NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

SEMESTER II: 2009–2010

EXAMINATION FOR

GEM 1501 – Problem Solving for Computing

April 2010 – Time Allowed 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of TEN (10) questions and comprises ELEVEN (11) printed pages.
2. Answer **ALL** questions.
3. This is a **Closed Book** examination.
4. Every question counts FIVE (5) marks which are distributed equally on sub-questions in the case that there are any. The maximum possible marks are 50.
5. Please write your Matriculation Number below:

MATRICULATION NO: SOLUTIONS

This portion is for examiner's use only

Qestion	Marks	Remarks	Qestion	Marks	Remarks
Q01:			Q06:		
Q02:			Q07:		
Q03:			Q08:		
Q04:			Q09:		
Q05:			Q10:		
			Total:		

Question 1 [5 marks] **GEM 1501**

Answer the following questions on the history of computing.

(a) Who wrote the first algorithm?
☐ Adam Riese ☐ Blaise Pascal ☒ Euclid ☐ Leonhard Euler

(b) Which of the following people constructed a *mechanical* calculator?
☒ Blaise Pascal ☐ Herman Hollerith ☐ Konrad Zuse

(c) For which type of technique were punch cards used first in the 19th century?
☐ Building bridges ☐ Controling railway engines ☒ Mechanical weaving

(d) What was the name of the machine which Charles Babbage wanted to build?
☒ Analytical Engine ☐ Eniac ☐ Enigma ☐ HAL

(e) Which of the following programming languages is the oldest?
☐ Algol ☐ Basic ☐ C ☒ Fortran ☐ Java Script

Question 2 [5 marks] **GEM 1501**

How many solutions have the following formulas in the variables x_1, x_2, x_3, x_4.

(a) $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_4 \vee \neg x_4)$:

This formula has <u> 12 </u> solutions.

(b) $(x_1 \wedge x_2 \wedge x_3) \vee (x_2 \wedge x_3 \wedge x_4)$:

This formula has <u> 3 </u> solutions.

(c) $(x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_3 \vee \neg x_4)$:

This formula has <u> 4 </u> solutions.

(d) $(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee x_4) \wedge (\neg x_3 \vee \neg x_4)$:

This formula has <u> 0 </u> solutions.

(e) $(x_1 \vee x_2 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee \neg x_4) \wedge (x_1 \vee \neg x_2 \vee x_3 \vee \neg x_4)$:

This formula has <u> 13 </u> solutions.

Question 3 [5 marks] **GEM 1501**

Which of the following statements are known to be true, known to be false and unknown? Here A $\subseteq$ B means that every problem which is in the class A is also in the class B; A $\subset$ B means that A $\subseteq$ B and, in addition, there is a problem in B which is not in A.

(a) P $\subseteq$ NP: [x] true [ ] false [ ] unknown;

(b) P = LOGSPACE: [ ] true [ ] false [x] unknown;

(c) P $\subset$ EXPTIME: [x] true [ ] false [ ] unknown;

(d) LOGSPACE = PSPACE: [ ] true [x] false [ ] unknown;

(e) P $\subset$ NC: [ ] true [x] false [ ] unknown.

Question 4 [5 marks] **GEM 1501**

Which of the following sets are decidable, recursively enumerable but undecidable and not recursively enumerable at all?

(a) The set of all Java Script programs P such that $P(888)$ terminates and outputs the value 333:

☐ decidable ☒ r.e. and undecideable ☐ not r.e.

(b) The set of all Java Script programs P such that there is a further Java Script program Q and some text R such that the sum of the lengths of Q and R is shorter than the length of P and Q with input R halts and outputs P:

☐ decidable ☒ r.e. and undecideable ☐ not r.e.

(c) The set of all Java Script programs which use at most 256 different variable names in its program text:

☒ decidable ☐ r.e. and undecideable ☐ not r.e.

(d) The set of all Java Script programs with 5 while-loops which do not terminate on input 555:

☐ decidable ☐ r.e. and undecideable ☒ not r.e.

(e) The set of all Java Script programs which halt on exactly 222 different inputs and which do not terminate on the other inputs:

☐ decidable ☐ r.e. and undecideable ☒ not r.e.

Question 5 [5 marks] **GEM 1501**

Determine the order of the following expressions:

(Example) $5 * n^2 + 22 * \log(n)$: $\underline{\quad O(n^2) \quad}$;

(a) $10 * n * \log(n) + 22 * \log^5(n)$: $\underline{\quad O(n * \log(n)) \quad}$;

(b) $75 * n^{20} * \log(n) + 88888 * \log^8(n)$: $\underline{\quad O(n^{20} * \log(n)) \quad}$;

(c) $(n + 1) * (n + 2) * (n + 3) * (n + \log(n))$: $\underline{\quad O(n^4) \quad}$;

(d) $(n^2 + n^3) * (n^2 - n^3) + n^6 + 777 * n^3$: $\underline{\quad O(n^4) \quad}$;

(e) $(n + \log(n) + 256)^{256}$: $\underline{\quad O(n^{256}) \quad}$.

Question 6 [5 marks] **GEM 1501**

Let `a` be an array with `n` entries such that for each `m < n` the entry `a[m]` has the fields `a[m].number` which is an index between `0` and `n-1`, `a[m].name` which is the name of the person and `a[m].salary` which is the income of this person. Furthermore, assume that each value of `a[m].number` occurs exactly once. How long does it take to solve the following tasks, where unit operations like accessing array elements and adding variables use up time 1? Give always the best possible answers and do not write $O(n^2)$ if a task can be done in $O(n \log n)$.

