

در این شبه کد، تابعی تعریف شده که ورودی هاش، دو متغیر صحیح هستند. در اولین اجرا، تابع با ورودی های $sum=0$ و n که احتمالاً ورودی کاربره، کارش رو شروع میکنه، که ما به عنوان مثال، n رو برابر 3107 در نظر میگیریم. ابتدا، چک میشه که آیا n ورودی، یعنی همون 3017، نابرابر با صفر باشه، که میدونیم اینطور هست و تابع، دستورات درون if رو اجرا میکنه؛ یعنی ابتدا باقیمونده تقسیم 3107 بر 10 رو که برابر 7 عه در متغیر m میریزه و سپس، حاصل جمع m و ده برابر مقدار قبلی sum رو در sum میریزه، پس در اینجا sum میشه 7؛ و دوباره تابع برای مقدار $n/10$ و مقدار جدید sum اجرا میشه؛ البته باید توجه داشته باشیم که چون هم 3107 و هم 10، اعدادی صحیح هستند، حاصل تقسیمشون هم یه عدد صحیح میشه، که در واقع خارج قسمت تقسیم 3107 بر 10 عه، یعنی 310.

این بار هم، با روندی مشابه دفعه قبل، مقدار m به 0 تغییر میکنه و سپس، 10 برابر sum قبلی، یعنی 70 با m جمع میشه و در خود sum ریخته میشه، و مقدار جدید sum میشه 70. و دوباره تابع برای مقادیر 31 و 70 اجرا میشه.

در دفعه سوم هم، مقدار m به 1 و مقدار sum به 701 تغییر میکنه تا تابع یکبار دیگه با مقادیر 3 و 701 اجرا شه؛ که در اون اجرا هم مقدار m و sum به ترتیب به 3 و 7013 تغییر میکنه.

با رسیدن کار به اینجا، دوباره نوبت این شده که تابع یه بار دیگه با مقدار 0 و $sum=7103$ اجرا شه (چون خارج قسمت تقسیم 3 بر 10 برابر صفره)؛ اما این بار دیگه n ورودی به تابع، شرط نامساوی بودن با صفر رو نداره! پس محاسبات تابع دیگه تموم شده و وقتشه که مقدار sum رو به عنوان حاصل نهایی تابع واسه سیستم عامل بفرسته!

اما 7013 چیه؟! خب اگه دقت کنی متوجه میشی که این عدد، همون 3107 عه که برعکس نوشته شده! و اگه بررسی کنی، متوجه میشی که خروجی تابع برای هر n ورودی، برعکس شده همون n عه!

البته اگه یه کم دقت کنی، متوجه میشی که این الگوریتم خیلیم کامل نیست و واسه ورودی هایی که آخرین ارقامشون، صفر باشه، خروجی مطلوبی تولید نمیکنه؛ یعنی مثلاً ورودی همه عدد های 12 و 120 و 1200 و... میشه 21؛ دلیلش هم اینه که 0 رو هر چند بار که در 10 ضرب کنی، بازم صفره و هر چقدرم که در سمت راست عددت صفر داشته باشی، وقتی ازشون میگذری باز هم مقدار sum ت صفره و یه چورایی انگار کان لم یکن شئی!

واسه حل این مشکل، میشه قبل از دستورات تعریف شده در تابع، حلقه ای تعریف کرد که با هر بار اجرا شدن، شرط $n \bmod 10 = 0$ رو بررسی کنه، و تا زمانی که باقیمونده تقسیم n بر 10 برابر صفره، یکی به متغیر شمارشگری مثلاً i که مقدار اولیه اش پیش از حلقه صفر بوده، اضافه کنه! در انتها هم قبل از برگردوندن sum نهایی به عنوان حاصل، باید حلقه ای تعریف بشه که به تعداد i ، sum رو ضرب در 10 کنه؛ به همین سادگی و خوشمزگی!