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 Artificial Intelligence & Data Science

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: Dream of where you want.	**Mission:** Means to achieve Vision

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): (statements that describe what a student should be able to do and know by the end of a program)

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.

Sanskruti. Paunikar 31/10/2025
Name and Signature of Student and Date
(Signature and Date in Handwritten)

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	2025-26 (ODD)	**Course Name**	High Performance Computing Lab
Semester	7 AIDS	**Course Code**	22ADS702
Roll No	21	**Name of Student**	Sanskruti. Paunikar

Practical Number	9-Project
Course Outcome	**CO1:-**Understand and Apply Parallel Programming Concepts **CO1:-**Analyze and Improve Program Performance. **CO3:-**Demonstrate Practical Skills in HPC Tools and Environments.
Aim	Mini Project: Performance Comparison Static Arrays vs Dynamic Arrays Performance
Theory (100 words)	This project compares the performance of static and dynamic arrays based on execution time and memory efficiency. Static arrays have a fixed size and allocate memory at compile time, making them faster for access and operations where size is known in advance. Dynamic arrays, on the other hand, can resize at runtime, offering flexibility but with additional overhead during resizing and memory allocation. Both support constant-time access (O(1)), but dynamic arrays may experience occasional delays due to resizing. Overall, static arrays are more efficient in speed, while dynamic arrays provide adaptability for variable-sized data.
Procedure and Execution (100 Words)	Steps of Implementation: 1. Update system: sudo yum update -y 2. Install Python: sudo yum install python3 -y 3. Install Matplotlib: pip3 install matplotlib 4. Create file: nano array_comparison.py and paste the code. 5. Save & exit: Ctrl + O, Enter, Ctrl + X 6. Run program: python3 array_comparison.py 7. View results: Check execution times and performance graph.

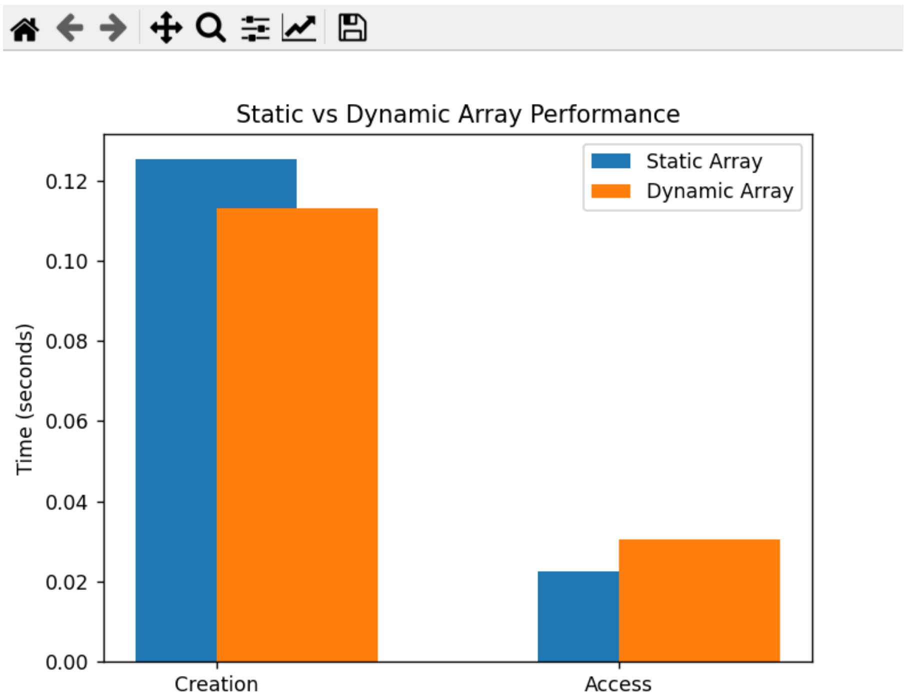
	Code:
	import time
	import array
	import random
	import matplotlib.pyplot as plt
	
