Problem 1

The original code:

```
#include <iostream>
using namespace std;
int main()
{
     cout << "Hello World! ’’;
     cout << "This is Tom’s first program, ";
     cout << ""stored at C:\Users\User\Documents."";
     return 0;
}
```

should be corrected as:

```
#include <iostream>
using namespace std;
int main()
{
     cout << "Hello World! ";
     cout << "This is Tom’s first program, ";
     cout << "stored at C:\\Users\\User\\Documents.";
     return 0;
}
```

1.On line 5: A string should be put between quotation marks, so two apostrophes (’) should be replaced with one quotation mark(“).
2.On line 7: To make the output exactly the same with required and legal, two quotation marks should be removed.
3.On line 7: Backslashes(\) are special characters in strings, and to print out one it should be put with an escape character, which is backslash itself.

Problem 2

(a)

```
#include <iostream>
using namespace std;
int main()
{
     int p;
     int q;
     cin >> p;
     cin >> q;
     while (p % q != 1)
     {
          int r = p % q;
          cout << r << " ";
          p = q;
          q = r;
     }
     cout << "\n" << p;
     return 0;
}
```

(b)

```
26 9 8
9
```

(c)

This program gets two inputs which are integers, and for every time in the while loop, it print out the remainder of p divided by q, assign q to p, and assign the remainder to q, it keeps doing it until p and q are co-prime. So this program is basically doing Euclidean algorithm and print out p in the end.

Problem 4

```
#include <iostream>
using namespace std;
int main()
{
     int p = 0;
     int q = 0;
     cin >> p;
     cin >> q;
     while (p % q > 0)
     {
          int r = p % q;
          cout << r << " ";
          p = q;
          q = r;
     }
     cout << "\n" << q;
     return 0;
}
```

1.
(1)All variables are initialized.
(2)The program can already accept cases in which second number is greater than the first one.

2.
The program goes wrong when p can be divided by q, which means the remainder of p divided by q is 0, when it is assigned to q, the next execution of p % q in the loop will go wrong since nothing can be divided by 0. So the control statement should be p % q > 0 to prevent this case.
And since this program is doing Euclidean algorithm, printing p out is meaningless, so I guess it should print out greatest common divisor, which is q in the end in this modified program.