

```
!apt-get --purge remove cuda nvidia* libnvidia-*
!dpkg -l | grep cuda- | awk '{print $2}' | xargs -n1 dpkg --purge
!apt-get remove cuda-*
!apt autoremove
!apt-get update

!wget https://developer.nvidia.com/compute/cuda/9.2/Prod/local_installers/cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64 -O cuda-repo-ubur
!dpkg -i cuda-repo-ubuntu1604-9-2-local_9.2.88-1_amd64.deb
!apt-key add /var/cuda-repo-9-2-local/7fa2af80.pub
!apt-get update
!apt-get install cuda-9.2
```

```
!nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

```
!pip install git+https://github.com/andreinechaev/nvcc4jupyter.git
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting git+https://github.com/andreinechaev/nvcc4jupyter.git
  Cloning https://github.com/andreinechaev/nvcc4jupyter.git to /tmp/pip-req-build-cg15v4oc
  Running command git clone --filter=blob:none --quiet https://github.com/andreinechaev/nvcc4jupyter.git /tmp/pip-req-build-cg15v4oc
  Resolved https://github.com/andreinechaev/nvcc4jupyter.git to commit aac710a35f52bb78ab34d2e52517237941399eff
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: NVCCPlugin
  Building wheel for NVCCPlugin (setup.py) ... done
  Created wheel for NVCCPlugin: filename=NVCCPlugin-0.0.2-py3-none-any.whl size=4305 sha256=942ee32d49a7cf4df1ac40b03f32372b9d90d67
  Stored in directory: /tmp/pip-ephem-wheel-cache-7r68oofs/wheels/a8/b9/18/23f8ef71ceb0f63297dd1903aedd067e6243a68ea756d6feea
Successfully built NVCCPlugin
Installing collected packages: NVCCPlugin
Successfully installed NVCCPlugin-0.0.2
```

```
%load_ext nvcc_plugin
```

```
created output directory at /content/src
Out bin /content/result.out
```

```
!sudo apt-get install gcc-7 g++-7
!sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 50
!sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-7 50
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cpp-7 gcc-7-base gcc-8-base libasan4 libcilkrts5 libgcc-7-dev libmpx2
  libstdc++-7-dev libubsan0
Suggested packages:
  gcc-7-locales g++-7-multilib gcc-7-doc gcc-7-multilib libstdc++-7-doc
The following NEW packages will be installed:
  cpp-7 g++-7 gcc-7 gcc-7-base libasan4 libcilkrts5 libgcc-7-dev
  libmpx2 libstdc++-7-dev libubsan0
0 upgraded, 11 newly installed, 0 to remove and 24 not upgraded.
Need to get 32.0 MB of archives.
After this operation, 111 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal/universe amd64 gcc-7-base amd64 7.5.0-6ubuntu2 [18.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/universe amd64 cpp-7 amd64 7.5.0-6ubuntu2 [8,588 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal/universe amd64 libasan4 amd64 7.5.0-6ubuntu2 [358 kB]
Get:4 http://archive.ubuntu.com/ubuntu focal/universe amd64 libubsan0 amd64 7.5.0-6ubuntu2 [126 kB]
Get:5 http://archive.ubuntu.com/ubuntu focal/universe amd64 libcilkrts5 amd64 7.5.0-6ubuntu2 [42.7 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal/universe amd64 gcc-8-base amd64 8.4.0-3ubuntu2 [18.7 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal/universe amd64 libmpx2 amd64 8.4.0-3ubuntu2 [11.8 kB]
Get:8 http://archive.ubuntu.com/ubuntu focal/universe amd64 libgcc-7-dev amd64 7.5.0-6ubuntu2 [2,311 kB]
Get:9 http://archive.ubuntu.com/ubuntu focal/universe amd64 gcc-7 amd64 7.5.0-6ubuntu2 [9,381 kB]
Get:10 http://archive.ubuntu.com/ubuntu focal/universe amd64 libstdc++-7-dev amd64 7.5.0-6ubuntu2 [1,471 kB]
Get:11 http://archive.ubuntu.com/ubuntu focal/universe amd64 g++-7 amd64 7.5.0-6ubuntu2 [9,689 kB]
Fetched 32.0 MB in 4s (8,084 kB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based frontend cannot be used. at /usr/share/perl5/Debconf/F
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package gcc-7-base:amd64.
(Reading database ... 122518 files and directories currently installed.)
```

