تابع Plusone : تابعی است که عددی از جنس string را دریافت میکند و به آن یک واحد اضافه میکند و حاصل را بصورت string برمیگرداند

تابع Substract : تابعی است که دو عدد (که عدد دارای رشته اول، از عدد دارای رشته دوم بزرگتر است) را دریافت و این دو عدد را تفریق می کند (در درون تابع دو ورودی به کمک تابع ToArr() به آرایه تبدیل شده و سپس ارقام شان را برابری کنیم و سپس عملیات را انجام می دهیم) و حاصل را به string تبدیل می کند و برمی گرداند

تابع IsBigger : دو عدد را بصورت string دریافت می کند، به آرایه تبدیل می کند و دو عدد را رقم به رقم مقایسه می کند. اگر عدد ورودی اول از ورودی دوم بزرگتر یا مساوی باشد، تابع مقدار True را برمی گرداند؛ درغیراینصورت False برمی گردد

تابع DivideRemainder : دو عدد را بصورت string دریافت می کند و به کمک یک حلقه while باررش تفریق های متوالی باقی مانده تقسیم ورودی اول بر ورودی دوم را برمی گرداند

تابع Divident : مانند تابع DivideRemainder عمل می کند، با این تفاوت که در اینجا یک شمارنده وجود دارد که تعداد تفریق ها را می شمارد و آن را برمی گرداند (تعداد تفریق ها همان خارج قسمت است)

تابع IsZero : یک عدد را به شکل string دریافت می کند و چک می کند که این عدد صفر است یا خیر

تابع IsPrime: بخش اصلی برنامه می باشد که به کمک توابع قبلی و با یک حلقه while از عدد ۲ تا $\frac{n}{2}$ (n=عددی) را بر n تقسیم کرده و هر باقی که باقی مانده تقسیم برابر صفر شود مقدار True را برای گردانند کند یعنی آن عدد، اول است و اگر تا آخر حلقه، باقی مانده هیچ تقسیمی صفر نشود، آن عدد اول نیست. قبل از تمام این مراحل، در این تابع چک می شود که عدد ورودی بر ۲ یا ۳ یا ۵ یا ۶ یا ۸ یا ۹ یا ۱۰ بخش پذیر نباشد؛ اگر بخش پذیر نبود، وارد بخش بعدی تابع می شود

تابع SayPrime: اگر مقدار true برنامه باشد عبارت "Prime" و اگر false باشد، عبارت "Not Prime" را برای گرداند

** کدهای نوشته شده داخل تابع Main() در ابتدا روش ورود اعداد توسط کاربر را در برنامه می گذارد و اعداد را داخل آرایه input می ریزند.
سپس با کمک توابع IsPrime و SayPrime، آرایه output را پر می کنیم. سپس با آرایه result بین اعداد یک خط فاصله می اندازیم و آن ها را خوانا تر می کنیم و در نهایت خروجی را در پنل کنسول نمایش می دهیم. **

۲- من برای بازهٔ مقسوم‌علیه از $\frac{n}{2}$ استفاده کردم چون $\sqrt{n}$ به سختی و با الگوریتم‌های پیچیده بدست می‌آمد

۳- $O(n^2)$ (چون هیچکدام از حلقه‌ها از عدد مجزا نیستند و حلقه‌ها از طریق توابع به یکدیگر وصل‌اند و تشکیل چندین حلقهٔ تو در تو را می‌دهند)

* این یک الگوریتم رام نشدنی می‌باشد

۴- برای اعداد ۱ تا ۳ رقمی زیر ۱ ثانیه، برای اعداد ۴ تا ۷ رقمی زیر ۳۰ ثانیه، اعداد ۷ الی ۱۰ رقمی بالای ۳ دقیقه و برای اعداد بزرگتر از ۱۰ رقم تایمی بسیار طولانی را نیاز خواهیم داشت

شاهین زمانی

۹۹۱۲۱۰۳۰