of this statement

1. $y = x;$
2. $y = (Cy) x;$
3. $z = y;$
4. $x.setY();$
5. $x.toString();$

(b) Define the terms *apparent class* and *real class* of a class variable.

(c) Define the terms *parameter* and *argument*.

```
public class Cx {
    public void setX() { ... }
}

-----

public class Cy extends Cx {
    public String toString() { ... }
    public void setY() { ... }
}
```

Question 7 (20 points). Classes. Software company N needs the ability to generate new, unique registration numbers (which are **ints**), each of which is assigned to a person (a string, the person's name) and an associated product name (also a string, like "NWord", or "NExcel"). There should be a static **int** method *register* such that the call

`register("Kurland", "NWord")`

assigns a new, unique registration number r (say) to Kurland's copy of Nword. This call should return the entry $(r, \text{"Nkurland"}, \text{"Nword"})$ as an instance of class Entry, discussed below. Also, this call should record the entry $(r, \text{"Kurland"}, \text{"Nword"})$ in a Vector, which should contain all entries that were ever created by function *register*.

It should be possible to ask through a function call whether there is already a recorded entry with a given registration number r ; the function should return the entry with that registration number, if there is one. (What should it return if there isn't one? You must specify this.)

It should be possible to ask through a function call whether there is already a recorded entry $(_, n, p)$ —i.e. an entry with a given person n and product p ; the function should return

the entry with that person and product. (What should it return if there isn't one? You must specify this.)

Write a class named Entry. An instance of Entry is an entry consisting of a registration number, a person's name, and a product name. It should contain declarations of all the variables and methods necessary to do what is described above. Variables and method should be static or nonstatic, private or public, as required by good programming practices and to have the class work properly. *Don't write method bodies*. Instead, put good specifications as comments on the methods headers. Write headers of getter methods that you feel are needed; they don't need specs if the method names follow standard conventions.

Question 8 (15 points). Subclasses

Below is a class MacEntry, which extends the class of question 7.

- Fill in the body of the constructor of the class.
- Draw a folder (instance) of the class.
- Write the body of function equals. Use your knowledge of this class and class Entry to make it as simple as possible.

```
public class MacEntry extends Entry {
    private int macaddress= 0; // The address of the computer on which the software is placed

    /** Constructor: an Entry with registration no. r, name n, product p, and machine address m */
    public MacEntry (int r, String n, String p, int m) {

    }

    /** = this instance contains the same values as instance e
    public boolean equals(EntrySoftware e) {

    }
}
```

Answers

- ```
z= 1; int x= a; int y= b;
// inv: y ≥ 0 and z*xy = ab
while (y != 0) {
 if (y % 2 == 0)
 { y= y/2; x= x*x; }
 else { y= y-1; z= z*x; }
}
```
- ```
// = c is a palindrome
public static Boolean isPal(String c) {
    int h= 0; int k= c.length()-1;
    // inv: c is a palindrome if c[h..k] is
    while (h < k) {
        if (c.charAt(h) != c.charAt(k))
            return false;
        h= h+1; k= k-1;
    }
    return true;
}
```
- ```
// = 2 if b is square, 1 if b is rectangular,
// and 0 if b is not rectangular
public static int check(int[][] b) {
 if (b.length == 0) return 2;
 int x= b[0].length; // size of first row
 for (int k= 1; k != b.length; k= k+1) {
 if (x != b[k].length) return 0;
 }
 // { All the rows have length x }
 if (x == b.length) return 2;
 return 1;
}
```
- ```
h= b.length-1; k= b.length-1;
while (h != -1) {
    if (b[h] % 2 == 0) h= h-1;
    else { Swap b[h] and b[k]; h= h-1; k= k-1;}
}
```
- function ans = series(n);

```

alternatingOnes= - cumprod( - ones(1,n));
numerator= alternatingOnes .* (1:n);
denominator= 3:(n+2);
ans= sum(numerator ./ denominator);

```

6a. 1. illegal. 2. legal. 3. legal. 4. illegal.
5. legal, because toString is declared in class Object.

6b. The apparent class of variable x is the class with which it was declared. The real class of x is the class of the folder whose name is in x.

6c. A parameter is a variable that is declared within the parentheses of a method header. An argument is an expression that appears within the parentheses of a method call.

```

7. public class Entry {
    /** class invariant: numbers 0..nextNo-1
    have
    been used. The Entries in which they ap-
    pear
    are in Vector v */
    private static int nextNo= 1;
    private static Vector v= new Vector();

    private int registrationNumber;
    private String name;
    private String product;

    /* Constructor: a new Entry with reg. No. r,
    name n, and product p*/
    private Entry(int r, String n, String p)

    /* Create a new Entry (r, person, product),
    where
    r is a new, unique registration number, and
    return the Entry */
    public static Entry register(String person,
    String product);

    /* If a triple (r, _, _) is already recorded,
    return that triple. Otherwise, return null. */
    public static Entry getEntry(int r)

    /* If an Entry (_, person, product) is already
    re-
    corded, return it; otherwise return null */
    public static Entry getEntry(String n, String
    p);

    public int getRegistrationNumber()
    public String getName()
    public String getProduct()
}

```

```

8a. super(r, n, p);
    macaddress= m;

```

8b. The folder contains in its top partition the nonstatic components of Entry —registrationNumber, name, product, Entry(**int**,String,String), register(String, String), getEntry(**int**), getEntry(String,String), getRegistrationNumber(), getName(), getProduct()— and in its lower partition the nonstatic components of MacEntry —macaddress, MacEntry(**int**, String, String, **int**), equals(MacEntry).

8c. Here's the simplest possible body. It works because all folder have unique registration numbers:

```

return this == p;

```

You can also test the fields, but then you have to make sure that p is not null:

```

p != null && getRegistrationNumber() !=
    p.getRegistrationNumber; &&
etc.

```