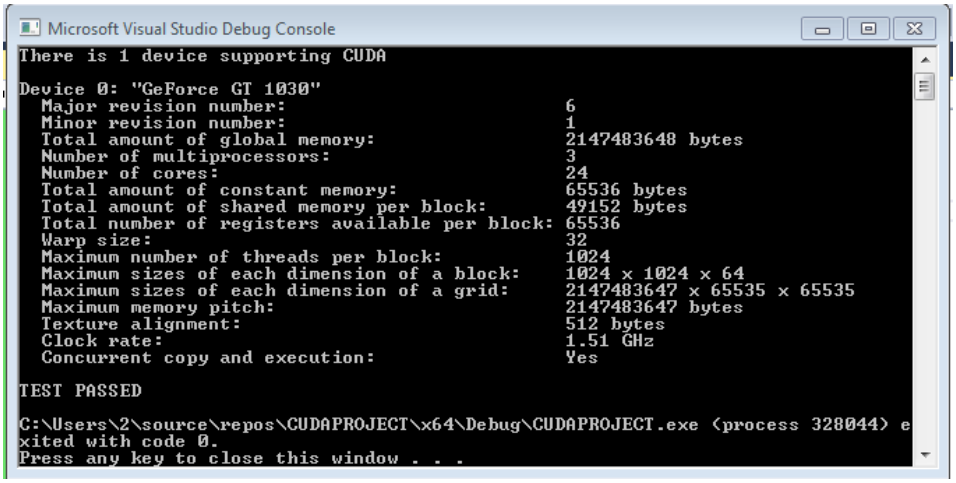
به نام خدا

گزارش آزمایش پنجم

علی نوروزبیگی ۹۹۳۱۰۶۲ – محمد مهدی نظری ۹۹۳۱۰۶۱

گام اول:

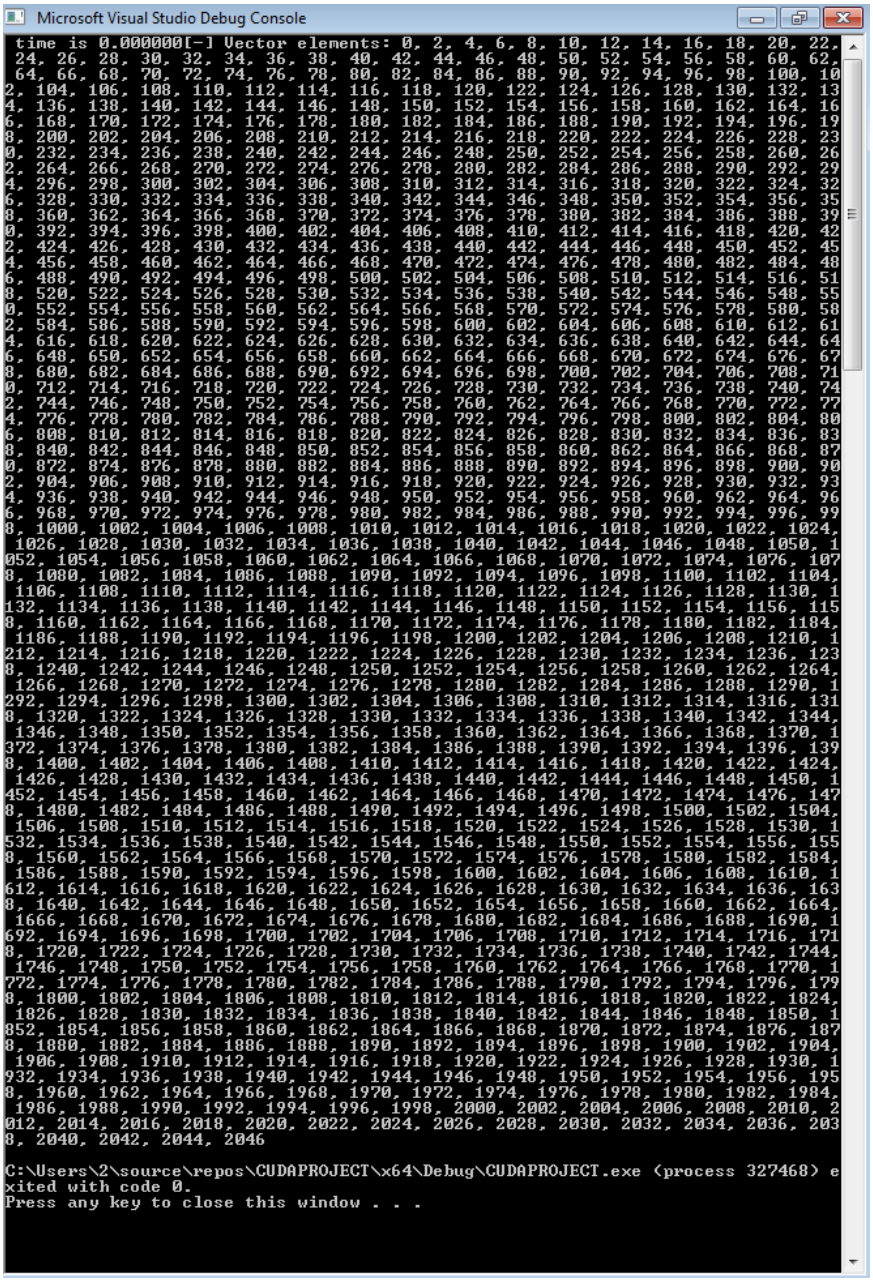
پس از اتصال به ماشین مجازی، با اجرای کد داده شده، مشخصات پردازنده گرافیکی را بررسی می کنیم:



