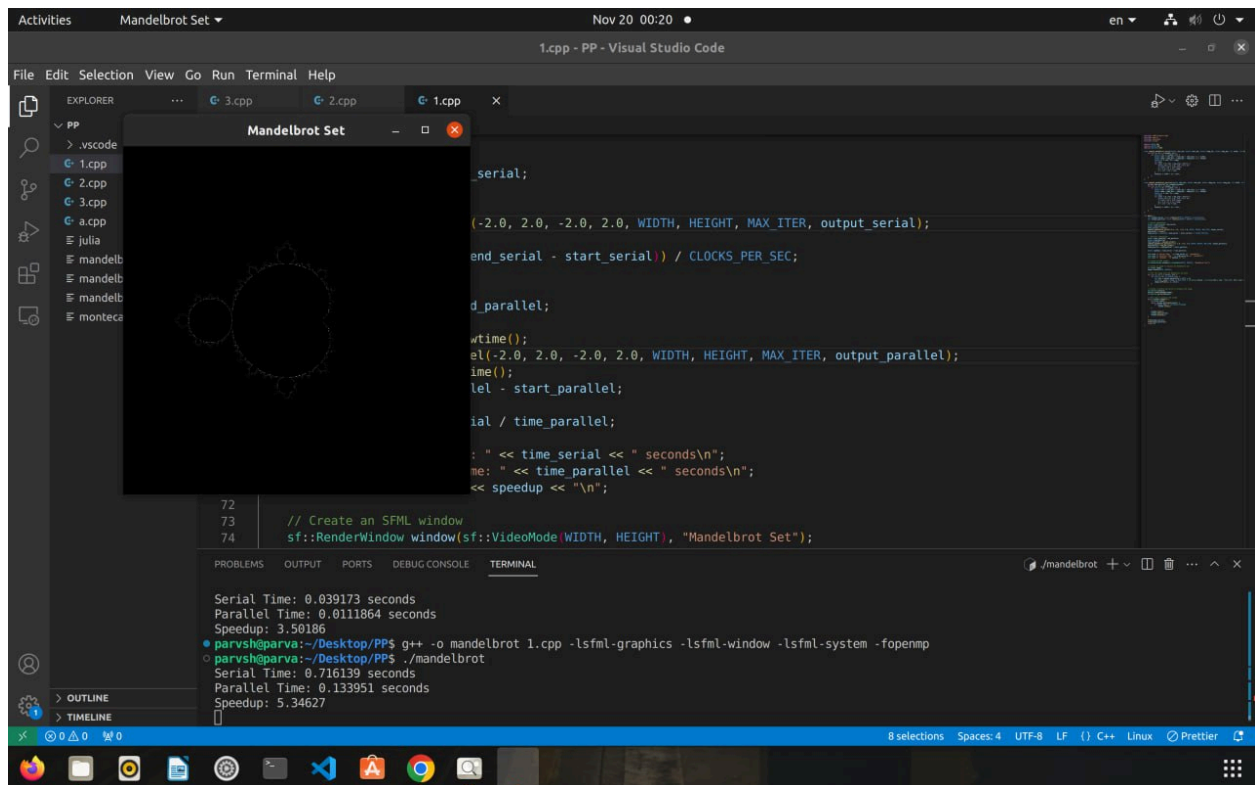


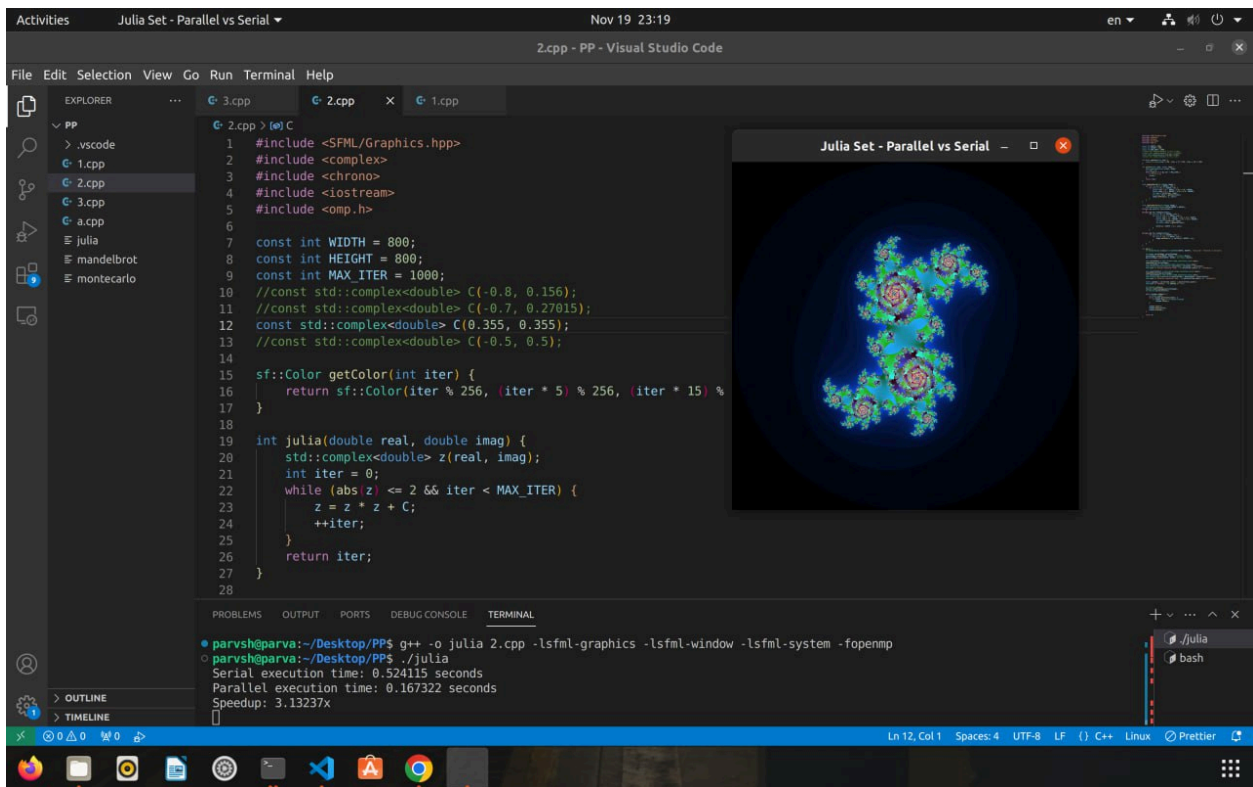
