1. Number the following steps from 1 to 4 in the order they should be performed (10 points)

_3_ Write the program.
_4_ Test the program.
_2_ Write the design.
_1_ Write the specification.

2. What do the following print? (3 points each)

```cpp
string s = "ABCDEF";
cout << s[2]; // C
cout << s.substr(3, 2); // DE
cout << (s + 'x' + s)[7]; // A
cout << int(s.size()); // 6
cout << char(tolower('D') + 1); // e
```

3. Write a function `count(s, c)` that returns the number of times the character `c` occurs in string `s`. For example (20 points):

```cpp
int n = count("hello world", 'l');
cout << n; // 3
```

```cpp
int count(string s, char c)
{
    int n = 0;
    for (int i=0; i<int(s.size()); ++i)
        if (s[i] == c)
            ++n;
    return n;
}
```

4. Write a function `append` that takes two strings and appends the second one to the first. The second string is unchanged. The function should not return a value. For example (15 points):

```cpp
string a = "hello", b = "world";
append(a, b);
cout << a; // helloworld
cout << b; // world
```

```cpp
void append(string& a, string b)
{
    a += b;
}
```

5. Write a program that reads from standard input to standard output (from cin to cout) searching for strings enclosed in parenthesis and printing them. Nothing else is printed. For example, if the input is

```
This(is)a(test).
This is another (very little) test.
```

then the output is

```
(is)test(very little)
```

First, draw a state machine diagram for a program that reads one character at a time (10 points).

(Hint: you only need 2 states).

```
+-------+   (     +-------+ not )
| not ( |------->|     |---------+
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>start</td>
<td>print c</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>print c</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Now write the program (30 points).

```cpp
#include <iostream>
using namespace std;

int main()
{
    char c;
    int state = 1;
    while (cin.get(c))
    {
        if (state == 1) // outside (...)
        {
            if (c == '('
            {
                cout << c;
                state = 2;
            }
        }
        else if (state == 2) // inside (...)
        {
            if (c == ')')
            {
                cout << c;
                state = 1;
            }
        }
    }
    return 0;
}