همانطور که مشاهده می شود، اطلاعات مفیدی در مورد مجموع حافظه گلوبال، مقدار حافظه اشتراکی در هر بلاک و تعداد SM ها (۳) و تعداد وارپ ها (۳۲) قابل مشاهده است. این اطلاعات را می توان با عکس موجود در دستورکار آزمایش نیز تطابق داد.

گام دوم:

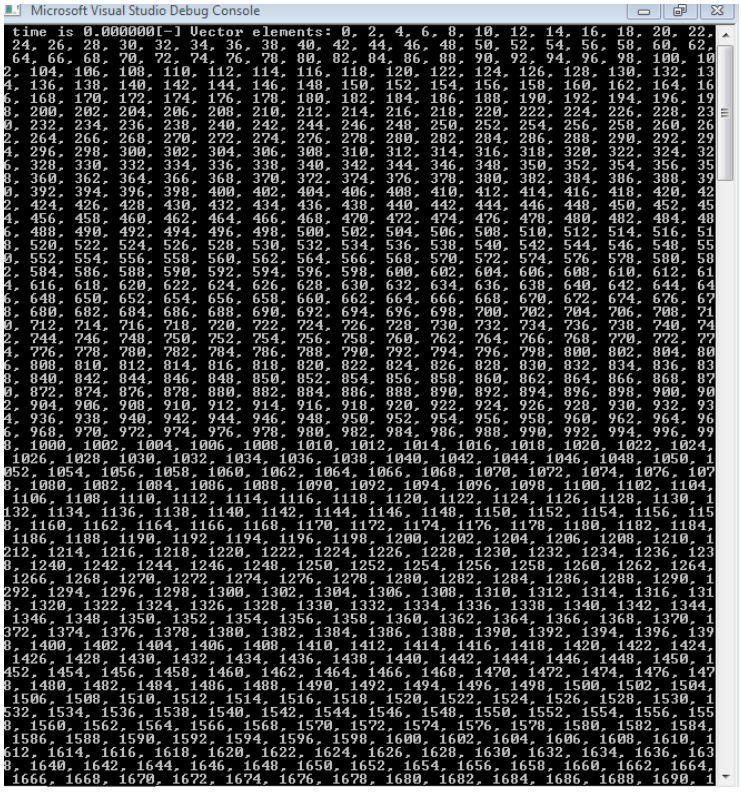
در این گام ابتدا متغیرهایی برای نگهداری آدرس شروع بردارها تعریف می شوند و سپس پردازنده گرافیکی که می خواهیم با آن عملیات را انجام دهیم انتخاب می کنیم. مطابق با دستورکار دستگاه 0 را انتخاب می کنیم. سپس باید از پردازنده گرافیکی برای آن سه بردار حافظه بگیریم و سپس دو بردار a و b که بردارهایی هستند که می خواهیم جمعشان کنیم را در حافظه پردازنده گرافیکی کپی می کنیم که بتوانیم عملیات را روی آنها انجام دهیم. اکنون نوبت به اجرای عمل جمع به صورت موازی در پردازنده گرافیکی می رسد. چون اندازه آرایه ۱۰۲۴ است، کرنلی که برای انجام عمل ضرب می نویسیم، هم ۱۰۲۴ نخ خواهد داشت و به همین دلیل تنها از threadIdx.x استفاده می کنیم و آن را در یک بلوک با ۱۰۲۴ نخ اجرا می کنیم. توجه داریم که باید cpu باید تا انتهای عمل gpu صبر کند، به همین دلیل باید آن را با gpu همگام کنیم. در انتها بعد از اطمینان از عدم وقوع خطا، نتیجه آرایه c را به cpu کپی می کنیم. نتایج جمع به صورت زیر است:

