1. Please explain the difference(s) between the initialization list of a constructor and normal member data initialization inside a constructor in a C++ program. Advantages and disadvantages of each approach should be listed. **(10%)**

    ```
    class ABC {
        int *a;int b;
        ABC(int *aa, int bb): a(aa), b(bb) {} // initialization list
        ABC(int c) { a=new int(c); b=c; } // normal member data initialization
    };
    ```

2. Please explain the following code fragment, especially why "foo" cannot be returned: **(10%)**

    ```
    Fraction& Fraction::operator=(const Fraction& foo) {
        if ( this == &foo ) return *this ;
        den = foo.den ;   num = foo.num ;
        return *this ;   }
    ```

3. For any positive real number x, its Engel expansion is defined as: **(30%)**

$$x = \frac{1}{a_1} + \frac{1}{a_1 a_2} + \frac{1}{a_1 a_2 a_3} + \dots$$

Calculation:

Let $u_1 = x$

$a_k = ceil(1/u_k)$

$u_{k+1} = u_k a_k - 1$

Repeat until $u = 0$

Note that ceil (r) will return the nearest integer greater than or equal to r.

Also note that $1 = 0.\overline{9} = 0.999\dots$

For example, x=0.123

u1=0.123	a1=ceil(1/0.123)=9	u2=0.123*9 - 1=0.107
a2=ceil(1/0.107)=10	u3=0.107*10-1=0.07	a3=ceil(1/0.07)=15
u4=0.07*15-1=0.05	a4=ceil(1/0.05)=20	u5=0.05*20-1=0

So, x=0.123=1/9+1/(9*10)+1/(9*10*15)+1/(9*10*15*20)

Engel = {9,10,15,20}

(1) Please write a program that can show the Engel expansion, given x.

(2) Please do not just write the codes. You need to develop your own version of "ceil()" and explain each step of the program.

(3) Please discuss potential use or applications of such the problem.

注意：背面有試題

4. Please clarify advantages and limitations of using "Pointer" in C/C++ Programming. (10%)

5. In C++ Programming, please clarify the difference between "class" and "struct". (10%)

6. Chose question:

(1). What's result of following code? (10%)

```
#include <iostream>
using namespace std;
int main ( void ) {
int x[5][3] = {{0,1,2},{3,4,5},{6,7,8},{9,10,11},{12,13,14}};
cout<< x[2][1]+x[3][2]; return 0;}
```

(a) 10 (b) 18 (c) 6 (d) 12

(2). What's correct description of following code? (10%)

```
#include <iostream>
using namespace std ;
void f1 (int &b){b=9;}
int main ( void ) {
int a=3；f1(a)；return 0;}
```

(a) a is 3, because a, as a Reference, is inputted in f1 function operation

(b) a is 3, because a, as a Pointer, is inputted in f1 function operation

(c) a is 9, because a, as a Reference, is inputted in f1 function operation

(d) a is 9, because a, as a Pointer, is inputted in f1 function operation

(3). What's correct description of following code? (10%)

```
#include <iostream>
using namespace std ;
int main() {
char a[] = "中央大學"; char b[] = "土木工程";
char c[20];
int i, j;
for(i=0; a[i]!='\0'; ++i) c[i] = a[i];
for(j=0; b[j]!='\0'; ++j) c[i+j] = b[j]; c[i+j] = '\0';
cout << c << endl;
return 1;}
```

(a) c=a=b;

(b) c[16]='\0';

(c) c[20]= '\0';

(d) c[15] ='\0';

注意：背面有試題