	# Static Array
	static_arr = array.array('i', [0]*1000000) # 1 million integers

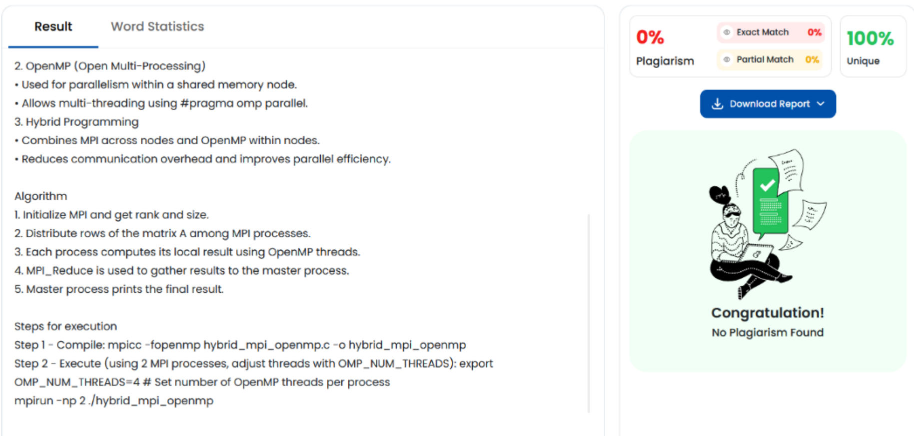
```
start_time = time.time()
for i in range(len(static_arr)):
    static_arr[i] = i
end_time = time.time()
static_creation_time = end_time - start_time
print(f"Static Array Creation Time: {static_creation_time:.6f} seconds")

#Dynamic Array
dynamic_arr = []

start_time = time.time()
for i in range(1000000):
    dynamic_arr.append(i)
end_time = time.time()
dynamic_creation_time = end_time - start_time
print(f"Dynamic Array Creation Time: {dynamic_creation_time:.6f}
seconds")

#Access Time Comparison
indices = [random.randint(0, 999999) for _ in range(100000)]

start_time = time.time()
for i in indices:
    _ = static_arr[i]
end_time = time.time()
static_access_time = end_time - start_time
print(f"Static Array Access Time: {static_access_time:.6f} seconds")

start_time = time.time()
for i in indices:
    _ = dynamic_arr[i]
end_time = time.time()
dynamic_access_time = end_time - start_time
print(f"Dynamic Array Access Time: {dynamic_access_time:.6f}
seconds")

#  Plot the Results
operations = ['Creation', 'Access']
static_times = [static_creation_time, static_access_time]
dynamic_times = [dynamic_creation_time, dynamic_access_time]

plt.bar(operations, static_times, width=0.4, label='Static Array',
align='center')
```

```
plt.bar(operations, dynamic_times, width=0.4, label='Dynamic Array', align='edge')

plt.ylabel('Time (seconds)')
plt.title('Static vs Dynamic Array Performance')
plt.legend()
plt.show()
```

Output:

```
PS C:\Users\Sanskruti> & C:/Users/Sanskruti/.ms-ad/python.exe c:/Users/Sanskruti/OneDrive/Desktop/array_comparison.py
Static Array Creation Time: 0.125394 seconds
Dynamic Array Creation Time: 0.113191 seconds
Static Array Access Time: 0.022655 seconds
Dynamic Array Access Time: 0.030466 seconds
PS C:\Users\Sanskruti>
```



Output Analysis	The output displays the execution times for creation and access operations of both static and dynamic arrays. Static arrays show faster creation time since memory is pre-allocated and fixed. Dynamic arrays take slightly longer due to resizing and memory reallocation during element insertion. However, access time for both remains almost the same, as both provide constant-time (O(1)) indexing. The bar graph clearly illustrates that static arrays are more efficient in speed, while dynamic arrays offer greater flexibility at a small performance cost.
Github link	https://github.com/sanskruti-1234/HPC.git
Conclusion	The performance comparison shows that static arrays are faster and more memory-efficient for fixed-size data, as they avoid resizing overhead. Dynamic arrays, though slightly slower due to runtime resizing, provide greater flexibility when the data size is unknown or frequently changes. Overall, static arrays are ideal for predictable, fixed datasets, while dynamic arrays are better suited for applications requiring scalability and adaptability.
Plag Report (Similarity index < 12%)	
Date	31/10/2025