National University of Singapore
Semester Examination

(Semester II)

GEK1536 – Computation and Machine: from ancient to modern

Apr/May 2005 – Time Allowed: 2 hours

Instructions to Candidates:

1. This is a closed book examination.
2. This examination paper contains SIX (6) questions and comprises THREE (3) printed pages.
3. Calculators are allowed, but steps should be given.
4. Total 100 marks.
5. Answer ALL the questions.

1. For each of the possible alternative statements, choose the one best fit the fact.
   (i) Babylonian base-60 number system is first used around
   (a) 5000 BC, (b) 2000 BC, (c) 11 AD, (d) 1000 AD.
   (ii) The ancient Greeks developed
   (a) base-20 system, (b) algebra, (c) geometry, (d) science.
   (iii) The Napier's bones are
   (a) toys for dogs, (b) for simplifying addition, (c) aid for multiplication, (d) for computing logarithms.
   (iv) Babbage's difference machine computes value of
   (a) differences, (b) sums, (c) polynomials, (d) divisions.
   (v) A quantum computer relies on
   (a) uncertainty principle, (b) deterministic principle, (c) quantitative principle, (d) superposition principle.

[15 marks]

Answers:

(i) (b) 2000 BC
(ii) (c) geometry
(iii) (c) aid for multiplication
(iv) (c) polynomials
(v) (d) superposition principle

3 marks each.

2. With one or two sentences, briefly describe the following concepts:
   (a) two's complement
   (b) positional number system
   (c) register
   (d) overflow
   (e) punched cards

[15 marks]

Answers:

(a) A scheme to represent negative numbers in binary bits. The rule: write the positive first, complementing all bits, add 1.
(b) A system to representing number. For a at i-th position, it denotes value $a \times b^i$, i=0,1,2,.
(c) Temporary memory of one word size inside central processing unit (CPU).
(d) When integer or floating point number is too big in a computation that its result cannot be presented within the representable range.
(e) A paper card to store information for computer to read.

3 marks each.

3. In Euclidean geometry, sum of the three internal angles of a triangle is 180°.
(a) based on this fact, derive the sum of the 4 internal angles of a quadrilateral (a four sided polygon).
(b) Generalize the derivation in part (a), derive a formula for the sum of internal angles for an n-sided polygon, for any $n = 3, 4, 5, \ldots$

[15 marks]

Answer:

(a) *Clearly, quadrilateral can be viewed as 2 triangles by drawing a line alone the diagonal. The sum of the internal angles of two triangles is the same as that of the original quadrilateral. So it is 360°. [7 marks]*
(b) *Every time we add a new triangle on the edge of a polygon, we increase the sides of polygon by 1, and increases the sum of internal angles by 180. So the formula is 180°(n-2). [8 marks]*

4. Design a Turing machine to determine if a binary encoded natural number is even or odd. When the machine halts, it erases the number on the tape and replaces it with the answer 0ΔΔΔΔ… if the number is even, or 1ΔΔΔΔ…, if the number is odd. The input number is put on tape with the least significant bit in the first cell [That is, we read the tape backward]. The symbol space will be 0, 1, the space Δ, and an extra dot •. The Turing machine rules will be specified by a diagram.

[15 marks]

Answer:
No matter what number it is on the tape, we start erasing the rest.

```
[start=1] -> (0,0,R) ->  [2] -> (0,Δ,R) -> (back to [2])
             (1,1,R)-> [2]       (1,Δ,R) -> (back to [2])
                                  (Δ,Δ,R) -> [halt=3]
```

assuming table always starts with a number.

5. (a) Convert the decimal number +255.125 into IEEE 32-bit single precision floating-point format. Note that the format has the highest bit for the sign, the next 8 bits for a biased exponent (the bias is 127), the rest of the bits for the fractional part of the number.
(b) Interpret the following bit pattern

0011 1111 1111 1111 1111 1111 1111 1111

in three ways (i) unsigned integer, (ii) signed integer, and (iii) floating point number (IEEE format). Find the three values in decimal notation.

[20 marks]

Answer:

(a) $255_{10} = 1111\ 1111_2$, $0.125_{10} = 1/8 = 0.001_2$, so $255.125_{10} = 11111111.001_2 = 1.1111111001 \times 2^7$

$f = .11111110010000\ldots0$

$E = 7+127 = 134 = 0111\ 111 + 111 = 1000\ 0110_2$

$s = + = 0$ *(positive)*

So the bit pattern is

0 1000 0110 1111111001 0000000000 000

[10 marks]

(b) (i) and (ii) $+2^{30}-1=1073741823$.

(iii) $s=0$, $E=127$, so $e=0$, $f = 1.1111\ldots.1_2 = 2 - 2^{-23} = 1.99999988079$ = *value of the IEEE number.*

[10 marks]

6. There are several ways that one can use to convert a decimal integer into binary digits. One possibility is to subtract a power of 2, starting from the largest. In this way, the most significant bit is found first and can be printed out. Implement this idea in a C program. The program first scans (that is, read) a non-negative integer. It then prints out the binary digits from most significant to least significant, one digit a time.

[20 marks]

Answer:

```
#include <stdio.h>
main()
{    int i, n;
     scanf("%d", &n);
     i = 1;
     while(i<n) {   /* find largest power  */
       i = i*2;
     }
   while(i > 0) {   /* subtract power, from large to small */
      if(n-i>=0) {
         printf("1");
         n = n – i;
    } else {
        printf("0");
     }
      i = i/2;
   }
   printf("\n");
}
```

– the End –