

## CS100M: Section Exercises for Feb. 21,22

1. For each of the following sub-problems, complete the program below so that it produces the desired result. You should not modify the programs in any way, only fill in the blanks that are provided. In every case you will need to use for loops with step other than one.

(a) The following program reads an integer  $k$ , and outputs all the multiples of  $k$  up to 1000.

```
k = input('Please enter a positive integer smaller than 1000: ');
for j = -----
    fprintf('%d ', j);
end
fprintf('\n');
```

(b) The following program reads in a real number  $x$  and an integer  $N$ , and computes the sum  $\sum_{k=0}^N \frac{(-1)^k x^{2k}}{(2k)!}$  to the first  $N$  terms. (This sum converges to  $\cos(x)$  as  $N \rightarrow \infty$ .)

```
x = input('Please input a real number between 0 and pi/2: ');
N = input('Please input a positive integer: ');
sum = 0;
for j = -----
    sum = sum + (-1)^(j/2) * x^j / factorial(j);
end
fprintf('The sum of the first %d terms is %12.8f\n', N, sum);
```

(c) The following does the same thing as in part (b), but this time we are not allowed to use exponentiation and the `factorial` function, and must compute these explicitly.

```
x = input('Please input a real number between 0 and pi/2: ');
N = input('Please input a positive integer: ');
sum = 1;    % Explicitly assign the first term (when j=0)
sign = 1;   % The sign of a term, either 1 or -1
jfact = 1;  % Current value of j!
xtoj = 1;   % Current value of x^j
for j = -----
    sign = -----;
    jfact = -----;
    xtoj = -----;
    sum = -----;
end
fprintf('The sum of the first %d terms is %12.8f\n', N, sum);
```

## 2. Sub-arrays

### 2.1 Examining a sub-array

Write a function `vectorQuery(v,n,r)` to determine whether the number `r` appears in the first `n` cells of vector `v`. The function returns 1 if `r` is in the first `n` cells of `v` and 0 otherwise. (Preconditions: `v` is a vector of numbers; `n` is a positive integer; `r` is a number)

### 2.2 Returning an array

Write a function `sequence(m)` that generates a sequence of random *integer* numbers between 1 and `m`, inclusive, stopping when a value is repeated for the first time. The function returns an array containing all the numbers generated (in the order in which they were generated) except for the last value that is a repeated occurrence.

Example: If the generated sequence is 3 1 9 5 7 2 5, the array to be returned should be 3 1 9 5 7 2.

*Hints:* The symbol for an empty array is `[]`. When "building" an array, the space or comma separator puts two items side by side—creates a row.

**3.** Add code below so that the vector `fib` contains all the Fibonacci numbers up to and including the first such number that is greater than one million. You should not use any variables other than `fib`, nor make any calls to the `length` function. Instead, build `fib` by vector growing, and make use of the `end` keyword (note that `end` has special meaning when used as an array index).

```
fib = [0, 1];  
% Add the necessary code here
```