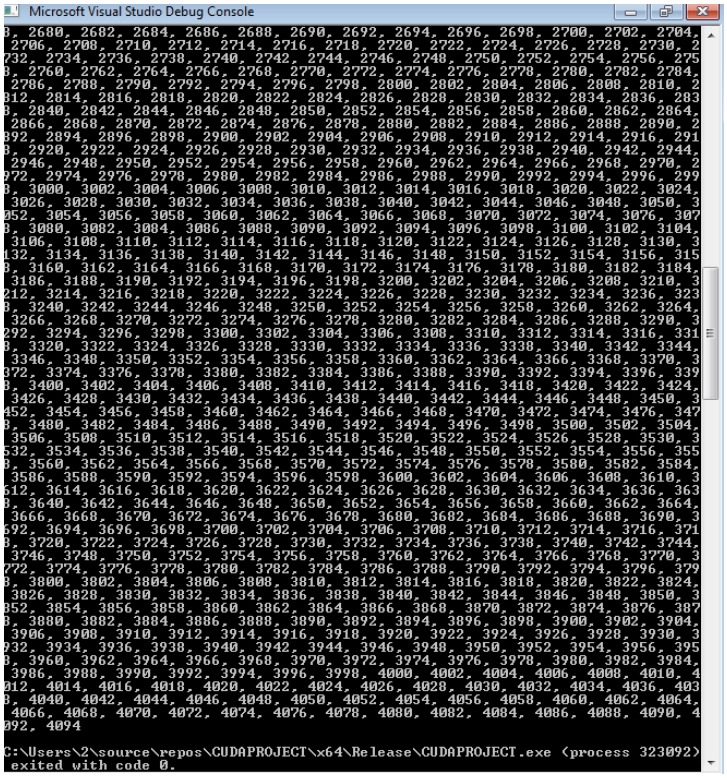


با چاپ نتایج، می بینیم که عمل جمع به درستی در پردازنده گرافیکی انجام شده است. همچنین زمان اجرای این برنامه نیز در ابتدای آن گزارش شده است. به دلیل اینکه تعداد این اعداد بسیار کم هستند، عمل جمع در زمان خیلی کمی انجام شده است.

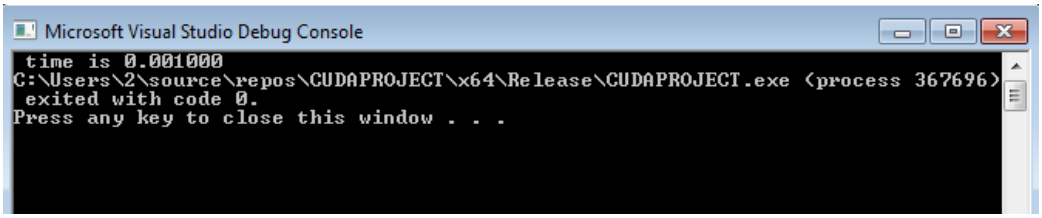
نحوه پیاده سازی برای گام های دوم و سوم به این صورت است که ابتدا دو ثابت برای تعداد بلاک و تعداد وظایفی که هر نخ انجام می دهد تعریف می کنیم. در صورتی که بخواهیم از روش اول (هر نخ n جمع انجام بدهد) گام سوم را انجام بدهیم، تعداد بلاک را برابر ۱ و تعداد وظیفه ای که هر نخ انجام می دهد را برابر n می گذاریم. در حالتی که بخواهیم n بلاک ۱۰۲۴تایی اجرا کنیم، تعداد بلاک را برابر n و تعداد وظیفه هر نخ را 1 می گذاریم.

