Fall 2012 BIOSTAT 615/815 Problem Set #1

Due is Saturday September 22nd, 2012 11:59PM by google document (shared to hmkang@umich.edu and atks@umich.edu) containing the source code and running examples, and email of the compressing tar.gz file containing the source codes

Problem 1. One-sided Fisher's Exact Test (30 pts)

Implement a program `oneSidedFastFishersExactTest`, as a modified version of the `fastFisherExactTest` presented in the class, with the following requirements.

1. When more than or less than 4 input arguments were specified, print an error message and terminate the program.
2. Two-sided test calculated the following p-value

$$p_{two-sided}(a,b,c,d) = \sum_x \Pr(x) I[\Pr(x) \leq \Pr(a)]$$

   In one-sided test, the following p-value needs to be calculated.

$$p_{one-sided}(a,b,c,d) = \sum_{x \geq a} \Pr(x)$$

Below is an example of expected output of the program.

```
user@host:~/> ./oneSidedFastFishersExactTest
Usage: oneSidedFastFishersExactTest [a] [b] [c] [d]

user@host:~/> ./oneSidedFastFishersExactTest 2 7 8 2
One-sided log10(p) = -0.000428027, p-value = 0.999015

user@host:~/> ./oneSidedFastFishersExactTest 7 2 2 8
One-sided log10(p) = -1.73232, p-value = 0.0185217
```

In your google document, include full source code and example output. Your source code name must be hw-1-1.cpp. Do not include any other files in your .tar.gz submission.

Problem 2 - Pointers and Arrays (30 pts)

Consider the following program `hw-1-2.cpp`.

```
#include <iostream>

int main(int argc, char** argv) {
  int nv = argc;
  int& nr = argc;
  int* pr = &argc;

  char** ppc = argv;
  char*  pc = *argv;
  char   c1 = **argv;
  char   c2 = argv[1][2];

  std::cout << "argc = " << argc << std::endl;
  std::cout << "nv = " << nv << std::endl;
  std::cout << "nr = " << nr << std::endl;
  std::cout << "pr[0] = " << pr[0] << std::endl << std::endl;

  std::cout << "pc = " << pc << std::endl;
  std::cout << "ppc[0] = " << ppc[0] << std::endl;
  std::cout << "argv[0] = " << argv[0] << std::endl;
  std::cout << "c1 = " << c1 << std::endl;
  std::cout << "c2 = " << c2 << std::endl << std::endl;

  nr = 10;
  ++argv;

  std::cout << "argc = " << argc << std::endl;
  std::cout << "nv = " << nv << std::endl;
  std::cout << "nr = " << nr << std::endl;
  std::cout << "pr[0] = " << pr[0] << std::endl << std::endl;

  std::cout << "pc = " << pc << std::endl;
  std::cout << "ppc[0] = " << ppc[0] << std::endl;
  std::cout << "argv[0] = " << argv[0] << std::endl;
  std::cout << "c1 = " << c1 << std::endl;
  std::cout << "c2 = " << c2 << std::endl << std::endl;

  return 0;
}
```

(a) What is the output of the following program `hw-1-2.cpp` when you run the following command?

```
user@host:~/> ./hw-1-2 Hello World
```

(b) Briefly explain why each output line changes or does not change between the first and second half of the outputs. You do not need to include the source code for this problem.

Problem 3 - Calculating Simple Statistics (40 pts)

Write a software program that calculates the average and standard deviation of all input argument values.

- You may assume that all input arguments are numeric values and use `atof()` function to convert the argument into a `double` type variable. No error handling for malformed argument is needed.
- However, if no input argument was given, you need to give error message and terminate the program, returning -1 in the `main()` function, instead of returning 0 for non-empty arguments.
- For average, use $\overline{x} = \sum_{i=1}^{n} x_i/n$ for input argument $x_1, \cdots, x_n$.
- For standard deviation of the sample, use $\sigma_x = \sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2/n}$
- You may use `sqrt()` function (with `#include <cmath>`).

Example runs of valid input arguments are

```
user@host:~/Private/biostat615/hw1$ ./hw-1-3 1 2 3 4 5
Average : 3
Standard Deviation 1.41421
user@host:~/Private/biostat615/hw1$ ./hw-1-3 1 2 3 4 5.5
Average : 3.1
Standard Deviation 1.56205
```

In your google document, include full source code and example output. Your source code name must be hw-1-3.cpp. Do not include any other files in your .tar.gz submission.