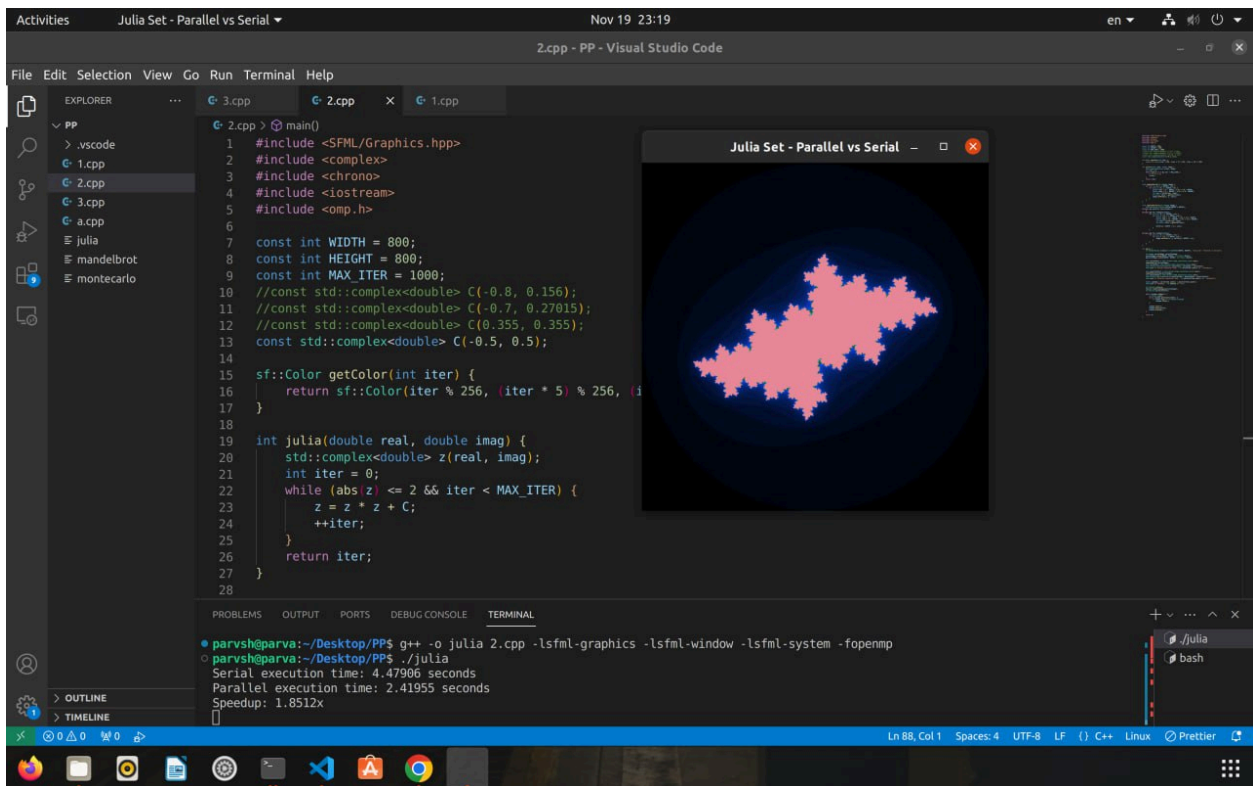
گلبو رشیدی 810100148