برای بررسی درست بودن برنامه، ابتدا دو بلاک را در نظر میگیریم که هر نخ آن یک کار را انجام دهد. نتایج به صورت زیر است:

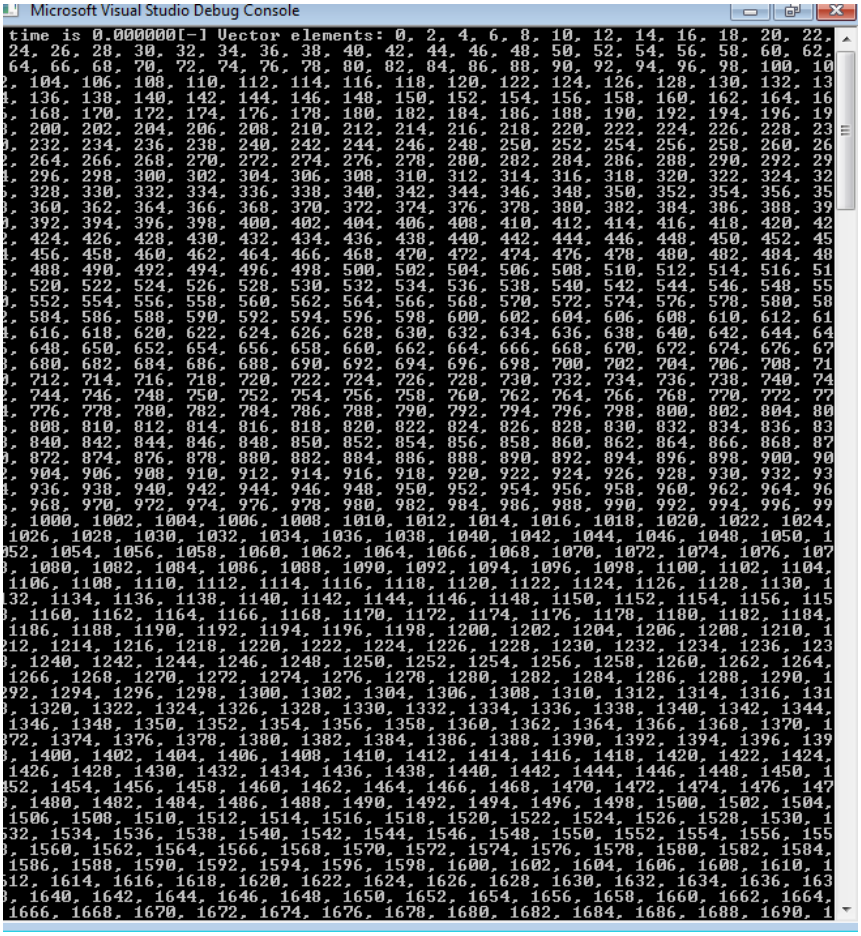


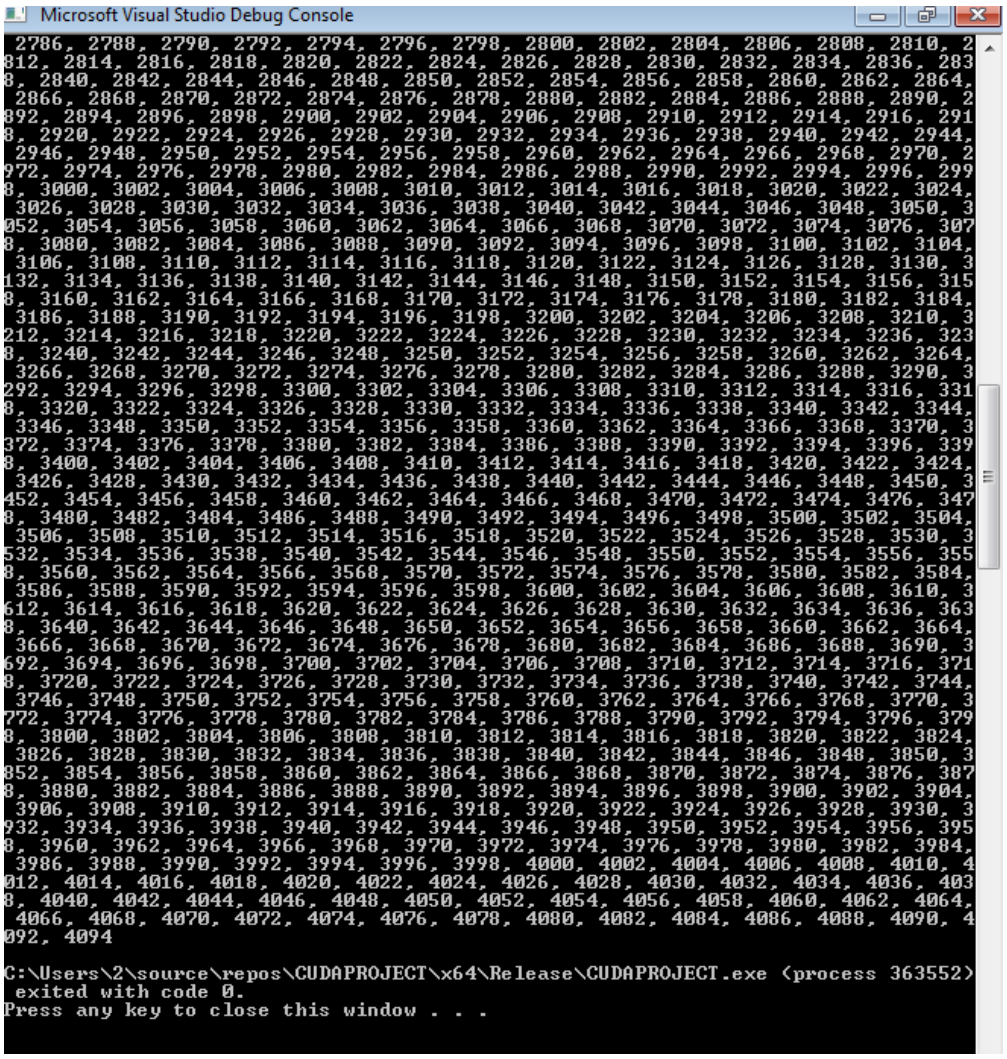


اکنون که از درستی عملکرد برنامه اطمینان