

Fibonacci Reloaded

Marcus Birkenkrahe (pledged)

May 4, 2024

Problem

The program below generates the first 10 Fibonacci numbers:

```
int f[10];    // declare array of 10 elements
f[0] = 1;     // initialize first element
f[1] = 1;     // initialize second element

for (int i = 2; i < 10; i++) {
    f[i] = f[i-2] + f[i-1];
}

for (int i = 0; i < 10; i++) printf("%d ", f[i]);
```

Your tasks:

1. Generalize the program so that the user is asked to input the length N of the series, and is given the first N Fibonacci numbers.
2. Print the first N=30 Fibonacci numbers as a 6 x 5 matrix:

```
fib[ 0] = 0    ... fib[23] = 28657
fib[ 1] = 1    ... fib[24] = 46368
fib[ 2] = 1    ... fib[25] = 75025
fib[ 3] = 2    ... fib[26] = 121393
fib[ 4] = 3    ... fib[27] = 196418
fib[ 5] = 5    ... fib[28] = 317811
fib[ 6] = 8    ... fib[29] = 514229
```

Submit your solutions in an Org-mode file with the usual title and author meta data. For full points, document your code and present code that compiles and produces the desired output.

Solution:

Generalize and set N first in the program using `#define`, then add input routine with `scanf`.

There are two solutions to this:

1. Compute Fibonacci series as 1d array and print it as matrix (this is what you are supposed to do).
2. Compute Fibonacci series as 2d array, then print it (bonus solution included below).

Compute series as 1d array and print as matrix.

- Pseudocode:

```
// set series upper bound N
// declare and initialize Fibonacci series as 1d array
// compute terms 2..N
// print series as 2d matrix of 6 rows and 5 columns
// iterate over series
// print series element
// if index is multiple of 5 go to next line
```

- Code for column-wise printing:

```
#include <stdio.h>

int main() {
    int N = 30; // The number of Fibonacci numbers to compute
    long long fib[N]; // Array to store Fibonacci numbers

    // Initial conditions for the Fibonacci sequence
    fib[0] = 0;
    fib[1] = 1;

    // Compute Fibonacci numbers
    for (int i = 2; i < N; i++) {
        fib[i] = fib[i - 1] + fib[i - 2];
    }
}
```

```

// Print the Fibonacci numbers in a 6 x 5 matrix format
for (int i = 0; i < N; i++) {
    printf("fib[%2d] = %-8lld", i, fib[i]);
    if ((i + 1) % 5 == 0) {
        printf("\n");
    }
}

return 0;
}

```

- Code for row-wise printing:

```

#include <stdio.h>

int main() {
    int N = 30; // The number of Fibonacci numbers to compute
    int fib[N]; // Array to store Fibonacci numbers

    // Initial conditions for the Fibonacci sequence
    fib[0] = 0;
    fib[1] = 1;

    // Compute Fibonacci numbers
    for (int i = 2; i < N; i++) {
        fib[i] = fib[i - 1] + fib[i - 2];
    }

    // Print the Fibonacci numbers in a 6 x 5 matrix format column-wise
    int rows = 6;
    int cols = 5;
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            int index = j * rows + i;
            if (index < N) {
                printf("fib[%2d] = %-8d", index, fib[index]);
            }
        }
        printf("\n");
    }
}

```

```

        return 0;
    }

```

Compute Fibonacci series as 2d array, then print it

```

#include <stdio.h>

int main() {
    int rows = 6;
    int cols = 5;
    long long fib[rows][cols]; // 2D array to store Fibonacci numbers

    // Initial conditions for the Fibonacci sequence
    if (rows * cols >= 1) fib[0][0] = 0;
    if (rows * cols >= 2) fib[1][0] = 1;

    // Compute Fibonacci numbers and populate the 2D array column-wise
    for (int j = 0; j < cols; j++) {
        for (int i = 0; i < rows; i++) {
            int index = j * rows + i;
            if (index >= 2) {
                int prev1_x = (index - 1) % rows;
                int prev1_y = (index - 1) / rows;
                int prev2_x = (index - 2) % rows;
                int prev2_y = (index - 2) / rows;
                fib[i][j] = fib[prev1_x][prev1_y] + fib[prev2_x][prev2_y];
            }
            if (index == 29) break; // Stop after the 30th number
        }
    }

    // Print the Fibonacci numbers stored in a 2D array, column-wise
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("fib[%2d] = %-8lld", j * rows + i, fib[i][j]);
        }
        printf("\n");
    }
}

```

```
    return 0;  
}
```