Filling an array

Write a program defining an array **nums** that will store 100 numbers initialized like below:

C:
```
int nums[100] = {1, 3, 5};
```

Although if the array can store 100 numbers, only the first three values have been initialized. Your program must generates the remaining values integers according to the next formula:

$$nums[i] = nums[0] + nums[1] + \ldots + nums[i - 1]$$

for example, for $i = 3$

$$nums[3] = nums[0] + nums[1] + nums[2] = 1 + 3 + 5 = 9$$

After filling every array element, print it.

Reverse Order (P20L)

Write a program that reads integers from the user and stores them in a list. Use 0 as a sentinel value to mark the end of the input. Once all of the values have been read your program should display them (except for the 0) in reverse order, with one value appearing on each line.

Random matrix multiplication

Create a program that generates two matrixes of m×n and i×j, fill them with random numbers and then performs the multiplication. The values of m, n, i and j must be entered by the user. Displays both matrixes and the result. Check for matrix conformity.

Sorted Order (PS21L)

Write a program that reads integers from the user and stores them in a list. Your program should continue reading values until the user enters 0. Then it should display all of the values entered by the user (except for the 0) in order from smallest to largest, with one value appearing on each line. Use either the sort method or the sorted function to sort the list.

Avoiding Duplicates (PS21L)

In this exercise, you will create a program that reads numbers from the user until the user enters a blank line. After the user enters a blank line your program should display each number entered by the user exactly once and the number of duplicates. The numbers should be displayed in the same order that they were entered. For example, if the user enters:

1
2
1
3
1

then your program should display:

1
2
3
duplicates removed: 2

Negatives, Zeros and Positives (PS38L)

Create a program that reads integers from the user until a blank line is entered. Once all of the integers have been read your program should display all of the negative numbers, followed by all of the zeros, followed by all of the positive numbers. Within each group the numbers should be displayed in the same order that they were entered by the user. For example, if the user enters the values 3, -4, 1, 0, -1, 0, and -2 then your program should output the values -4, -1, -2, 0, 0, 3, and 1. Your program should display each value on its own line.