Eng101
Sample Final
Fall 2004

You may use any printed materials or notes on the exam. You may not use any electronic devices.

Unless otherwise specified you can implement algorithms in either C++or Matlab.

Printed Name:

8 Digit UM ID Number: ☐☐☐☐ – ☐☐☐☐

Lab Section:

Please sign the Engineering Honor Code:

I have neither given nor received aid on this examination.

Question 1: What are the values of x and y after the following Matlab statements are executed:

Score: xx

```
x = ones(5,5);
x(2:4,2:4) = 2.0 * ones(3,3);
y = sum(ones(3,3)) .* (1:3);
```

Question 2: Consider the following function (the C++ and Matlab functions are equivalent)

Score: xx

```
double ak(double x,
          unsigned int n)
{
   double result = 1;
   while(1 <= n)
    {
      if (0 == n % 2)
       {
          n = n/2;
          x = x*x;
       }
      else
       {
          n = n - 1;
          result = result*x;
       }
   }
   return result;
}
```

```
function result = ak(x, n)

result = 1;
while(1 <= n)
 if (0 == rem(n,2)) % rem(n,m): remainder n/m
    n = floor(n/2);
    x = x*x;
 else
    n = n - 1;
    result = result*x;
 end
end
```

What does it return for `x = 2.0` and `n = 4`? For `n = 5`? Write a single line of C++ or Matlab that does the same thing as `y = ak(x,n);`.

Question 3: A polynomial

Score: xx

$$P(x) = a_0x^0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n$$

can be stored in a computer as an array of coeffcients $\{a_0, a_1, \cdots a_n\}$ of length $n + 1$. The "derivative" of P is the polynomial

$$P'(x) = a_1x^0 + 2a_2x^1 + \cdots + (n-1)a_{n-1}x^{n-2} + na_nx^{n-1}\ .$$

Write a function that takes a polynomial (as an array of coeffcients) and returns its derivative (as an array of coeffcients). Be sure to deal correctly with a polynomial like $P(x) = 1$, where $P'(x) = 0$ is still a polynomial with one coeffcient (that just happens to be zero).

Question 4: I am trying to compute the cosine of angles using the expression

Score:

xx

$$\cos(\theta) = 1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} \ldots (-1)^n \frac{\theta^{2n}}{(2n)!} \ldots = \sum_{n=0}^{\infty} (-1)^n \frac{\theta^{2n}}{(2n)!}\,.$$

For this purpose I have partially written the following code. Please complete it.

```
function trigSum = Cosine(angle, epsilon)

trigSum = 0.0;      % estimate of cosine
nextTerm = 1.0;     % next term to add into sum
f = 0;              % f! will be built in denominator

% Do one step outside the loop so there will be previous
% term included in the sum

trigSum = trigSum + nextTerm;    % add term
previousTerm = nextTerm;  % record what was added
nextTerm = -nextTerm * [                    ]; % for next time
f = [          ]; % for next time
while abs(previousTerm) [      ] epsilon
   trigSum = trigSum + nextTerm;
   previousTerm = nextTerm;
   nextTerm = -nextTerm * [                    ];
   f = [          ];
end
```

Question 5: The discrete Laplacian is a measure of the curvature of data on a grid. Given a rectangular array of values, and calling a typical value $a_{i,j}$ at the grid point i, j, the discrete Laplacian at i, j is

Score:

xx

$$L_{i,j} = a_{i-1,j} + a_{i+1,j} - 4a_{i,j} + a_{i,j-1} + a_{i,j+1}$$

Write a routine `Laplacian` that computes this at all points on a grid, including the edges, with the understanding that any missing data needed to compute the Laplacian at the edges of the array is zero. `Laplacian` should take a two-D array of doubles as input, and return a two-D array of double of the same size.