ADVANCED MATHEMATICAL PROGRAMMING

MTH399 / U15161 – Level 3

UNIVERSITY OF PORTSMOUTH

Spring Semester Coursework, Part I

Academic Session: 2010–2011

INSTRUCTIONS

a) **Deadline: Thursday, April 7, 2010 (before 3:30PM) at CAM office**

b) This assignment makes up **40% of the unit assessment**.

c) Answer **both** Exercises, 1 and 2.

d) Submit a hardcopy printout of all materials, along with a hardcopy of this file.

e) Upload all code (`.h` and `.cpp` files) to the Victory assignment.

f) Make sure to label all materials with your reference number.

g) Explain your solutions using comments in the code.

h) If you cannot complete a question, show what you attempted to do.

i) You must conform to the C++ standards used in the unit MTH399.

j) It is **not** permitted to do assignments in practical sessions unless said sessions are explicitly devoted to such purpose.

k) Assignment must be undertaken **alone**.

STUDENT ID NUMBER: ....

Exercise 1. The purpose of this exercise is to compute the following quantities:

$$\binom{n}{k_1,\, k_2,\, \ldots,\, k_m} := \frac{n!}{k_1!\, k_2! \,\cdots\, k_m!}, \tag{1}$$

for any given $n \in \mathbb{N}$ and $k_1, k_2, \ldots, k_m \in \mathbb{N} \cup \{0\}$ such that $k_1 + k_2 + \cdots + k_m = n$.

Your function `long long int coef ( ... )` will have at least one argument: vector $\boldsymbol{k} = (k_1, \ldots, k_m)$, *its space in memory dynamically allocated.* You are free to decide whether you want n and the length $m \geq 1$ of vector $\boldsymbol{k}$ as further inputs to your function.

Your project must contain functions testing the correctness of `coef` by means of known properties, examples and boundary conditions. Some useful properties:

$$\binom{n}{k_1,\, k_2,\, \ldots,\, k_m} = \binom{k_1}{k_1}\binom{k_1+k_2}{k_2}\cdots\binom{k_1+\cdots+k_m}{k_m} = \prod_{i=1}^{m}\binom{\sum_{j=1}^{i} k_j}{k_i}, \tag{2}$$

$$\binom{n}{k_1,\, k_2,\, \ldots,\, k_m} = \binom{n}{k_1+k_2,\, \ldots,\, k_m}\binom{k_1+k_2}{k_1,\, k_2} \tag{3}$$

$$= \binom{n}{k_1+k_2+k_3,\, k_4,\, \ldots,\, k_m}\binom{k_1+k_2+k_3}{k_1,\, k_2,\, k_3} = \ldots, \tag{4}$$

$$\binom{n}{\sum_{i=1}^{n} k_i,\, 0\, \ldots,\, 0} = \binom{n}{0, \sum_{i=1}^{n} k_i,\, 0\, \ldots,\, 0} = \cdots = \binom{n}{0,\, \ldots,\, 0,\, \sum_{i=1}^{n} k_i}, \tag{5}$$

Needless to say, the binomial coefficients involved in (2) can be computed using `coef` as well and need no additional function for their definition. But you are free to create such an additional function.

Comments:

a) Make sure your project comprises separate header (`*.h`) and `.cpp` files.

b) If n and m are included as explicit arguments of `coef`, make sure your function returns error messages whenever necessary.

c) In the definition of (1), you will need to use a function *implemented by you*: `factorial`. You do not need to check any properties for it, but are more than welcome to do so if you already have them.

d) Make sure you keep track of each property that is being checked, be it by means of exception throwing or by writing down on an output file passed by reference as an input of your test functions – *do not* declare the `ofstream` file variable as global.

e) Think of ways in which to effectively test any of (2)–(5) (or any other property) for finite collections of integers m, n and integer vectors $(k_1, k_2, \ldots, k_m)$. Test the correctness of your implementation with deliberate wrong assertions.

Exercise 2. Given an arbitrary finite closed interval $[a, b] \subset \mathbb{R}$ $(a < b)$ and an integrable function $f : [a, b] \to \mathbb{R}$, consider the following approximation of the definite integral of f:

$$\int_a^b f(x)\,dx \simeq Q_N(f, a, b, N) := \frac{h}{6}\left[f(x_0) + f(x_N) + 4\sum_{i=1}^{N} f(y_i) + 2\sum_{i=1}^{N-1} f(x_i)\right] \qquad (6)$$

$x_0, \ldots, x_N$ being the nodes of a partition of $[a, b]$ into N subintervals of equal length h, and $y_1, \ldots, y_N$ being the respective middle points of said subintervals. Needless to say, $x_0 = a$ and $x_N = b$.

The purpose of this exercise is to approximate the definite integral of any given continuous function f by $Q_N(f, a, b, N)$ for different values of N. The prototype of your function will be

```
double numint( double a, double b, int N, (...) );
```

where `a` and `b` are the extremes of the definite integration interval, `N` is the number of subintervals and `f` is to be defined on a separate function. `res`, passed through reference, must contain the value of $Q_N(f, a, b, N)$ when `numint` returns to the function calling it.

Steps: Given a function f *whose primitive F you can compute exactly*, and a maximal number of subintervals $N_{\max}$,

a) your program will compare $Q_N(f, a, b, N)$ and $\int_a^b f(x)\,dx = F(b) - F(a)$ for increasing values of N. Each of these values $r_N := \left|Q_N(f, a, b, N) - \int_a^b f(x)\,dx\right|$, $N = 2, 3, \ldots, N_{\max}$, will be stored in an output file having extension `.txt`.

b) Upon scrutiny of the given `*.txt` file and the evolution of r_N for increasing N, what are your conclusions? Does r_N steadily decrease for increasing values of N if $N_{\max}$ is very large? Justify your answer and summarize your conclusions in a short informal paragraph. Make sure to try different intervals $[a, b]$, different functions f and different values of $N_{\max}$ before answering.

Comments:

a) Your project should comprise *at least*:
   - two header files;
   - three files having extension `.cpp`.

b) `numint` and/or the function immediately calling it will return 0 if numerical integration was possible, and 1 otherwise.

c) You will need to define another function, with a prototype to the effect of

```
double F ( double x );
```

for the exact primitive of the original function, having prototype

```
double f ( double x );
```

d) You are free to try more than one function f (meaning more than one `.txt` output file), but make sure the corresponding primitive F is correct.