PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

Midterm Examination: Semester 1 **Academic Year**: 2002-2003

Date: 28th July 2002 **Time**: 9.00 – **11.00** (**2 hours**)

Subject Number: 240-311 **Room**: R300

Subject Title: Mathematics for Computer Engineering

Lecturer: Aj. Andrew Davison

Exam Duration: 2 hours

This paper has 3 pages.

Authorized Materials:

- Writing instruments (e.g. pens, pencils).
- Books (e.g. dictionaries) and calculators are **not** permitted.

Instructions to Students:

- *Answer questions in English.* Perfect English is **not** required.
- Attempt all questions.
- Write your answers in an answer book.
- Start your answer to each question on a new page
- Clearly number your answers.
- Any unreadable parts will be considered wrong.
- When writing programs, use good layout, and short comments; marks will not be deducted for minor syntax errors.
- The marks for each part of a question are given in brackets (…).

1. Use induction to show that each equation is true:

a) $\frac{1}{1*2}+\frac{1}{2*3}+...+\frac{1}{(n-1)n}=1-\frac{1}{n}$, *when* $n \geq 2$ (11)

b) $(\cos x + i \sin x)^n = (\cos nx + \mathrm{i} \sin nx)$, *when* $n \geq 1$ (14)

Hint: use the following equalities:

$\cos(a + b) = \cos a \cos b - \sin a \sin b$

$\sin(a + b) = \sin a \cos b + \cos a \sin b$

(25 minutes; 25 marks)

2. Consider the following C function:

```
void foobar(int a, int d)
{
  int r = a;
  int q = 0;
  while (r >= d) {
    r = r-d;
    q = q+1;
  }
  printf("q=%d; r=%d\n", q, r);
}
```

The loop invariant S(k) is $d*q_k + r_k = a$, where $q_k = k$ and $r_k = a - d*k$ are the values of `q` and `r` after k iterations of the loop. `a` and `d` are both positive integers.

a) Prove that the loop invariant is correct, by induction on k. (5)

b) Give some examples of the output produced when `foobar` is called with different arguments. (2)

c) Say in words what `foobar` does. (3)

(10 minutes 10 marks)

3. Consider the sequence $\{a_0, a_1, a_2, \ldots\}$ defined by:

$a_0 = 1$ $a_1 = 3$ $a_2 = 5$

$a_n = a_{n-1} * (a_{n-2} * a_{n-2}) * (a_{n-3} * a_{n-3} * a_{n-3})$

a) Write a *recursive* C function which returns the n^{th} value in the sequence. (10)

b) Write an *iterative* C function which returns the n^{th} value in the sequence. (15)

c) Compare the functions of part (a) and (b), and say in words which is more efficient. Explain your decision. *Hint*: efficiency in this case means the amount of memory used to store data. (10)

(35 minutes; 35 marks)

Question 4 on next page.

4. a) Work out the worst case big-oh running time for the following *recursive* function. Show all your working. (25)

```
void sort(int A[], int n)
{ int imin, i;
  if (n > 1) {
    imin = 0;
    for (i=1; i < n; i++)
      if (A[i] < A[imin]) imin = i;
    swap(A, n-1, imin);
    sort(A, n-1);
  }
}
```

Note: you do **not** have to implement `swap()`. Assume that `swap()` has a constant running time.

b) Rewrite `sort()` to use loops instead of recursion. The new version should use the same input arguments as in part (a). Do not implement `swap()`. (15)

c) Work out the worst case big-oh running time for the iterative version of `sort()` from part (b). Show all your working. (5)

d) Compare the big-oh values for parts (a) and (c). Explain in words what the comparison means. (5)

(50 minutes; 50 marks)

--- End of Examination ---