Best Version: CPU Scheduling Algorithms in C

Aim

To create C programs for the different scheduling algorithms.

To Perform

Create and execute C programs for following CPU Scheduling Algorithms:
1. First Come First Serve (FCFS)
2. Shortest Job First (SJF)
3. Round Robin Scheduling

1. First Come First Serve (FCFS)

```
#include <stdio.h>
int main() {
    int n;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int bt[n], wt[n], tat[n];
    printf("Enter burst times:\n");
    for(int i = 0; i < n; i++) scanf("%d", &bt[i]);

    wt[0] = 0;
    for(int i = 1; i < n; i++)
        wt[i] = wt[i-1] + bt[i-1];

    for(int i = 0; i < n; i++)
        tat[i] = wt[i] + bt[i];

    printf("Process\tBT\tWT\tTAT\n");
    for(int i = 0; i < n; i++)
        printf("%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);
    return 0;
}
```


2. Shortest Job First (SJF)

```
#include <stdio.h>
void sort(int n, int bt[], int p[]) {
    for(int i = 0; i < n - 1; i++) {
        for(int j = i + 1; j < n; j++) {
            if(bt[j] < bt[i]) {
                int temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;
                temp = p[i]; p[i] = p[j]; p[j] = temp;
            }
        }
    }
}

int main() {
    int n;
```

```
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int bt[n], wt[n], tat[n], p[n];

    printf("Enter burst times:\n");
    for(int i = 0; i < n; i++) {
        scanf("%d", &bt[i]);
        p[i] = i+1;
    }

    sort(n, bt, p);

    wt[0] = 0;
    for(int i = 1; i < n; i++)
        wt[i] = wt[i-1] + bt[i-1];

    for(int i = 0; i < n; i++)
        tat[i] = wt[i] + bt[i];

    printf("Process\tBT\tWT\tTAT\n");
    for(int i = 0; i < n; i++)
        printf("%d\t%d\t%d\t%d\n", p[i], bt[i], wt[i], tat[i]);

    return 0;
}
```

3. Round Robin Scheduling


```
#include <stdio.h>
int main() {
    int n, tq;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int bt[n], rt[n], wt[n], tat[n], ct[n], i;
    printf("Enter burst times:\n");
    for(i = 0; i < n; i++) {
        scanf("%d", &bt[i]);
        rt[i] = bt[i];
        wt[i] = 0;
        ct[i] = 0;
    }
    printf("Enter time quantum: ");
    scanf("%d", &tq);

    int time = 0, done;
    do {
        done = 1;
        for(i = 0; i < n; i++) {
            if(rt[i] > 0) {
                done = 0;
                if(rt[i] > tq) {
                    time += tq;
                    rt[i] -= tq;
                } else {
```

```
                time += rt[i];
                ct[i] = time;
                rt[i] = 0;
            }
        }
    }
    } while(!done);

    for(i = 0; i < n; i++) {
        tat[i] = ct[i];
        wt[i] = tat[i] - bt[i];
    }

    printf("Process\tBT\tWT\tTAT\n");
    for(i = 0; i < n; i++)
        printf("%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);

    return 0;
}
```