**Sortest Job First(SJF):**

#include<iostream>

#include<algorithm>

using namespace std;

struct Process {

int at; // Arrival time

int bt; // Burst time

int ct; // Completion time

int tat; // Turnaround time

int wt; // Waiting time

};

bool compareBurstTime(const Process &a, const Process &b) {

return a.bt < b.bt;

}

int main()

{

int p;

float avgt = 0, avgw = 0;

cout << "Enter the number of processes: ";

cin >> p;

Process processes[20];

cout << "Enter the arrival time for " << p << " processes:\n";

for (int i = 0; i < p; i++) {

cin >> processes[i].at;

}

cout << "Enter the burst time for " << p << " processes:\n";

for (int i = 0; i < p; i++) {

cin >> processes[i].bt;

}

// Sort the processes based on their burst times in ascending order (SJF)

sort(processes, processes + p, compareBurstTime);

// Calculate completion time for each process

int currentTime = processes[0].at;

for (int i = 0; i < p; i++) {

currentTime += processes[i].bt;

processes[i].ct = currentTime;

}

// Calculate turnaround time and waiting time for each process

cout << "Turnaround time is:\n";

for (int i = 0; i < p; i++) {

processes[i].tat = processes[i].ct - processes[i].at;

cout << processes[i].tat << "\n";

}

cout << "Waiting time is:\n";

for (int i = 0; i < p; i++) {

processes[i].wt = processes[i].tat - processes[i].bt;

cout << processes[i].wt << "\n";

}

// Print tabular format for SJF

cout << "Tabular format for SJF is:\n";

cout << "| Process Number | Arrival Time | Burst Time | Turnaround Time | Waiting Time |\n";

for (int i = 0; i < p; i++) {

cout << "| P" << i + 1 << "\t\t| " << processes[i].at << "\t\t| " << processes[i].bt

<< "\t\t| " << processes[i].tat << "\t\t\t| " << processes[i].wt << "\t\t|\n";

}

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

// Calculate average turnaround time

for (int i = 0; i < p; i++) {

avgt += processes[i].tat;

}

cout << "\nAverage of turnaround time is: " << avgt / p << "\n";

// Calculate average waiting time

for (int i = 0; i < p; i++) {

avgw += processes[i].wt;

}

cout << "Average of waiting time is: " << avgw / p << "\n\n";

}

**OUTPUT:**

Enter the number of processes: 4

Enter the arrival time for 4 processes:

1 2 1 4

Enter the burst time for 4 processes:

3 4 2 4

Turnaround time is:

2

5

8

10

Waiting time is:

0

2

4

6

Tabular format for SJF is:

| Process Number | Arrival Time | Burst Time | Turnaround Time | Waiting Time |

| P1 | 1 | 2 | 2 | 0 |

| P2 | 1 | 3 | 5 | 2 |

| P3 | 2 | 4 | 8 | 4 |

| P4 | 4 | 4 | 10 | 6 |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Average of turnaround time is: 6.25

Average of waiting time is: 3