C PROGRAMMING: SUMMING ARITHMETIC SERIES

An arithmetic series is a collection of integers starting with one integer A and then increasing by the same, second, integer B each time, *ie*

$$A, \quad A+B, \quad A+2B, \quad A+3B, \quad A+4B, \quad \cdots$$

The most simple example has $A = 1$ and $B = 1$. The series enumerates the positive integers:

$$1, \quad 2, \quad 3, \quad 4, \quad 5, \quad 6, \quad \cdots$$

We first write a program to print the first N members of an arithmetic series.

```
/* this is a program to print an arithmetic series*/
#include <stdio.h>
#include <math.h>
int main(void)
{
int sum=0;
int A,B,j,N;
printf("Enter A");
printf("\n");
scanf("%i",&A);
printf("Enter B");
printf("\n");
scanf("%i",&B);
printf("Enter N");
printf("\n");
scanf("%i",&N);
for (j=0; j<N; j=j+1)
{
printf("%i \n",A+j*B);
}
return 0;
}
```

Comments:

[1] There is not too much new going on here as far as C is concerned. We are once again declaring variables, printing out prompts to the screen, reading numbers in, using a simple loop, and writing things out. We will now set ourselves the task of *summing* up the first N elements of an arithmetic series.

[2] But first let's discuss the answer we should get. Let's only do the simple case $A = B = 1$. Does anyone know how to do it?

[3] There are many approaches. One is to notice that if we add up the first and last elements of the series we get $N + 1$. Likewise if we pair up the second and second to last we get $2 + (N - 1) = N + 1$. There are $N/2$ such pairs, so the sum is $(N + 1)N/2$.

[4] Another method is geometric. Consider a collection of squares of area one. Make a first stack of a single square, then a second, adjacent, stack of two squares, a third of three squares, and so on. You form a triangle with a jagged hypotenuse. The height $h = N$ and the base $b = N$. Ignoring the jags, the area is $bh/2 = N^2/2$. The jags conntribute an extra $N/2$, so the total area is $N^2/2 + N/2 = (N + 1)N/2$.

[5] A third, and very sophisticated, method is to use mathematical induction. (The method of mathematical induction is one of the 'Peano postulates' upon which all of the mathematics of the natural numbers can be built.) Suppose you have some statement which involves an integer N. You want to prove it is true for all N. The postulate (which is sort of obvious) is this: If you can prove the statement for $N = 1$, and if you can prove that its truth for N implies its truth for $N + 1$, then it is true for all N. (Think about the logic behind why this works.)

Here our statement is that the sum of the first N natural numbers is $(N + 1)N/2$. Is it true for $N = 1$? Clearly so! $1 = (1 + 1)1/2$. The beauty (power) of mathematical induction is that we now get to *assume* truth for N! We try to use this assumption to demonstrate truth for $N + 1$. Here goes:

$$1 + 2 + 3 + \cdots N + N + 1 = (N + 1)N/2 + N + 1 = (N + 1)(N + 2)/2$$

In writing the first equality we used the statement for N. In writing the second equality we used some simple algebra. The last step to notice is that these two steps have proven the statement for $N+1$. That is, connecting the first and last expressions is precisely the statement we desired to prove for $N+1$.

[**6**] Exercise for you: Can you get the general formula when A and B are not necessarily equal to one?

[**7**] (Hard) exercise for you: Can you work out the sum of the *squares* of the first N integers?

Our final exercise is to take our program which writes the arithmetic series $1, 2, 3, \cdots$ and actually sum it up.

Remember: Do **not** monkey around with your working program! Make a copy and modify that.

```
/* this is a program to add the first N integers*/
#include <stdio.h>
#include <math.h>
int main(void)
{
int sum=0;
int j, N;
printf("Enter N");
printf("\n");
scanf("%i",&N);
printf("   j          sum   ");
for (j=0; j<N; j=j+1)
{
sum=sum+j;
printf("\n   %i",j);
printf("  ");
printf("%20i",sum);
}
printf("\n");
return 0;
}
```