**Vision of the Department**

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

**Mission of the Department**

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.*

**Session 2025-2026**

| | |
|---|---|
| **Vision:** To help businesses uncover crucial insights | **Mission:** To be a good data scientist |

**Program Educational Objectives of the program (PEO):** (broad statements that describe the professional and career accomplishments)

| | | | |
|---|---|---|---|
| PEO1 | **Preparation** | **P: Preparation** | **Pep-CL abbreviation pronounce as Pep-si-lL easy to recall** |
| PEO2 | **Core Competence** | **E: Environment (Learning Environment)** | |
| PEO3 | **Breadth** | **P: Professionalism** | |
| PEO4 | **Professionalism** | **C: Core Competence** | |
| PEO5 | **Learning Environment** | **L: Breadth (Learning in diverse areas)** | |

**Program Outcomes (PO):** 1. Understand and Apply Parallel Programming Concepts

2. Analyse and Improve Program Performance.

3. Demonstrate Practical Skills in HPC Tools and Environments.

**Keywords of POs:**
Engineering knowledge, Problem analysis, Design/development of solutions, Conduct Investigations of Complex Problems, Engineering Tool Usage, The Engineer and The World, Ethics, Individual and Collaborative Team work, Communication, Project Management and Finance, Life-Long Learning

**PSO Keywords:** Cutting edge technologies, Research

“I am an engineer, and I know how to apply engineering knowledge to investigate, analyse and design solutions to complex problems using tools for entire world following all ethics in a collaborative way with proper management skills throughout my life.” *to contribute to the development of cutting-edge technologies and Research*.

**Integrity:** I will adhere to the Laboratory Code of Conduct and ethics in its entirety.

**Name and Signature of Student and Date**
Soham pimpalgaonkar – 04/11/2025

| **Session** | **2025-26 (ODD)** | **Course Name** | **HPC Lab** |
|---|---|---|---|
| **Semester** | **7** | **Course Code** | 22ADS706 |
| **Roll No** | 62 | **Name of Student** | Soham pimpalgaonkar |

| | |
|---|---|
| Practical Number | 8 |
| Course Outcome | 1. Understand and Apply Parallel Programming Concepts<br>2. Analyse and Improve Program Performance |
| Aim | Introduction to GPU Computing |
| Problem Definition | Introduction to GPU Computing |
| Theory (100 words) | **CUDA (Compute Unified Device Architecture)** is a parallel computing platform and programming model developed by **NVIDIA**. It provides a set of extensions to standard programming languages like C/C++ (called **CUDA C/C++**) that allow developers to harness the parallel power of NVIDIA GPUs.<br><br>**The Heterogeneous Programming Model**<br>The CUDA model is **heterogeneous**, meaning an application uses both the CPU and the GPU:<br>1. **Host Code:** The sequential parts of the application run on the **CPU (Host)**.<br>2. **Device Code:** The parallel, compute-intensive parts are written as special functions called **kernels** that run on the **GPU (Device)**.<br><br>**Typical CUDA Program Flow**<br>A standard CUDA program involves the following steps:<br>1. **Allocation:** Allocate memory on both the Host (CPU RAM) and the Device (GPU VRAM).<br>2. **Transfer:** Copy input data from **Host memory** to **Device memory** (via the PCIe bus).<br>o *Function:* cudaMemcpy(..., cudaMemcpyHostToDevice)<br>3. **Kernel Launch:** The Host launches the **kernel** function on the Device. The kernel runs in parallel across thousands of threads.<br>o *Syntax:* kernel_name<<<GridDim, BlockDim>>>(arguments);<br>4. **Transfer Back:** Copy the results from **Device memory** back to **Host memory**. |

- *Function:* cudaMemcpy(..., cudaMemcpyDeviceToHost)

5. **Synchronization:** The CPU code waits for the GPU kernel execution to complete (often implicit in the copy-back step).
6. **Cleanup:** Free memory on both the Host and the Device.