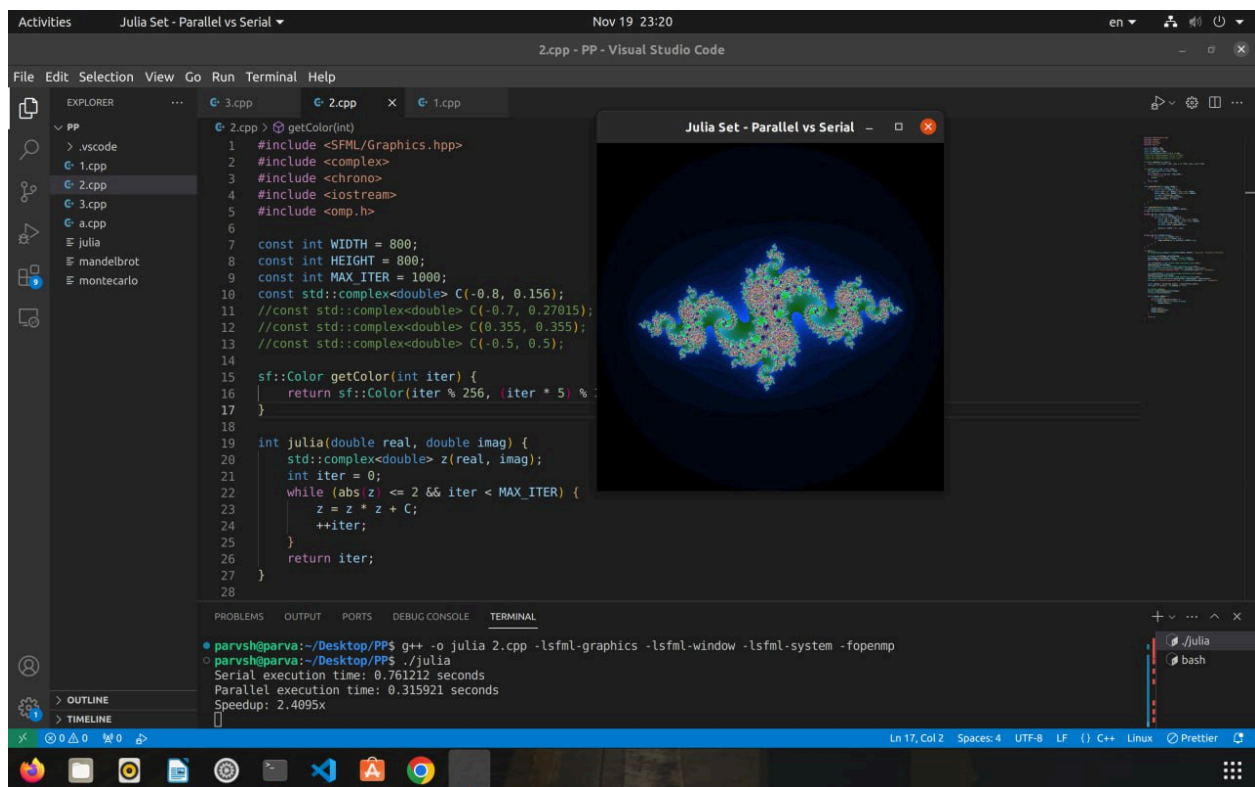
