	Marwadi University Faculty of Engineering and Technology Department of Information and Communication Technology
Subject: Design and Analysis of Algorithm	**Aim: :** Implementing application-based algorithms using Greedy Programming Approach
Experiment No: 07	**Date: 13/09/2025** **Enrollment No: 92301733049**

AIM: Implementing application-based algorithms using Greedy Programming Approach

I. Job Scheduling Problem

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

struct Job {
    char id;     // Job Id
    int deadline; // Deadline of job
    int profit;   // Profit if job is done before deadline
};

// Comparator function to sort jobs according to profit
bool compare(Job a, Job b) {
    return (a.profit > b.profit);
}

void jobScheduling(Job arr[], int n) {
    // Sort jobs by profit (descending)
    sort(arr, arr + n, compare);

    // Find max deadline
    int maxDeadline = 0;
    for (int i = 0; i < n; i++)
        maxDeadline = max(maxDeadline, arr[i].deadline);

    // Create slots
    int slot[maxDeadline + 1];
    char result[maxDeadline + 1];

    for (int i = 0; i <= maxDeadline; i++)
        slot[i] = -1; // -1 means empty slot

    int totalProfit = 0;
```

![Marwadi University logo]	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Design and Analysis of Algorithm	**Aim: :** Implementing application-based algorithms using Greedy Programming Approach
Experiment No: 07	**Date: 13/09/2025**	**Enrollment No: 92301733049**

```cpp
    // Schedule jobs
    for (int i = 0; i < n; i++) {
        for (int j = arr[i].deadline; j > 0; j--) {
            if (slot[j] == -1) { // If slot empty
                slot[j] = i;
                result[j] = arr[i].id;
                totalProfit += arr[i].profit;
                break;
            }
        }
    }

    cout << "Scheduled Jobs: ";
    for (int i = 1; i <= maxDeadline; i++) {
        if (slot[i] != -1)
            cout << result[i] << " ";
    }
    cout << "\nTotal Profit: " << totalProfit << endl;
}

int main() {
    Job arr[] = { {'a', 2, 100},
            {'b', 1, 19},
            {'c', 2, 27},
            {'d', 1, 25},
            {'e', 3, 15} };

    int n = sizeof(arr) / sizeof(arr[0]);

    jobScheduling(arr, n);

    return 0;
}
```

Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**
Subject: Design and Analysis of Algorithm	**Aim:** : Implementing application-based algorithms using Greedy Programming Approach
Experiment No: 07	**Date: 13/09/2025**	**Enrollment No: 92301733049**

II. Activity Selection Problem

```cpp
#include <iostream>
#include <algorithm>
using namespace std;

struct Activity {
   int start, finish;
};

// Comparator: sort activities by finish time
bool compare(Activity a, Activity b) {
   return a.finish < b.finish;
}

void activitySelection(Activity arr[], int n) {
   // Sort by finish time
   sort(arr, arr + n, compare);

   cout << "Selected Activities: ";

   // First activity is always selected
   int i = 0;
   cout << "(" << arr[i].start << "," << arr[i].finish << ") ";
```
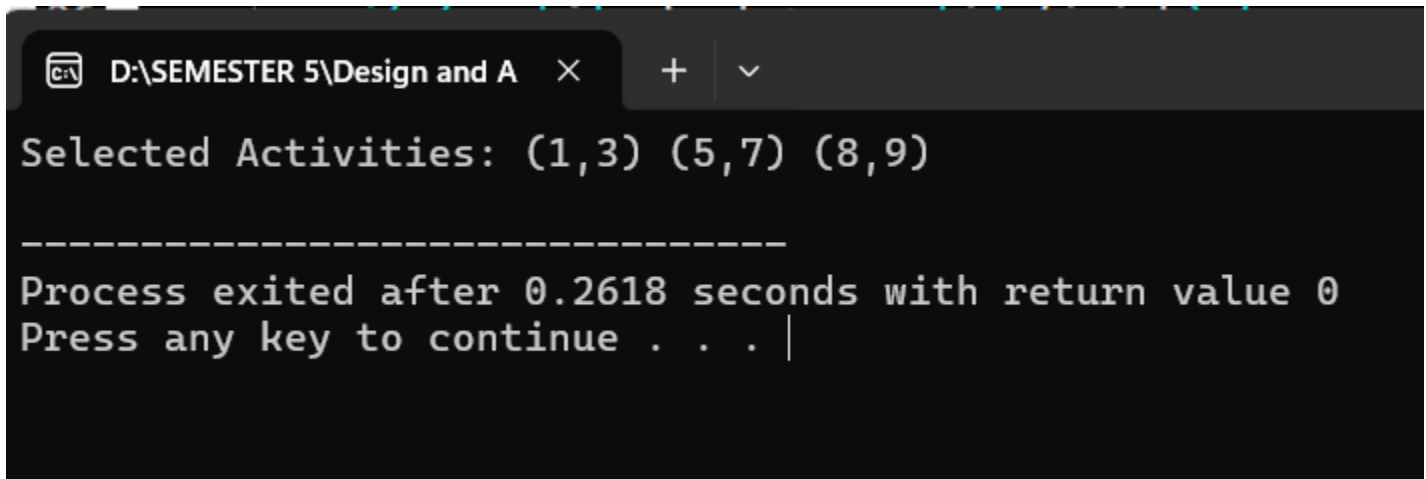
Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering and Technology** **Department of Information and Communication Technology**	
Subject: Design and Analysis of Algorithm	**Aim: : Implementing application-based algorithms using Greedy Programming Approach**	
Experiment No: 07	**Date: 13/09/2025**	**Enrollment No: 92301733049**

```cpp
    // Consider rest of the activities
    for (int j = 1; j < n; j++) {
        if (arr[j].start >= arr[i].finish) {
            cout << "(" << arr[j].start << "," << arr[j].finish << ") ";
            i = j;
        }
    }
    cout << endl;
}

int main() {
    Activity arr[] = { {1, 3}, {2, 5}, {0, 6},
                {5, 7}, {8, 9}, {5, 9} };
    int n = sizeof(arr) / sizeof(arr[0]);

    activitySelection(arr, n);

    return 0;
}
```



```
Selected Activities: (1,3) (5,7) (8,9)

---------------------------------

Process exited after 0.2618 seconds with return value 0
Press any key to continue . . .
```