Steps for execution

**Step 1: Check Prerequisites**

nvcc --version

**Step 2: Create a CUDA File**

**Step 3: Compile the Program**

nvcc vector_add.c -0 vector_add

**Step 4: Run the Program**

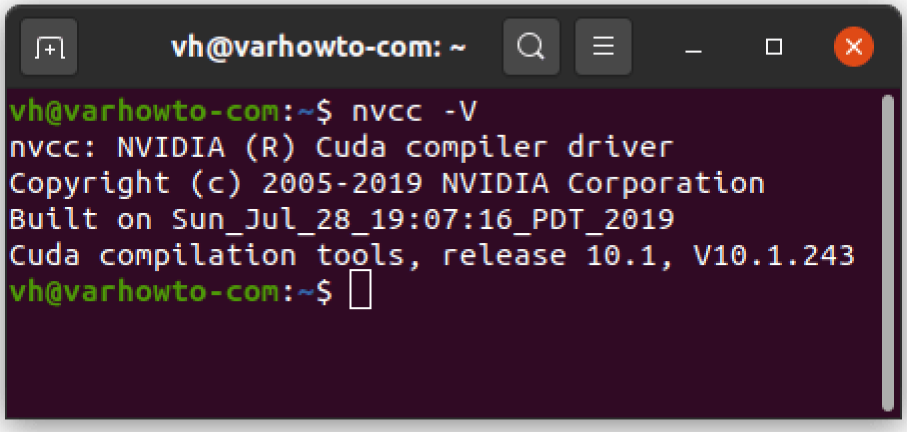
./vector_add

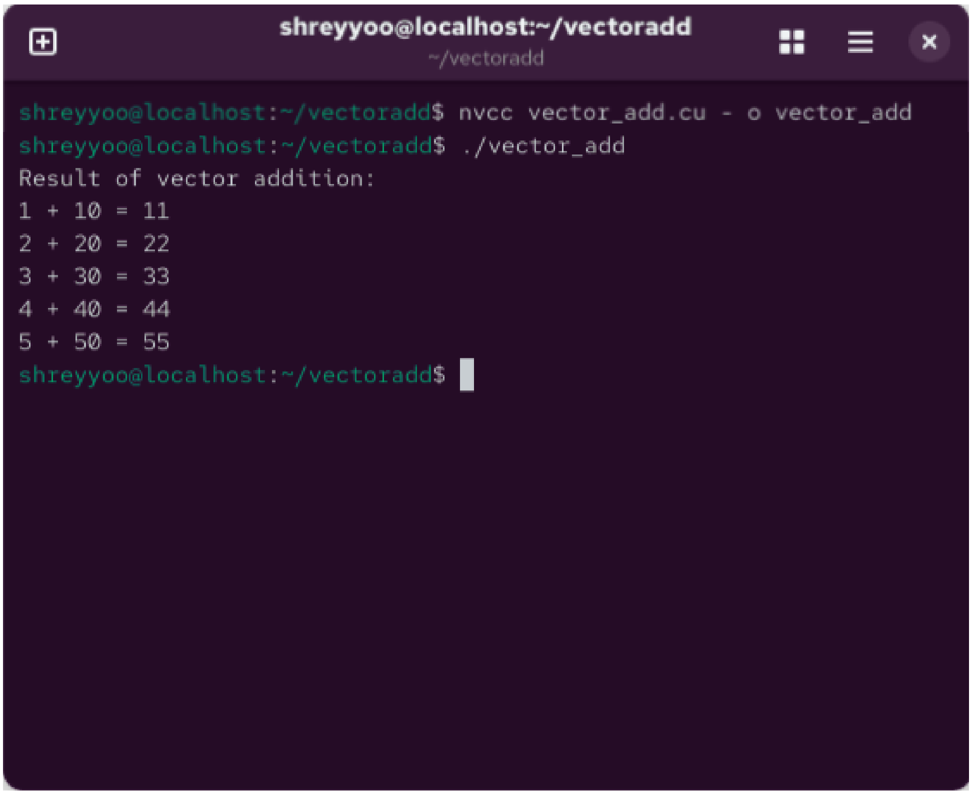
**Step 5: Verify GPU Availability**

Nvidia-smi

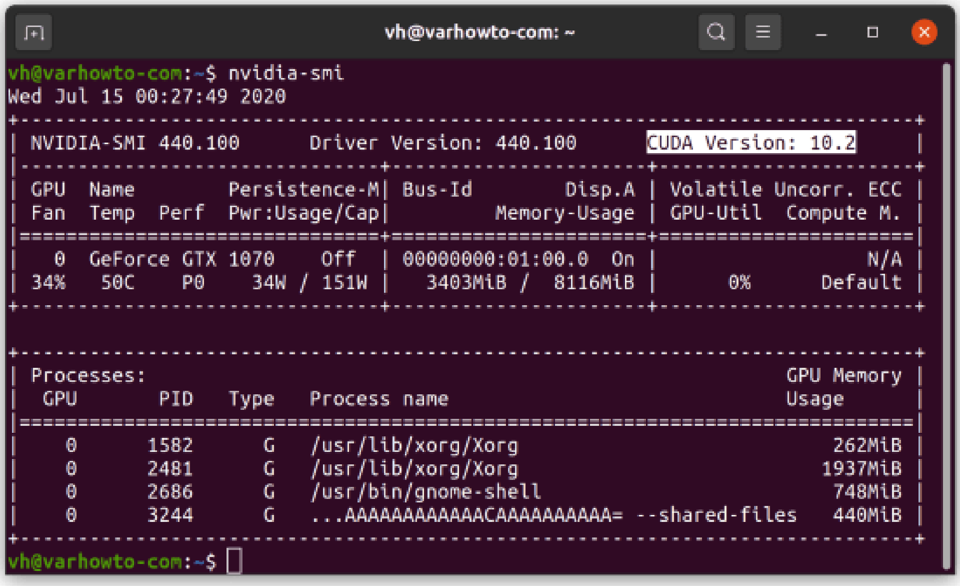
Code:

```
#include <stdio.h>
__global__ void add(int *a, int *b, int *c, int n) {
int index = threadIdx.x;
if (index < n)
c[index] = a[index] + b[index];
}
int main(void) {
int n = 5;
int a[5] = {1, 2, 3, 4, 5};
int b[5] = {10, 20, 30, 40, 50};
int c[5] = {0};
int *d_a, *d_b, *d_c;
int size = n * sizeof(int);
cudaMalloc((void **)&d_a, size);
cudaMalloc((void **)&d_b, size);
cudaMalloc((void **)&d_c, size);
cudaMemcpy(d_a, a, size, cudaMemcpyHostToDevice);
cudaMemcpy(d_b, b, size, cudaMemcpyHostToDevice);
add<<<1, n>>>(d_a, d_b, d_c, n);
cudaMemcpy(c, d_c, size, cudaMemcpyDeviceToHost);
printf("Result of vector addition:\n");
for (int i = 0; i < n; i++)
printf("%d + %d = %d\n", a[i], b[i], c[i]);
cudaFree(d_a);
cudaFree(d_b);
cudaFree(d_c);
return 0;
```