پیدا کردیم، ۶۴ بلوک ۱۰۲۴ تایی را محاسبه می کنیم و زمان آن را بررسی می کنیم:



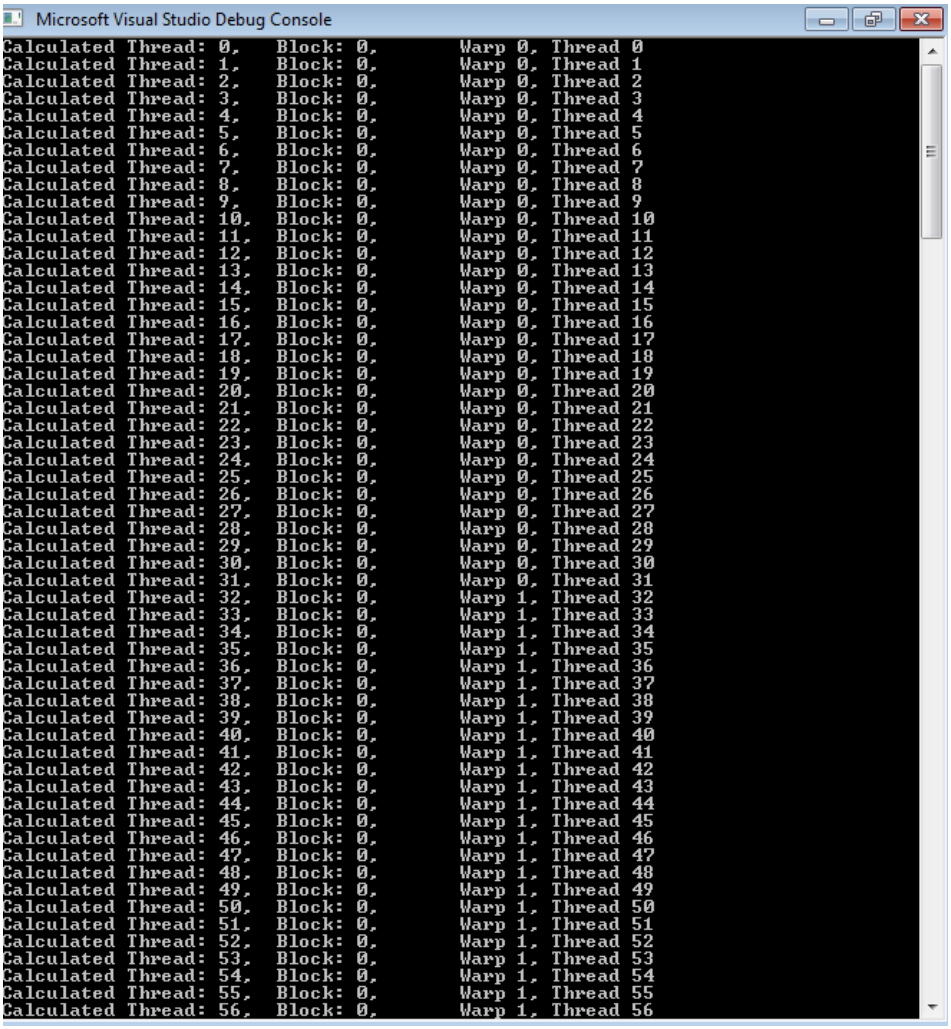
اکنون برای اطمینان از درستی برنامه در حالت بعد، حالتی را در نظر میگیریم که یک بلاک داریم و هر نخ دو کار را انجام می دهد.

