MCS 256 Discrete Calculus and Probability
Set 4: Summation Calculus

Homework rules

- Acknowledge your sources (people and texts).
- In nontrivial problems, show *how* you get your answers.
- Turn in neat, well-written solutions, not messy first drafts. Trim "fringes."
- Do not copy collaborative solutions; write up solutions in your own words.
- Turn in homework on time. Each class day late reduces the possible points by 25%.

"The whole is more than the sum of its parts." —Aristotle [Metaphysica 10f-104]

Problems

(1) (a) Make a table showing the first few values of $\sum_{k=1}^{n} k^2$ for $n = 1, 2, 3, \ldots$.

(b) Form a divided difference table and use it to figure out the Newton interpolating polynomial $p_3(n)$ for $f(n) := \sum_{k=1}^{n} k^2$.

(c) By mathematical induction, prove that $\sum_{k=1}^{n} k^2 = p_3(n)$ for $n = 1, 2, 3, \ldots$.

(2) The purpose of this problem is to show you parallels and contrasts between integration and summation. Evaluate the following definite integrals and sums. Assume that a and b are integers with $a < b$ and n is a positive integer. Compile your answers in a table on one page, and attach your readable supporting work on a separate page.

(1)	$\int_a^b f'(x)\,dx$	$\sum_{a \le k < b} \Delta f(k)$
(2)	$\int_a^b e^x\,dx$	$\sum_{a \le k < b} 2^k$
(3)	$\int_a^b 2^x\,dx$	$\sum_{a \le k < b} e^k$
(4)	$\int_0^n x^4\,dx$	$\sum_{0 \le k < n} k^{\underline{4}}$
(5)	$\int_0^n x^{\underline{4}}\,dx$	$\sum_{0 \le k < n} k^4$

(3) (a) Give a simple formula for $\Delta\binom{x}{k}$ that is valid for each fixed integer k and every real value of the variable x.

(b) Using the Fundamental Theorem of Summation Calculus, determine a simple formula for $\sum_{j=m}^{n} \binom{j}{m}$ valid for integers $0 \le m \le n$. (Alternatively, you could discern the correct answer and prove it by mathematical induction.)

(4) In the following pseudocode, assume that it takes k "steps" to do task(i, j, k) and that n is a positive integer.

```
| For i from 1 to n do
|       For j from 1 to i do
|             For k from 1 to j do
|                   task(i, j, k)
|             end do
|       end do
| end do.
```

Determine, as a function of n, the total number of steps it takes to execute this pseudocode. (If you represent the answer in terms of nested summations, you are advised to evaluate your sums from the inside out.)