<table>
<tr><td></td><td>}</td></tr>
<tr><td>Output</td><td>




</td></tr>
</table>



```
vh@varhowto-com:~$ nvidia-smi
Wed Jul 15 00:27:49 2020
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 440.100      Driver Version: 440.100      CUDA Version: 10.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 1070     Off  | 00000000:01:00.0  On |                  N/A |
| 34%   50C    P0    34W / 151W |   3403MiB /  8116MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|    0      1582      G   /usr/lib/xorg/Xorg                           262MiB |
|    0      2481      G   /usr/lib/xorg/Xorg                          1937MiB |
|    0      2686      G   /usr/bin/gnome-shell                         748MiB |
|    0      3244      G   ...AAAAAAAAAAAACAAAAAAAAAA= --shared-files   440MiB |
+-----------------------------------------------------------------------------+
vh@varhowto-com:~$
```

| | |
|---|---|
| Output Analysis | The program executes successfully using CUDA and gives us the output. |
| Link of student Github profile where lab assignment has been uploaded | https://github.com/Sohampimpalgaonkar/HPC |
| Conclusion | The experiment successfully validated the principles of heterogeneous parallel computing by implementing a performance-critical task using NVIDIA CUDA on a CentOS platform. |

Nagar Yuwak Shikshan Sanstha's

# Yeshwantrao Chavan College of Engineering

(An Autonomous Institution affiliated to Rashtrasant Tukadoji Maharaj Nagpur University)

Hingna Road, Wanadongri, Nagpur - 441 110

NAAC A++

Ph.: 07104-237919, 234623, 329249, 329250 Fax: 07104-232376, Website: www.ycce.edu

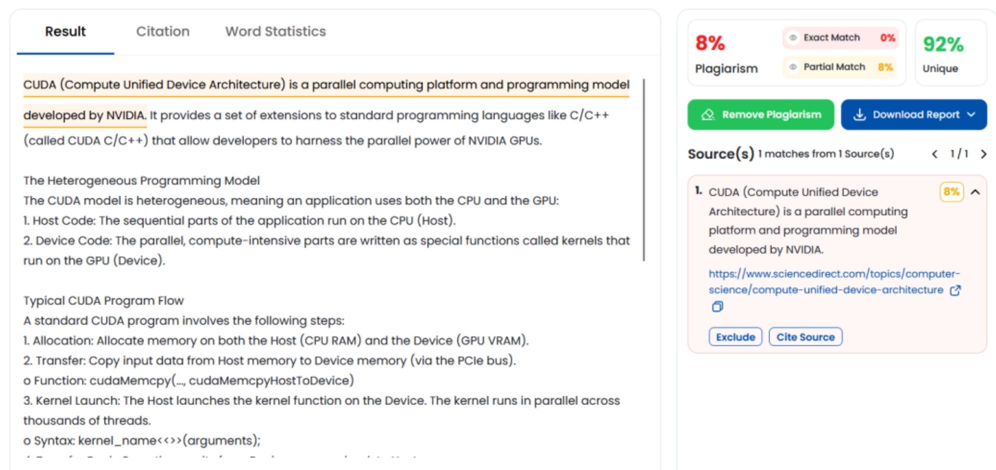
## Department of Computer Technology

**Vision of the Department**

*To be a well-known centre for pursuing computer education through innovative pedagogy, value-based education and industry collaboration.*

**Mission of the Department**

*To establish learning ambience for ushering in computer engineering professionals in core and multidisciplinary area by developing Problem-solving skills through emerging technologies.*

<table>
<tr><td>Plag Report (Similarity index < 12%)</td><td>
</td></tr>
<tr><td>Date</td><td>04/11/2025</td></tr>
</table>