LAB 10/03

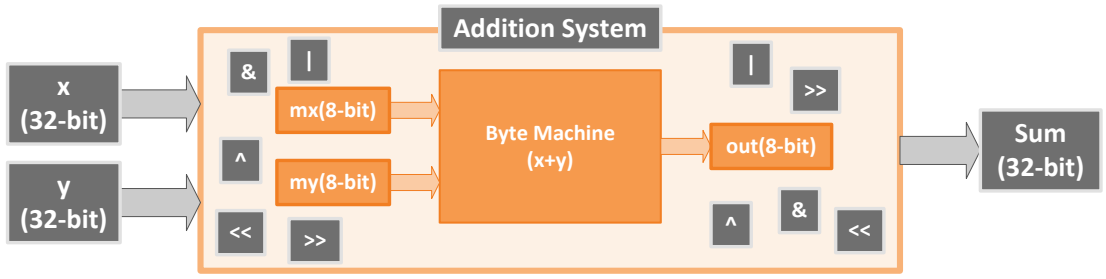
1. Find and correct both the syntax and logical errors of following program, and illustrate what this code does.

```
#inculde<stdio.h>

int mian(void)
{
	int r
	const float area
	const double PI=3.14159
	double 3v
	double x y

	printf("Please input the radius:');
	scanf("%d\n,r");
	area==PI*r^2;
	printf("The area of circle is %d\n\n",&area);

	printf(""Please input a fraction number:");
	scanf("%d\n,3v");
	printf("The decimal part of float number is %d\n,(int)3v");
	printf("The fraction part of float number is %lg\n\n,v-(int)3v")

	print("What's the difference between -1>>2 and -1/4 ??\n");
	printf("-1>>2 = %d	and	-1/4 = %d\n\n",-1>>2,-1/4);

	printf("Plaese input two integer x and y: '');
	scanf("%lf,%lf\n",x,y);
	printf("\"x\" / \"y\" = %x ..... %d",x/y,x%y);

	getch()
	return 0;
}
```

2. Write a program that prompts the user to input two numbers with data type double: *x* and *y*. Then output the value of *x (+,-,*,/) y* as the following example. Notice the spacing between numbers, please complete each formula with only one printf().

```
Input: 3.14159 1.8765432

12345678901234567890123456789012345678901234567890123456789012345678901234567890
   3.14159  = 3.14          + 00000.0015000000 + 0000.00009
00000003.1  +      1.87654  = 5.018
003.141590  * 1.8765432000  = 00000005.8953294
3.142       / 00000001.877  = 1.67414
```

3. An addition system can do addition with two "int" numbers whose lengths are both 32 bits (4 bytes) and then output an "int" result (32-bit, 4-byte). But this system has a "byte machine" which can only do addition with two numbers whose lengths are both 8 bits (1 byte) and then output an 8-bit result. Outside the byte machine, the system can only do bit operation, i.e., &, |, ^, >>, <<. Please implement this system as above requirement.




```c
#inculde<stdio.h>
int main(void)
{
    int x, y, sum;    // input: x,y ; output: sum
    unsigned char mx, my, out;    // machine input: mx,my ; machine output: out
    bool ca=0;  // carry

    // The rest part you have to finish...

    return 0;
}
```