Examination, Programming, bridging course, 2009-12-17
Time, 8.00-13.00

Materials allowed: None

Each problem is worth 5 points.

Problems number 5 and 6 are for C++-students and problems number 7 and 8 are for FORTRAN-students. You do either 5 and 6 OR 7 and 8 but NOT both.

Problems 1, 3, 5 and parts of 7 and 8 are for grade 4. Thus to get grade 4 you can ignore the other problems.

To get the grade 4 you should complete 2/3 of the problems marked for grade four. To get the grade 5 you must complete 2/3 of all the problems.

If you achieve neither, you will get grade 3.
1. (a) What is the primary difference between a list and a tuple in Python?
   (b) What can you do with the Pickle module? When is this useful?
   (c) The time performance of your Python program is not good and needs to be improved. How can you proceed?

2. (a) The following function takes a list as input, what is the function output? How does the function work? Especially, what is happening at the 'yield' statements?

```python
def function(input_list):
    if len(input_list) <= 1:
        yield input_list
    else:
        for elem in function(input_list[1:]):
            for i in range(len(elem)+1):
                yield elem[:i] + input_list[0:1] + elem[i:]
```

(b) Below is a function that was intended to take a list of values and remap them to a new range.

```python
def remap(input_list, new_start, new_end):
    min_val = min(input_list)
    max_val = max(input_list)
    for elem in input_list:
        elem = (elem-min_val) / (max_val-min_val) * \n        (new_end-new_start) + new_start
A = [0, 20, 50, 10]
remap(A, 0, 5)
print A
```

The test input values:

0, 20, 50, 10

were supposed to result in these new values:

0, 2, 5, 1

as the new range was set to 0-5. However, the remap function leaves A unchanged. What are the two causes for not getting the wanted output? What changes could be done to make it work like intended?
3. To convert a character string that contains a floating point number to double in C, you use the `double atof(const char*)` function.

You can call it like this

```c
double d = atof("-1.234125");
```

Your problem is to implement the function

```c
double atof(const char* s){
    code here
}
```

to perform the conversion described above.

To simplify a bit you can assume that fixed notation is used, i.e. numbers look like `-4.32`, not like `1.23e-4`. There is no exponent part, an optional leading sign is allowed. There need not be a decimal point, thus `-2` is allowed.

Your function should detect syntax errors in the input string. If an error is detected, some action that you design should be taken.

4. A matrix in C can be used as parameter to a function, but you have to specify all dimensions but the first. If we need a function that can check if a 10x10 matrix has diagonal dominance we can declare it like this, where `a` is the matrix.

```c
boolean diagonal_dominant(double a[][10]);
```

This function can only test matrices of the size 10x10, this is not very general. We would like a function that can test this for a square matrix of any size.

Your task is:

(a) Design a C-function that can do this. Describe parameters, result and a method that can used to implement it.

(b) Implement your function. As stated above, it should take a square matrix of arbitrary size, test it and return the result of the test.

Some hints:

- A square matrix has diagonal dominance if
  \[ |a_{ii}| > \sum_{j \neq i} |a_{ij}|, \text{ for all } i \]  

  where `i` is the line index and `j` is the column index.

- To get a pointer to a matrix `a`, that has NO structure information attached, you need to use `&a[0][0]`. The pointer `a` is seen as pointer to the first line, and this line has a length.
5. Assume that we have the following class defined in C++

```cpp
// a coordinate in (x,y)-space
#include <cmath>

class Point {
    private:
        double x,y;
    public:
        Point():x(0),y(0) {}
        Point(double xp, double yp):x(xp),y(yp) {}
        double getX() {return x;}
        void setX(double xp) {x = xp;}
        double getY() {return y;}
        void setY(double yp) {y = yp;}
        // distance from me to the point p
        double distance(const Point& p) {
            return sqrt((x-p.x)*(x-p.x) + (y-p.y)*(y-p.y));
        }
};
```

Write a class `Triangle` that describes a triangle using the objects of the type `Point`. The class should contain constructors so that we can create objects of the type Triangle. It should also contain a member function `area` that computes and returns the area of the triangle represented by the object.

You can compute the area using Heron’s formula

\[
area = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)}
\]

\[
s = \frac{1}{2}(a + b + c)
\]

where \(a, b \text{ and } c\) are the length of the three sides in the triangle.

6. Write a stack in form of a template class. A stack is a LIFO (Last In First Out) queue. The stack should have one method for adding elements to the top and one for getting the topmost element. The maximum number of elements that the stack can hold should be a parameter to the constructor. The class should use an array allocated using new to store the elements. Here is a starting point:
template<class T>
class Stack
{
    // Member variables
    ..

public:
    // Constructor and destructor
    ..
    // The functions for adding and receiving elements
    void push(T x) {...}
    T pop() {...}
};

It should be possible to use the stack like this:

int main(int argc, char **argv)
{
    Stack<int> stack(100);
    stack.push(3);
    stack.push(5);
    stack.push(2);
    int first = stack.pop();
    int second = stack.pop();
    cout << first << " " << second << endl;
}

The syntax for allocating and deallocating an array with new is:

a = new int[n];
delete [] a;
7. Write a FORTRAN subroutine declared as

```fortran
subroutine findprime( a, b )
```

Here, `a` is an array containing some prime numbers, and `b` is an array (possibly long) containing positive integers. All elements in `b` that are evenly divisible with at least one of the prime numbers in `a`, should be changed to a zero. At return, the changed array `b` should be accessible to the calling program.

For 4 points on this problem (aiming for grade 4), the array `b` can be considered to contain numbers in any order. For 5 points (aiming for grade 5), the array `b` should be considered to contain *consecutive* numbers, (the first element is not necessarily equal to 1), and the code should take advantage of this fact.

8. A data file is structured so that it contains data values in *sections* as shown in the following example:

```
12 4
.
12 lines with data
.
8 3
.
8 lines with data
.
-1 -1
```

Note that the file above is just an *example* of how the file could look like, there can be an arbitrary number of sections. Each section is headed by a line containing two integer values, the first one giving the number of data lines in the section, the second one the number of data items on each line, (there are never more that 8 data items per line). The data items are floating-point numbers. After the last section comes a line with the value -1 occuring two times.

Write a FORTRAN main program that reads the file, and for each section of data finds the maximum value contained in that section. The number of the section and the maximum value should be written to standard output.

For 4 points, (aiming for grade 4), the problem can be solved according to the specifications above. For 5 points (aiming for grade 5), the program should handle the situation that the file does not contain a trailing line with -1 -1, but instead end-of-file follows directly after the last section.