

نقطه شروع می‌کنیم از تابع اصلی. در خط اول دستور `printf`، قرار است $f(1)$ توسط $1/2$ چاپ بشود. بابت n به فراخوانی تابع f ، باید $f(1)$ را بدست آوریم. تابع f تابعی بازگشتی است که حالت پایه آن 1 است و $f(1) = 1$ بابت n به شرط خط اول تابع. حالا n یک عدد صحیح با مقدار 1 تعریف می‌شود.

حالا یک حلقه داریم با دستور `for`. در اولین دفعه اجرای حلقه، $i = 1$ و می‌دانیم هر بار باید $n + f(n-1) * f(i)$ را در n پرریم. اولین دفعه $n = 1$ و مقدار $n + f(n-1) * f(i)$ برابر است با $1 + f(0) * f(1)$ که برابر است با 1 . در نتیجه پس از اولین دفعه اجرای حلقه، مقدار n برابر با 1 می‌شود و بابت n به اینکه تابع به حالت پایه است فراخوانده شده، حلقه فقط یکبار اجرا می‌شود و در نهایت از حلقه خارج می‌شود و n بازگر دانه می‌شود به عنوان مقدار $f(n)$. پس می‌دانیم $f(2) = 2$ و در نتیجه در تابع اصلی، $f(2)$ که برابر است با 2 چاپ می‌شود.

در نتیجه خروجی برنامه برابر است با $\leftarrow 2$

و در نهایت با توجه به دستور `"return 0"`، برنامه بسته می‌شود.

در واقع اتفاقی که در تابع f که ورودی از جنس `int` دارد و خروجی از جنس `int` می‌دهد برای $f(2)$ ابتدا شرط حلقه می‌شود و بابت n به اینکه $n \neq 1$ ، از این شرط رد می‌شویم. سپس متغیر n از نوع `int` به مقدار 1 تعریف می‌شود. سپس در حلقه ابتدا $i = 1$ در نتیجه $n = n + f(1) * f(1)$ که با توجه به اینکه مقدار 1 به تابع پاس می‌شود، و در شرط اول تابع صدق می‌کند؛ مقدار 1 ، `return` می‌شود و به فریم قبلی برمی‌گردیم. و با انجام دوباره این عملیات مقدار n برابر با 2 می‌شود و از حلقه بیرون می‌آئیم و مقدار 2 به عنوان مقدار $f(2)$ از تابع `return` می‌شود.