(a) Computing the sum `a[0].salary+a[1].salary+...+a[n-1].salary`:

☐ $O(1)$ ☐ $O(\log n)$ ☒ $O(n)$ ☐ $O(n \log n)$ ☐ $O(n^2)$

(b) Making an array `b` with fields `b[k].name` and `b[k].salary` such that for each `m` the values `a[m].name` and `b[a[m].number].name` as well as `a[m].salary` and `b[a[m].number].salary` are equal, respectively.

☐ $O(1)$ ☐ $O(\log n)$ ☒ $O(n)$ ☐ $O(n \log n)$ ☐ $O(n^2)$

(c) Making an array `c` with fields `c[k].name`, `c[k].number` and `c[k].salary` such that this array is a copy of the array `a` sorted according to the values of the salaries in ascending order.

☐ $O(1)$ ☐ $O(\log n)$ ☐ $O(n)$ ☒ $O(n \log n)$ ☐ $O(n^2)$

(d) Having the arrays `b` and `c` in addition to `a`, how long does it take to check whether there is a person with salary `d` for some input `d`?

☐ $O(1)$ ☒ $O(\log n)$ ☐ $O(n)$ ☐ $O(n \log n)$ ☐ $O(n^2)$

(e) Having the arrays `b` and `c` in addition to `a`, how long does it take to find the value `a[m].name` for the person with `a[m].number` being 263?

☒ $O(1)$ ☐ $O(\log n)$ ☐ $O(n)$ ☐ $O(n \log n)$ ☐ $O(n^2)$

Question 7 [5 marks] **GEM 1501**

Find the syntax errors and other errors in the following program for the factorial; here the factorial of 0 is 1 and the factorial of n with $n > 0$ is $1 * 2 * \ldots * n$. Write at each error the correct line below the erroneous line. There are in total 5 errors.

```
function factorial(int n)
function factorial(n)      // No types in Java Script

  { var m;


    var res = 0;
    var res = 1;           // Initial value was wrong

    if (n > 1)


      { res = 2; }


    for (m=3; m > n+1; m=m+1)
    for (m=3; m < n+1; m=m+1)  // Upper bound checked in wrong direction

      { res += m; }
      { res *= m; }            // Factorial consists of multiplications

    return(res,n); }
    return(res); }             // Result is only return-value
```

Question 8 [5 marks] **GEM 1501**

Write a counter program using the following commands: `x=y+1`, `x=y-1`, `x=y`, `x=0`, `goto ln`, `if x==0 then goto ln` and `if x>0 then goto ln` where "ln" stands for a line number and "x" and "y" stand for variable names. The counter program should never subtract 1 from a variable containing 0.

The function to be computed should be the remainder of x divided by y and the function should return 0 if y is 0.

```
function remainder(x,y)
  { 1. var u;
    2. var v;
    3. var w;
    4. u = 0;
    5. v = x;
    6. w = y;
    7. if w == 0 then goto 16;
    8. if v == 0 then goto 16;
    9. u = u+1;
   10. w = w-1;
   11. v = v-1;
   12. if w > 0 then goto 8;
   11. w = y;
   12. u = 0;
   15. goto 8;
   16. return(u); }
```

Question 9 [5 marks] **GEM 1501**

Write a function which counts how many numbers between inputs m and n are of the form x*x+y*y+z*z where x, y and z are positive natural numbers. For example, 6 is of this form as 6 is 1*1+1*1+2*2 but 5 and 7 are not of the desired form. The borders m and n have to be counted as well, whenever they are of the specified form.

```
function threesquare(m,n)
  { if (n<m) { return(0); }
    if (n<0) { return(0); }
    var k = 0;
    var x; var y; var z; var h; var c;
    for (h=m;h<=n;h++)
      { c = 0;
        for (x=1; x<=h; x++)
          { for (y=1; y<=h; y++)
              { for (z=1; z<=h; z++)
                  { if (x*x+y*y+z*z == h) { c = 1; } } } }
        k = k+c; }
    return(k); }
```

Question 10 [5 marks] **GEM 1501**

Assume that natural numbers of arbitrary size could be processed by the computer. Furthermore, let `randomnum(n)` be a function which returns a random number between `0` and `n` where each of the values `0,1,2,...,n` have the same probability. Furthermore, assume that two polynomials `f` and `g` of degree `999` are given. Write a computer program which returns `1` when `f` and `g` are the same and which returns with probability of at least $\frac{d}{d+1}$ a `0` when `f` and `g` are different.

```
function f(x)
  { var y; ....; return(y); }  // first polynomial

function g(x)
  { var y; ....; return(y); }  // second polynomial

function randomnum(n)
  { var m; ....; return(m); }  // draws an integer random number from 0 to n

SOLUTION ONE:
function polyequal(d)
  { var x = randomnum(1001*(d+1));
    if (f(x) == g(x))
      { return(1); }
    else
      { return(0); }
  } // this function should be implemented and test with high
    // confidence whether f and g are the same.

SOLUTION TWO:
function polyequal(d)
  { var x;
    for (x=0;x<1003;x=x+1)
      { if (f(x) != g(x)) { return(0); } }
    return(1);
  } // this function should be implemented and test with high
    // confidence whether f and g are the same.
```

END OF PAPER