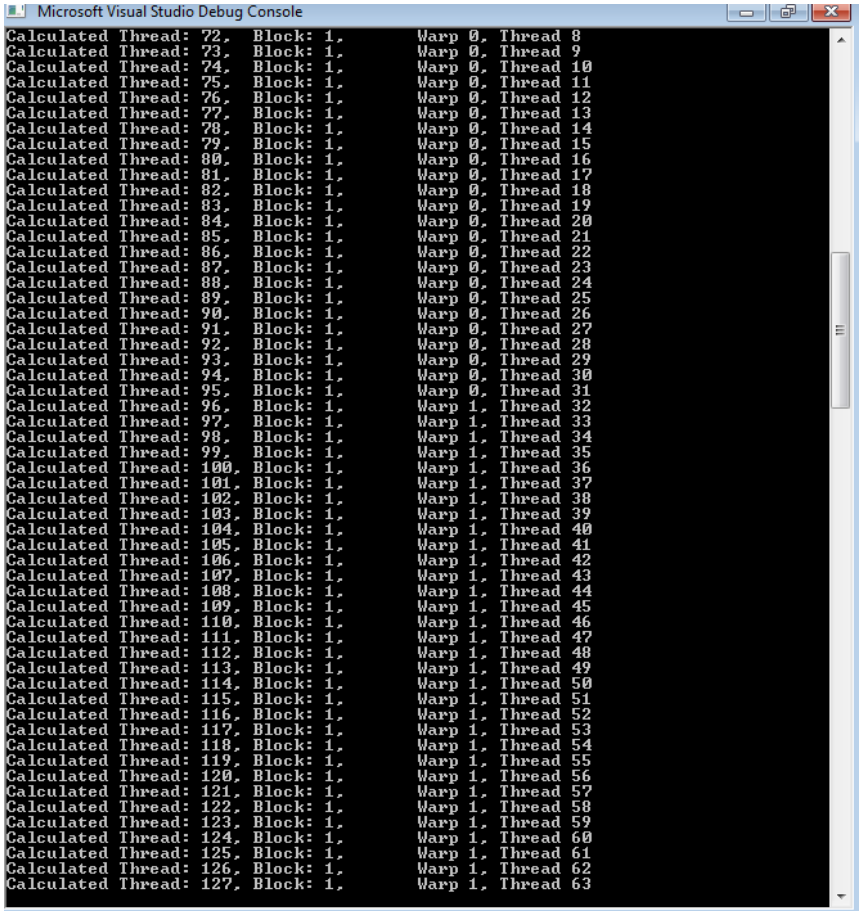




اکنون مشابه قبل با عدد بزرگتر داریم:

برای اجرای گام چهارم، به جهت اینکه بتوانیم در gpu از آرایه های cpu استفاده کنیم، از cudaMallocManaged() استفاده کردیم. سپس در کدی که برای کرنل نوشتیم، هر نخ شماره بلاک خود، آیدی و وارپ مربوط به خود را در آرایه های متناظر با آن می نویسد. سپس در cpu مقادیر آن آرایه چاپ می شود. کد این گام در انتهای کد پیوست شده موجود است. نتایج به صورت زیر است:




