**Practical 1:** Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).

**ROUND ROBIN:**

#include<stdio.h>

int main() {

int i, j, n, bu[10], wa[10], tat[10], t, ct[10], max;

float awt = 0, att = 0, temp = 0;

printf("\nEnter the number of processes: ");

scanf("%d", &n);

// Input burst times for each process

for(i = 0; i < n; i++) {

printf("Enter Burst Time for Process %d: ", i + 1);

scanf("%d", &bu[i]);

ct[i] = bu[i]; // Copy burst time to ct array for later calculations

}

printf("Enter the size of time slice: ");

scanf("%d", &t);

max = bu[0];

for(i = 1; i < n; i++) {

if(max < bu[i])

max = bu[i];

}

// Round Robin Scheduling logic

for(j = 0; j < (max / t) + 1; j++) {

for(i = 0; i < n; i++) {

if(bu[i] != 0) {

if(bu[i] <= t) {

tat[i] = temp + bu[i];

temp = temp + bu[i];

bu[i] = 0;

} else {

bu[i] = bu[i] - t;

temp = temp + t;

}

}

}

}

// Calculate waiting time and turnaround time for each process

for(i = 0; i < n; i++) {

wa[i] = tat[i] - ct[i];

att += tat[i];

awt += wa[i];

}

// Print Average Turnaround Time and Average Waiting Time

printf("\nAverage Turnaround Time: %.2f", att / n);

printf("\nAverage Waiting Time: %.2f\n", awt / n);

// Display process details

printf("\nPROCESS\t BURST TIME \t WAITING TIME\t TURNAROUND TIME\n");

for(i = 0; i < n; i++) {

printf("%d\t %d\t\t %d\t\t %d\n", i + 1, ct[i], wa[i], tat[i]);

}

return 0;

}

**Output:**

/tmp/cLPP1KUzvE.o

Onkar lonsane Roll No = 28

Enter the number of processes: 2

Enter Burst Time for Process 1: 4

Enter Burst Time for Process 2: 6

Enter the size of time slice: 2

Average Turnaround Time: 8.00

Average Waiting Time: 3.00

PROCESS BURST TIME WAITING TIME TURNAROUND TIME

1 4 2 6

2 6 4 10

=== Code Execution Successful ===