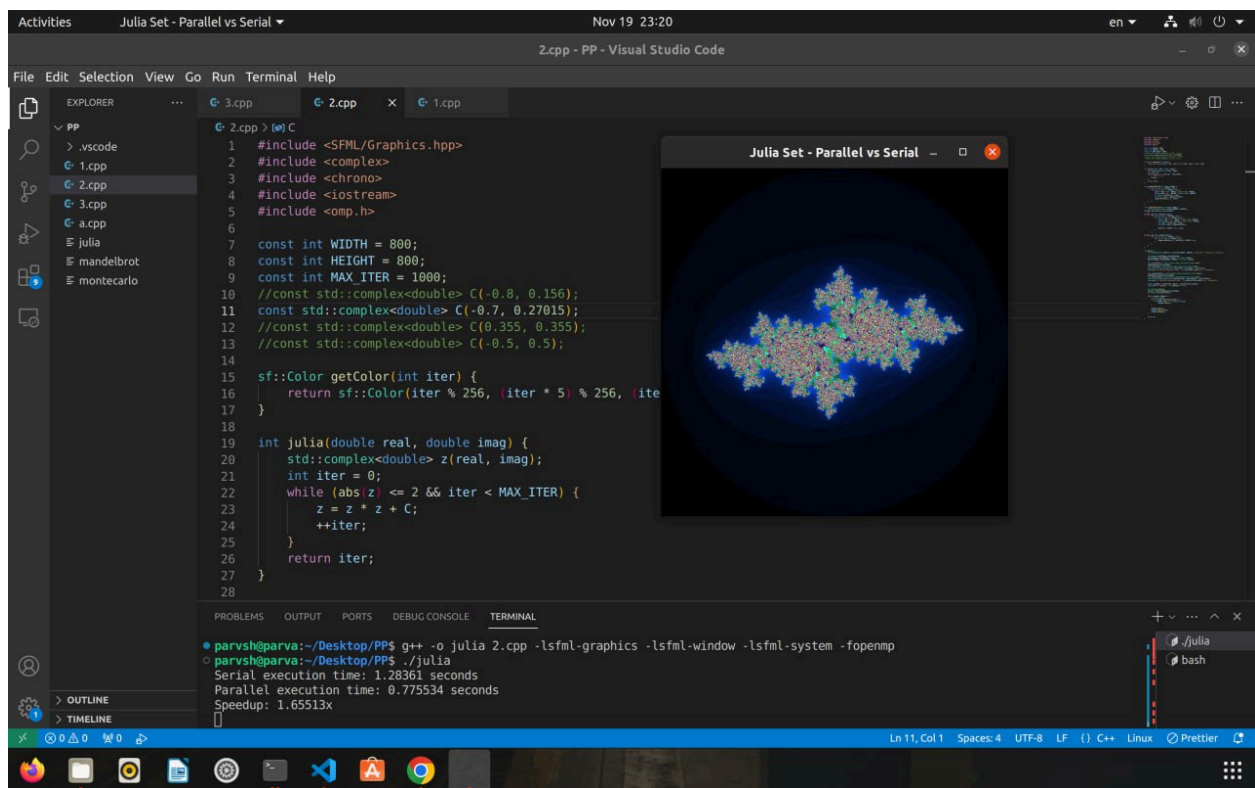
پروا شریفی 810100171