```

Preparing to unpack .../00-gcc-7-base_7.5.0-6ubuntu2_amd64.deb ...
Unpacking gcc-7-base:amd64 (7.5.0-6ubuntu2) ...
Selecting previously unselected package cpp-7.
Preparing to unpack .../01-cpp-7_7.5.0-6ubuntu2_amd64.deb ...
Unpacking cpp-7 (7.5.0-6ubuntu2) ...
Selecting previously unselected package libasan4:amd64.
Preparing to unpack .../02-libasan4_7.5.0-6ubuntu2_amd64.deb ...
Unpacking libasan4:amd64 (7.5.0-6ubuntu2) ...
Selecting previously unselected package libubsan0:amd64.
Preparing to unpack .../03-libubsan0_7.5.0-6ubuntu2_amd64.deb ...
Unpacking libubsan0:amd64 (7.5.0-6ubuntu2) ...
Selecting previously unselected package libcilkrts5:amd64.
Preparing to unpack .../04-libcilkrts5_7.5.0-6ubuntu2_amd64.deb ...
Unpacking libcilkrts5:amd64 (7.5.0-6ubuntu2) ...
Selecting previously unselected package gcc-8-base:amd64.
Preparing to unpack .../05-gcc-8-base_8.4.0-3ubuntu2_amd64.deb ...
Unpacking gcc-8-base:amd64 (8.4.0-3ubuntu2) ...
Selecting previously unselected package libmpx2:amd64.
Preparing to unpack .../06-libmpx2_8.4.0-3ubuntu2_amd64.deb ...
Unpacking libmpx2:amd64 (8.4.0-3ubuntu2) ...
Selecting previously unselected package libgcc-7-dev:amd64.
Preparing to unpack .../07-libgcc-7-dev_7.5.0-6ubuntu2_amd64.deb ...
Unpacking libgcc-7-dev:amd64 (7.5.0-6ubuntu2) ...

```

```

%%cu
#include <stdio.h>

// Size of array
#define N 1048576

// Kernel
__global__ void add_vectors(double *a, double *b, double *c)
{
    int id = blockDim.x * blockIdx.x + threadIdx.x;
    if(id < N) c[id] = a[id] + b[id];
}

// Main program
int main()
{
    // Number of bytes to allocate for N doubles
    size_t bytes = N*sizeof(double);

    // Allocate memory for arrays A, B, and C on host
    double *A = (double*)malloc(bytes);
    double *B = (double*)malloc(bytes);
    double *C = (double*)malloc(bytes);

    // Allocate memory for arrays d_A, d_B, and d_C on device
    double *d_A, *d_B, *d_C;
    cudaMalloc(&d_A, bytes);
    cudaMalloc(&d_B, bytes);
    cudaMalloc(&d_C, bytes);

    // Fill host arrays A and B
    for(int i=0; i<N; i++)
    {
        A[i] = 1.0;
        B[i] = 2.0;
    }

    // Copy data from host arrays A and B to device arrays d_A and d_B
    cudaMemcpy(d_A, A, bytes, cudaMemcpyHostToDevice);
    cudaMemcpy(d_B, B, bytes, cudaMemcpyHostToDevice);

    // Set execution configuration parameters
    //   thr_per_blk: number of CUDA threads per grid block
    //   blk_in_grid: number of blocks in grid
    int thr_per_blk = 256;
    int blk_in_grid = ceil( float(N) / thr_per_blk );

    // Launch kernel
    add_vectors<<< blk_in_grid, thr_per_blk >>>(d_A, d_B, d_C);

    // Copy data from device array d_C to host array C
    cudaMemcpy(C, d_C, bytes, cudaMemcpyDeviceToHost);

    // Verify results
    double tolerance = 1.0e-14;
    for(int i=0; i<N; i++)
    {
        if( fabs(C[i] - 3.0) > tolerance)
        {
            printf("\nError: value of C[%d] = %d instead of 3.0\n\n", i, C[i]);

```

```

        exit(1);
    }
}

// Free CPU memory
free(A);
free(B);
free(C);

