10%(I) Describe function name overloading and polymorphism. Illustrate both with examples. Explain their difference.

10%(II) Regarding C programs:
5%(a) Consider the following recursive function in C:

```c
void mystery(int n) {
    if (n>1) mystery(n/2); /* integer division */
    printf("%d", n%2);
}
```

(a-1) (3%) What is the output generated by mystery(25)?
(a-2) (2%) What task does this function perform?

5%(b) The following C code is intended to transpose (switch rows and columns) a square two-subscript array with n rows and n columns. Assume that the array (with name a) is of type int. However, this code does not generate the correct result. Show how you can correct it. Make as few changes as possible.

```c
int i, j, temp;
for (i=0; i<n; i++) {
    for (j=0; j<n; j++) {
        temp=a[i][j]; a[i][j]=a[j][i]; a[j][i]=temp;
    }
}
```

10%(III) Briefly answer the following questions:
4%(a) Explain and distinguish public-key Encryption and Digital signature.
4%(b) Write down the truth table of an exclusive-OR (xor) gate, and implement the exclusive-OR gate with NAND gates only.
2%(c) Assume the integer a contains 38 and integer b contains 49.

What would be the result of a and b after doing the following 3 statements:

- a = a xor b;
- b = a xor b;
- a = a xor b;

20%(IV) Answer the following questions (4 points each):
4%(a) Give examples of arithmetic overflow and underflow. Explain how they yield results that are different from what are expected.
4%(b) What is the purpose of the cache in a CPU? Also describe a criterion for selecting an existing entry in the cache to be replaced when we need to put a new entry into the cache.
4%(c) Give the one's complement and two's complement of the value -10 (minus ten) in 8-bit binary form. What is the benefit of using two's complements to represent negative integers?
4%(d) Explain the steps of "hand-shaking" when two computers try to establish a connection.
4%(e) Explain the cause of the phenomenon of thrashing, and its effect on CPU and disk usage.

10%(V) The quick sort and heap sort are NOT stable. While the merge sort is stable.

5%(a) What does that mean? Explain it with an example.
5%(b) Also list three other sorting algorithms and discuss their time complexity.
10%(VI) Briefly answer the following questions:

3%(a) Compare circuit switching and packet switching.
3%(b) What is a Rapid Prototyping Generator (RPG)?
4%(c) Explain RPC/RMI/CORBA and Web Service technologies.

10%(VII) Consider the standard library int rand( ) function.

1%(a) Why do we say it is a pseudo random number generator?
2%(b) The numbers generated by rand( ) are in uniform distribution.

What does that mean?
2%(c) The rand( ) can generate an integer random number between 0 to RAND_MAX.
Using rand( ), write a function double frand( ), so that
0 <= frand( ) < 1.0
2%(d) Using frand( ), write a function double grnd( ), so that
grnd( ) can generate random number of Normal distribution
with mean 0.0, and standard deviation 1.0
Hint: Use the above's frand( ) successively to get 12 random numbers and take the average which should be close to 6.0.
3%(e) Using grnd( ), write a function double hrnd( ), so that
hrnd(X, STD) can generate random number of Normal distribution
with mean X, and standard deviation STD.

10%(VIII) Briefly answer the following questions.

3%(a) Convert the decimal number 1234567.2 into binary.
3%(b) Convert the decimal number 1234567.3 into binary.
4%(c) Explain "Big endian" and "Little endian" with an example?

Hint: Consider the following C program:

```c
#include<stdio.h>
int main() {
    unsigned char buf[] ={1,2,3,4};
    short * p = (short*) buf;
    printf("==%d\n", *p); return 0;
}
```

Discuss the output of this program if you run it on a Big-Endian machine? and what if run it on a Little Endian machine?

10%(IX) Consider the following two infix expressions:

```
a+b*c/d-e/f
(a+b)*c/(d-e)/f
```

3%(a) Convert them into prefix and postfix expressions.
3%(b) Discuss their evaluation rules.
4%(c) Draw the expression tree for each of the two expressions.