1. Rewrite the following code using proper indentation (10 points).

```
int i=1;while(i<10){if(i%3==0)
break;else ++i;}cout<<i<<endl;
```

```
// ANSWER
int i=1;
while (i<10)
{
  if (i%3==0)
    break;
  else
    ++i;
}
cout<<i<<endl;
```

2. What does the code in problem 1 print (5 pts)?

**ANSWER: 3**

3. What do the following print (5 pts each)?

```
cout << 3.5 + 1/2 << endl;
```

**ANSWER: 3.5**  (1/2 is integer division)

```
int i, j = 5;
for (i = 2; i < j; --j)
  cout << i*j << ' ';
```

**ANSWER: 10 8 6**

```
for (char c = 'A'; c <= 'F'; c += 2)
  cout << c;
```

**ANSWER: ACE**

```
int x = 4;
if (x = 3)
  cout << x*1.5;
else
  cout << x*10;
```

**ANSWER: 4.5** ( = is assignment)

```
int x = 4;
if (x == 3);
  cout << x;
```

**ANSWER: 4** ( ; is an empty statement)

```
string s = "ABC";
cout << (s+s+s).substr(2, 4);
```

**ANSWER: CABC**

```
string s = "ABC";
s[1] = 'R';
cout << s;
```

**ANSWER: ARC**

4. Suppose the file **test.txt** contains a single line of text (7 characters including a newline) as follows:

```
A TEST
```

For each problem below, assume that the input has been redirected to test.txt (as in **a < test.txt**, where the compiled program is **a.exe**).  What do programs containing the following code print? (5 pts each)

```
char a, b, c;
cin >> a >> b >> c;
cout << a << b << c;
```

**ANSWER: ATE**

```
char c;
while (cin.get(c))
  cout << c;
```

**ANSWER: A TEST**

```
string s;
while (cin >> s)
  cout << s << endl;
```

**ANSWER:**
**A**
**TEST**

5. Write a program that prints all of the 4 digit integers from smallest to largest, except those whose last digit is 0 or 5.  Print 8 numbers on each line. The first 2 lines and last line should look like this (35 pts):

```
1001 1002 1003 1004 1006 1007 1008 1009
1011 1012 1013 1014 1016 1017 1018 1019
...
9991 9992 9993 9994 9996 9997 9998 9999
```

```
// ANSWER
#include <iostream>
using namespace std;
int main()
{
  for (int i=1000; i<10000; ++i)
  {
    if (i%10 == 0)
      cout << '\n';
    else if (i%5 != 0)
      cout << i << ' ';
  }
  return 0;
}
```