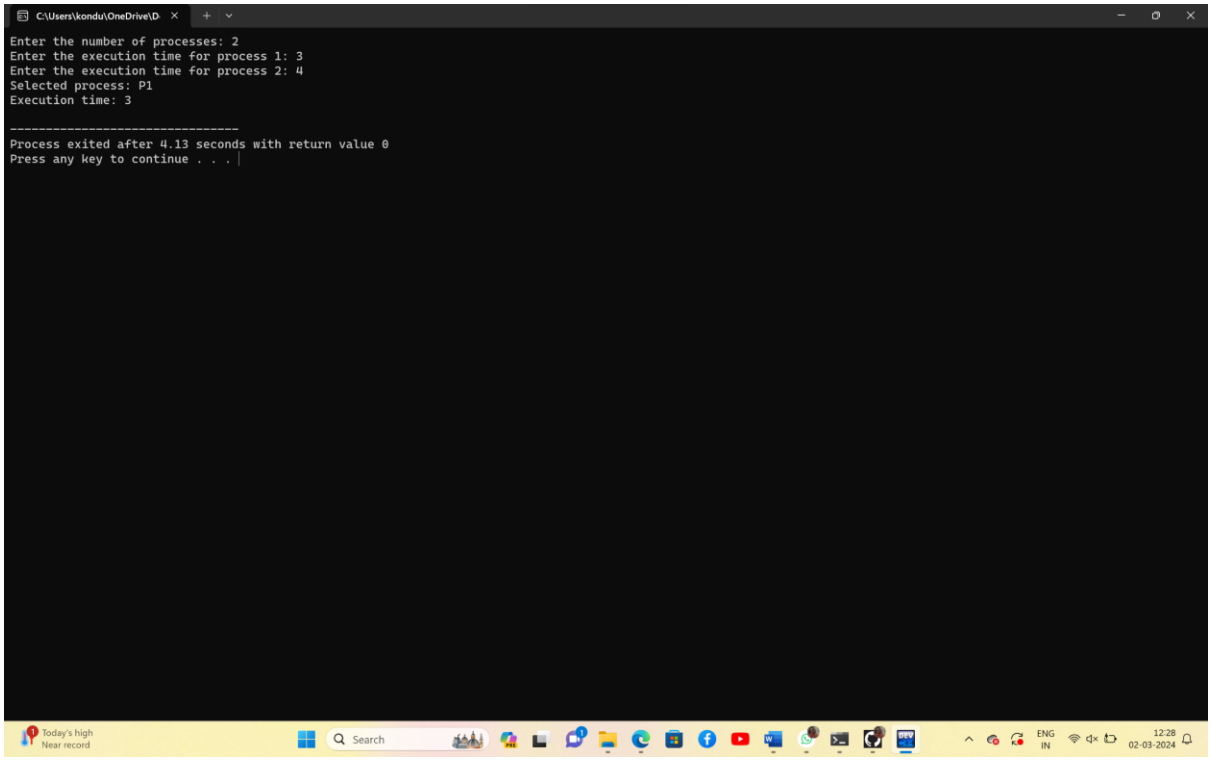
```c
#include <stdio.h>

// Structure to represent a process
struct Process {
    int id;            // Process ID
    int executionTime;  // Execution time
};

// Function to select the process with the smallest execution time
int selectProcess(struct Process processes[], int n) {
    int shortestTime = processes[0].executionTime;
    int shortestIndex = 0;

    // Find the process with the smallest execution time
    for (int i = 1; i < n; i++) {
        if (processes[i].executionTime < shortestTime) {
            shortestTime = processes[i].executionTime;
            shortestIndex = i;
        }
    }

    return shortestIndex;
}

int main() {
    int n; // Number of processes
    printf("Enter the number of processes: ");
    scanf("%d", &n);

    struct Process processes[n]; // Array to store processes
```

```
    // Input the process details
    for (int i = 0; i < n; i++) {
        printf("Enter the execution time for process %d: ", i + 1);
        scanf("%d", &processes[i].executionTime);
        processes[i].id = i + 1;
    }

    // Select the process with the smallest execution time
    int selectedProcessIndex = selectProcess(processes, n);
    struct Process selectedProcess = processes[selectedProcessIndex];

    // Output the selected process
    printf("Selected process: P%d\n", selectedProcess.id);
    printf("Execution time: %d\n", selectedProcess.executionTime);

    return 0;
}
```


C:\Users\kondu\OneDrive\D
Enter the number of processes: 2
Enter the execution time for process 1: 3
Enter the execution time for process 2: 4
Selected process: P1
Execution time: 3

Process exited after 4.13 seconds with return value 0
Press any key to continue . . .
Today's high
Near record
Search
ENG
IN
12:28
02-03-2024