Programming/Data Structures Mid-term test 10/10/2005

1. Provide the function prototypes and function definitions in the following program. Note that function overloading is used to define the function *Report* in three different ways.

```
/*********************************************
This program reads the integers in a file and outputs appropriate  messages to console. For each
integer read, the program reports the value read, and the sum of integers read up to that point.
When all integers are read, the program reports the total number of integers read and the grand
sum. A sample output is shown below.
*********************************************/
#include<iostream>
#include<fstream>

//Provide function prototypes

int main() {
  int a,b;
  double sum;
  ifstream input;
  //open text file for reading
  input.open("input.txt");
  //initialize values
  b=0; // b counts the number of integers read from file
  sum=0; //sum of the integers in the file - sum is made double, just in case.
  while (input>>a){
    Incrementcounter(b); /* Increment the value of b by 1.  Provide function definition. */
    sum +=a; //add current value of a to sum
    Report(a,sum); /** Provide function definition. Write a message to console
                       in the format "Read a; sum became (new sum value)" */
  }
  Report(b); /** Write to console the number of integers read.
                 Provide function definition, if needed. Output format is
                 "Read a total of (number) integers" */

  Report(sum); /** Write to console the final sum in the form
                   "The sum of all integers in your file is : (value)"
                   Provide function definition, if needed. */
  input.close();

}
/*********************************************************************
A sample output of the program is:

Read 12; sum became 12
Read 1; sum became 13
Read 13; sum became 26
Read 19; sum became 45
Read a total of 4 integers
The sum of all integers in your file is : 45

*********************************************************************/
```

2. Write a recursive C++ function *void f(int n)* that prints to console a pattern of the type:

```
1 1
1 1 2
1 1 1 3
1 1 1 1 4
.....
1 1 1 .....n
```

Here's a test program for the function:

```
#include<iostream>
using namespace std;
void f(int);
int main(){
  int n;
  cout <<"Enter an integer\n";
  cin>>n;
  f(n);
}
```

A sample output of the program is:

```
linux2:~/PDS/C++$ ./test1-2
Enter an integer
5

1 1
1 1 2
1 1 1 3
1 1 1 1 4
1 1 1 1 1 5
```