Microsoft Visual Studio Debug Console
Calculated Thread: 72, Block: 1,   Warp 0, Thread 8
Calculated Thread: 73, Block: 1,   Warp 0, Thread 9
Calculated Thread: 74, Block: 1,   Warp 0, Thread 10
Calculated Thread: 75, Block: 1,   Warp 0, Thread 11
Calculated Thread: 76, Block: 1,   Warp 0, Thread 12
Calculated Thread: 77, Block: 1,   Warp 0, Thread 13
Calculated Thread: 78, Block: 1,   Warp 0, Thread 14
Calculated Thread: 79, Block: 1,   Warp 0, Thread 15
Calculated Thread: 80, Block: 1,   Warp 0, Thread 16
Calculated Thread: 81, Block: 1,   Warp 0, Thread 17
Calculated Thread: 82, Block: 1,   Warp 0, Thread 18
Calculated Thread: 83, Block: 1,   Warp 0, Thread 19
Calculated Thread: 84, Block: 1,   Warp 0, Thread 20
Calculated Thread: 85, Block: 1,   Warp 0, Thread 21
Calculated Thread: 86, Block: 1,   Warp 0, Thread 22
Calculated Thread: 87, Block: 1,   Warp 0, Thread 23
Calculated Thread: 88, Block: 1,   Warp 0, Thread 24
Calculated Thread: 89, Block: 1,   Warp 0, Thread 25
Calculated Thread: 90, Block: 1,   Warp 0, Thread 26
Calculated Thread: 91, Block: 1,   Warp 0, Thread 27
Calculated Thread: 92, Block: 1,   Warp 0, Thread 28
Calculated Thread: 93, Block: 1,   Warp 0, Thread 29
Calculated Thread: 94, Block: 1,   Warp 0, Thread 30
Calculated Thread: 95, Block: 1,   Warp 0, Thread 31
Calculated Thread: 96, Block: 1,   Warp 1, Thread 32
Calculated Thread: 97, Block: 1,   Warp 1, Thread 33
Calculated Thread: 98, Block: 1,   Warp 1, Thread 34
Calculated Thread: 99, Block: 1,   Warp 1, Thread 35
Calculated Thread: 100, Block: 1,   Warp 1, Thread 36
Calculated Thread: 101, Block: 1,   Warp 1, Thread 37
Calculated Thread: 102, Block: 1,   Warp 1, Thread 38
Calculated Thread: 103, Block: 1,   Warp 1, Thread 39
Calculated Thread: 104, Block: 1,   Warp 1, Thread 40
Calculated Thread: 105, Block: 1,   Warp 1, Thread 41
Calculated Thread: 106, Block: 1,   Warp 1, Thread 42
Calculated Thread: 107, Block: 1,   Warp 1, Thread 43
Calculated Thread: 108, Block: 1,   Warp 1, Thread 44
Calculated Thread: 109, Block: 1,   Warp 1, Thread 45
Calculated Thread: 110, Block: 1,   Warp 1, Thread 46
Calculated Thread: 111, Block: 1,   Warp 1, Thread 47
Calculated Thread: 112, Block: 1,   Warp 1, Thread 48
Calculated Thread: 113, Block: 1,   Warp 1, Thread 49
Calculated Thread: 114, Block: 1,   Warp 1, Thread 50
Calculated Thread: 115, Block: 1,   Warp 1, Thread 51
Calculated Thread: 116, Block: 1,   Warp 1, Thread 52
Calculated Thread: 117, Block: 1,   Warp 1, Thread 53
Calculated Thread: 118, Block: 1,   Warp 1, Thread 54
Calculated Thread: 119, Block: 1,   Warp 1, Thread 55
Calculated Thread: 120, Block: 1,   Warp 1, Thread 56
Calculated Thread: 121, Block: 1,   Warp 1, Thread 57
Calculated Thread: 122, Block: 1,   Warp 1, Thread 58
Calculated Thread: 123, Block: 1,   Warp 1, Thread 59
Calculated Thread: 124, Block: 1,   Warp 1, Thread 60
Calculated Thread: 125, Block: 1,   Warp 1, Thread 61
Calculated Thread: 126, Block: 1,   Warp 1, Thread 62
Calculated Thread: 127, Block: 1,   Warp 1, Thread 63