// Free GPU memory
cudaFree(d_A);
cudaFree(d_B);
cudaFree(d_C);

printf("\n-----\n");
printf("__SUCCESS__\n");
printf("-----\n");
printf("N                = %d\n", N);
printf("Threads Per Block = %d\n", thr_per_blk);
printf("Blocks In Grid    = %d\n", blk_in_grid);
printf("-----\n\n");

return 0;
}

```

```

-----
__SUCCESS__
-----
N                = 1048576
Threads Per Block = 256
Blocks In Grid    = 4096
-----

```

Addition of 2 vectors

```

%%cu
#include<stdio.h>
#include<iostream>
#include<cuda.h>
using namespace std;
__global__ void arradd(int *x,int *y, int *z)
{
    int id=blockIdx.x * blockDim.x+threadIdx.x;
    z[id]=x[id]+y[id];
}
int main()
{
    int a[6]={1,2,3,4,7,9};
    int b[6]={1,2,3,4,9,7};
    int c[6];
    int *d,*e,*f;
    int i;
    cudaMalloc((void **)&d,6*sizeof(int));
    cudaMalloc((void **)&e,6*sizeof(int));
    cudaMalloc((void **)&f,6*sizeof(int));
    cudaMemcpy(d,a,6*sizeof(int),cudaMemcpyHostToDevice);
    cudaMemcpy(e,b,6*sizeof(int),cudaMemcpyHostToDevice);
    arradd<<<2,3>>>(d,e,f);
    cudaMemcpy(c,f,6*sizeof(int),cudaMemcpyDeviceToHost);
    printf("\nSum of two arrays:\n ");
    for(i=0;i<6;i++)
    {
        cout<<c[i]<<" ";
    }
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);
    return 0;
}

```

```

Sum of two arrays:
2 4 6 8 16 16

```

Multiplication of 2 Matrices

```

%%cu
#include<stdio.h>
#include<cuda.h>
#define row1 2
#define col1 3
#define row2 3
#define col2 2
__global__ void matproduct(int *l,int *m, int *n)
{
    int x=blockIdx.x;
    int y=blockIdx.y;
    int k;

    n[col2*y+x]=0;
    for(k=0;k<col1;k++)
    {
        n[col2*y+x]=n[col2*y+x]+l[col1*y+k]*m[col2*k+x];
    }
}
int main()
{
    int a[row1][col1];
    int b[row2][col2];
    int c[row1][col2];
    int *d,*e,*f;
    int i,j;
    printf("\n Enter elements of first matrix of size 2*3\n");
    int k=1;
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            a[i][j]= k;
            k++;
        }
    }
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
    }

    printf("\n Enter elements of second matrix of size 3*2\n");
    for(i=0;i<row2;i++)
    {
        for(j=0;j<col2;j++)
        {
            b[i][j]=k;
            k++;
        }
    }
    for(i=0;i<row2;i++)
    {
        for(j=0;j<col2;j++)
        {
            printf("%d\t",b[i][j]);
        }
        printf("\n");
    }
    cudaMalloc((void **)&d,row1*col1*sizeof(int));
    cudaMalloc((void **)&e,row2*col2*sizeof(int));
    cudaMalloc((void **)&f,row1*col2*sizeof(int));
    cudaMemcpy(d,a,row1*col1*sizeof(int),cudaMemcpyHostToDevice);
    cudaMemcpy(e,b,row2*col2*sizeof(int),cudaMemcpyHostToDevice);
    dim3 grid(col2,row1);
    matproduct<<<grid,1>>>(d,e,f);
    cudaMemcpy(c,f,row1*col2*sizeof(int),cudaMemcpyDeviceToHost);
    printf("\nProduct of two matrices:\n ");
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col2;j++)
        {
            printf("%d\t",c[i][j]);
        }
        printf("\n");
    }
    cudaFree(d);
    cudaFree(e);
    cudaFree(f);

```

```
    return 0;  
}
```

Enter elements of first matrix of size 2*3

1	2	3
4	5	6

Enter elements of second matrix of size 3*2

7	8
9	10
11	12

Product of two matrices:

58	64
139	154

✓ 1s completed at 11:29 AM



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.