Factorial of a large number

How to compute factorial of 100 using a C/C++ program?
Factorial of 100 has 158 digits. It is not possible to store these many digits even if we use long long int. Following is a simple solution where we use an array to store individual digits of the result. The idea is to use basic mathematics for multiplication.

The following is detailed algorithm for finding factorial.

factorial(n)
1) Create an array 'res[]' of MAX size where MAX is number of maximum digits in output.
2) Initialize value stored in 'res[]' as 1 and initialize 'res_size' (size of 'res[]') as 1.
3) Do following for all numbers from x = 2 to n.
……a) Multiply x with res[] and update res[] and res_size to store the multiplication result.

How to multiply a number 'x' with the number stored in res[]?
The idea is to use simple school mathematics. We one by one multiply x with every digit of res[]. The important point to note here is digits are multiplied from rightmost digit to leftmost digit. If we store digits in same order in res[], then it becomes difficult to update res[] without extra space. That is why res[] is maintained in reverse way, i.e., digits from right to left are stored.

multiply(res[], x)
1) Initialize carry as 0.
2) Do following for i = 0 to res_size – 1
….a) Find value of res[i] * x + carry. Let this value be prod.
….b) Update res[i] by storing last digit of prod in it.
….c) Update carry by storing remaining digits in carry.
3) Put all digits of carry in res[] and increase res_size by number of digits in carry.

Example to show working of multiply(res[], x)

```
A number 5189 is stored in res[] as following.
res[] = {9, 8, 1, 5}
x = 10

Initialize carry = 0;

i = 0, prod = res[0]*x + carry = 9*10 + 0 = 90.
res[0] = 0, carry = 9

i = 1, prod = res[1]*x + carry = 8*10 + 9 = 89
res[1] = 9, carry = 8

i = 2, prod = res[2]*x + carry = 1*10 + 8 = 18
res[2] = 8, carry = 1

i = 3, prod = res[3]*x + carry = 5*10 + 1 = 51
res[3] = 1, carry = 5

res[4] = carry = 5

res[] = {0, 9, 8, 1, 5}
```

Below is C++ implementation of above algorithm.

```
// C++ program to compute factorial of big numbers
#include<iostream>
using namespace std;

// Maximum number of digits in output
#define MAX 500

int multiply(int x, int res[], int res_size)

// This function finds factorial of large numbers and pr
void factorial(int n)
{
    int res[MAX];

    // Initialize result
```

```
    res[0] = 1;
    int res_size = 1;

    // Apply simple factorial formula n! = 1 * 2 * 3 * 4
    for (int x=2; x<=n; x++)
        res_size = multiply(x, res, res_size);

    cout << "Factorial of given number is \n";
    for (int i=res_size-1; i>=0; i--)
        cout << res[i];
}

// This function multiplies x with the number represente
// res_size is size of res[] or number of digits in the
// by res[]. This function uses simple school mathematic
// This function may value of res_size and returns the n
int multiply(int x, int res[], int res_size)
{
    int carry = 0;  // Initialize carry

    // One by one multiply n with individual digits of r
    for (int i=0; i<res_size; i++)
    {
        int prod = res[i] * x + carry;
        res[i] = prod % 10;  // Store last digit of 'pro
        carry  = prod/10;    // Put rest in carry
    }

    // Put carry in res and increase result size
    while (carry)
    {
        res[res_size] = carry%10;
        carry = carry/10;
        res_size++;
    }
    return res_size;
}

// Driver program
int main()
{
    factorial(100);
    return 0;
}
```

Output:

```
Factorial of given number is
93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000
```

The above approach can be optimized in many ways. We will soon be discussing optimized solution for same.