1. Mandelbrot set

```
g++ -o mandelbrot 1.cpp -fopenmp -lsfml-graphics -lsfml-window -lsfml-system
```







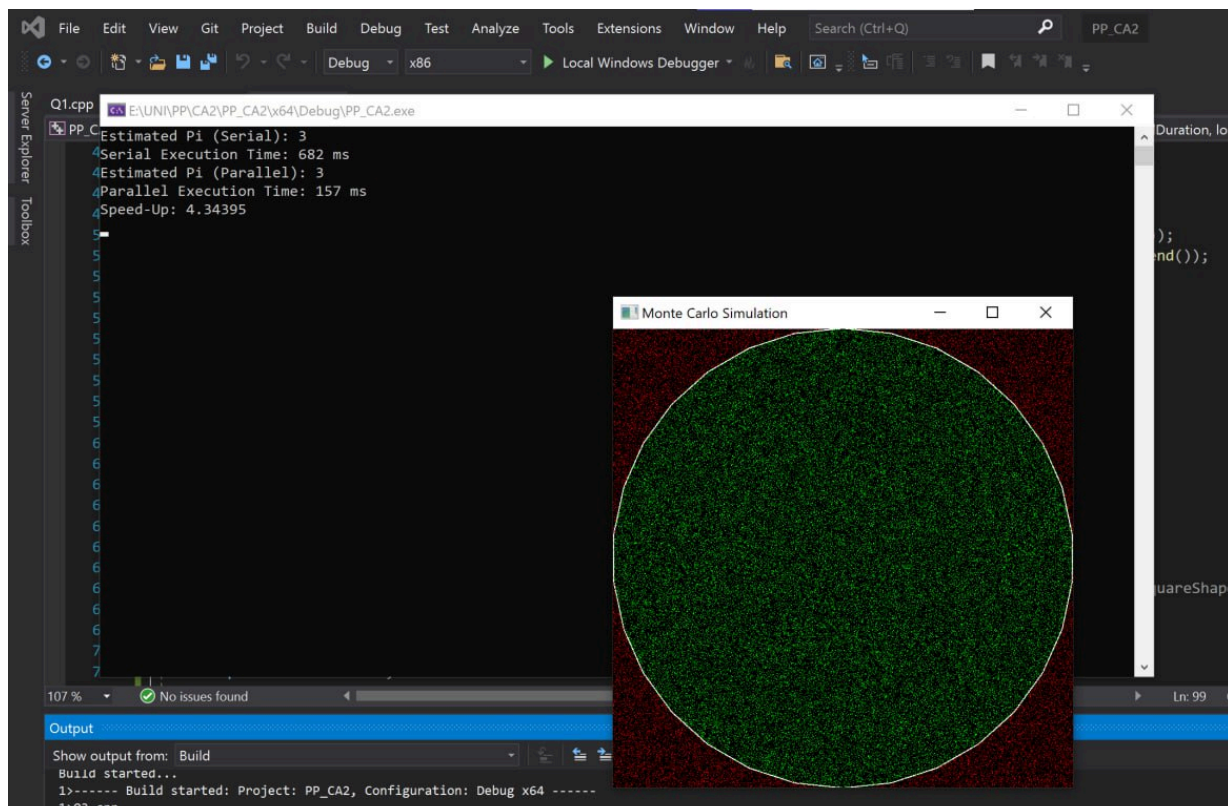
تابع `computeSerial()` مجموعه Julia را برای هر پیکسل در تصویر به طور سریال محاسبه می‌کند. مختصات حقیقی و موهومی هر پیکسل از موقعیت آن در تصویر محاسبه و سپس

برای هر نقطه تابع `julia()` فراخوانی می‌شود. رنگ هر پیکسل بر اساس تعداد تکرارها در تابع `julia()` تعیین می‌شود.

تابع `computeParallel()` همان عملیات را به‌طور موازی انجام می‌دهد:

دستور `#pragma omp parallel num_threads(8)` به‌کار رفته است تا پردازش‌ها به ۸ ترد تقسیم شوند. برای اجتناب از تداخل در نوشتن داده‌ها، ابتدا رنگ‌ها در یک بافر محلی ذخیره می‌شوند و سپس در مرحله‌ی بعدی به تصویر منتقل می‌شوند.

3. Monte Carlo



حالت سریال (`performPiCalcSerial()`):

در این حالت، تمام نقاط به‌طور خطی پردازش می‌شوند. نقاط داخل دایره با رنگ سبز و نقاط خارج از دایره با رنگ قرمز نمایش داده می‌شوند. برای هر نقطه، بررسی می‌شود که آیا در داخل دایره است یا خیر، و سپس شمارش انجام می‌شود.

حالت موازی (`performPiCalcParallel`):

در این حالت، نقاط تصادفی به طور موازی بین تردها تقسیم می‌شوند. برای جلوگیری از مشکلات همزمانی، از `thread_local` برای تولید اعداد تصادفی منحصر به فرد برای هر ترد استفاده شده است. پس از محاسبه تعداد نقاط داخل و خارج دایره در هر ترد، نتایج با استفاده از `pragma omp critical` به روزرسانی می‌شوند.