FAIZAN CHOUDHARY

20BCS021

OS LAB

27th January 2022

CODE: (code pasted in this format for readability)

```
#include <iostream>
#include <string.h>
using namespace std;

struct node
{
    char n[10];
    int burst;
    int arrival;
    int completion;
    int waiting;
    int turnaround;
    int response;
    struct node *next;
};
struct node *front=NULL, *p, *ptr, *temp;

bool isEmpty () {
    if (front==NULL)
        return true;
    else
        return false;
}

void insertProcess (char *pr, int bt, int at) {
    ptr = (struct node *) malloc (sizeof(struct node));
    if (ptr == NULL) {
        cout<<"\nMemory could not be allocated!\n";
        return;
    }

    strcpy(ptr->n, pr);
    ptr->burst = bt;
    ptr->arrival = at;
    ptr->next=NULL;

    if (front == NULL || at < (front->arrival)) {
        ptr->next = front;
        front=ptr;
    }
    else {
        p=front;
```

```
        while (p->next != NULL && p->next->arrival <= at)
            p=p->next;
        ptr->next = p->next;
        p->next = ptr;
    }
}

void FCFS () {
    int time = 0;
    p = front;
    while (p != NULL) {
        if (time < p->arrival) {
            while (time != p->arrival)
                time++;
        }
        p->response = time - p->arrival;
        time += p->burst;
        p->completion = time;   // completion occurs after burst time ends
        p->turnaround = p->completion - p->arrival;     // tat = ct - at = wt + bt
        p->waiting = p->turnaround - p->burst;          // wt = tat - bt
        p = p->next;
    }
}

void display () {
    double tot_ct = 0, tot_wt =0, tot_tat = 0, tot_rt =0;
    int count = 0;
    p = front;
    cout<<"\n\nProcess | Burst Time | Arrival Time | Completion Time | Waiting Time |
Turnaround Time | Response Time\n";
    cout<<"_____________________________________________________________________________
_____________________\n\n";
    while (p != NULL) {
        printf("   %s        %2d            %2d                %2d               %2d
     %2d              %2d\n", p->n, p->burst, p->arrival, p->completion, p->waiting, p-
>turnaround, p->response);

        tot_ct += p->completion;
        tot_wt += p->waiting;
        tot_tat += p->turnaround;
        tot_rt += p->response;

        count++;
        p = p->next;
    }
    cout<<"_____________________________________________________________________________
_____________________\n\n";

    printf("\nAverage Completion time: %.2f",tot_ct / (float) count);
    printf("\nAverage Waiting time: %.2f", tot_wt / (float) count);
    printf("\nAverage Turnaround time: %.2f",tot_tat / (float) count);
    printf("\nAverage Response time: %.2f\n",tot_rt / (float) count);
}
```

```
void displayGantt () {
    int time = 0;
    p = front;
    cout<<"\nGantt chart: \n";
    // for printing structure
    while (p != NULL) {
        cout<<"|";
        if (time < p->arrival) {
            while (time != p->arrival) {
                time++;
            }
            time += p->burst;
            cout<<"  |";
        }
        else {
            time += p->arrival;
            if (front->arrival == 0)
                time += p->burst;
        }
        for (int i=0; i<(p->burst-1); i++)
            cout<<" ";
        cout<<p->n;
        for (int i=0; i<(p->burst-1); i++)
            cout<<" ";
        p = p->next;
    }
    cout<<"|"<<endl;
    p = front;
    time = 0;
    // for printing time below each process
    if (time < p->arrival && p->arrival != 0) {
        cout<<time;
        while (time != p->arrival) {
            time++;
        }
        time += p->burst;
        cout<<"  ";
    }
    cout<<p->arrival;
    while (p != NULL) {
        if (time < p->arrival) {
            while (time != p->arrival) {
                time++;
            }
            if (time < 9)
                cout<<"  "<<time;
            else
                cout<<" "<<time;
            time += p->burst;
        }
        else {
            time += p->arrival;
            if (front->arrival == 0)
                time += p->burst;
```

```
        }
        for (int i=0; i< 2*(p->burst)-1; i++)
            cout<<" ";
        if (p->completion < 9)
            cout<<" "<<p->completion;
        else
            cout<<p->completion;
        p = p->next;
    }
    cout<<endl<<endl;
}

void del () {
    p = front;
    front=front->next;
    delete p;
}

int main () {
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";
    cout<<"\nFirst Come First Serve Scheduling Algorithm\n";
    int n;

    cout<<"\nEnter the number of processes: ";
    cin>>n;
    char k[n][10];
    int bt[n], at[n];                   // burst time and arrival time

    cout<<"\nEnter process names: ";
    for (int i=0; i<n; i++)
        cin>>k[i];
    cout<<"\nEnter burst time for each process: ";
    for (int i=0; i<n; i++)
        cin>>bt[i];
    cout<<"\nEnter arrival time for each process: ";
    for (int i=0; i<n; i++)
        cin>>at[i];

    for (int i=0; i<n; i++)
        insertProcess(k[i],bt[i],at[i]);

    FCFS ();            // logic for calculating various times
    display ();         // displaying calculated values of time
    displayGantt ();    // to display Gantt chart
    del ();             // releasing memory

    return 0;
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

First Come First Serve Scheduling Algorithm

Enter the number of processes: 3

Enter process names: p1 p2 p3

Enter burst time for each process: 2 1 6

Enter arrival time for each process: 0 3 5
```

```
Process | Burst Time | Arrival Time | Completion Time | Waiting Time | Turnaround Time | Response Time
--------------------------------------------------------------------------------------------------

   p1         2            0              2              0              2              0
   p2         1            3              4              0              1              0
   p3         6            5             11              0              6              0
--------------------------------------------------------------------------------------------------


Average Completion time: 5.67
Average Waiting time: 0.00
Average Turnaround time: 3.00
Average Response time: 0.00

Gantt chart:
| p1 |  |p2|  |     p3     |
0    2  3  4  5           11
```

```
FAIZAN CHOUDHARY
20BCS021

First Come First Serve Scheduling Algorithm

Enter the number of processes: 5

Enter process names: p1 p2 p3 p4 p5

Enter burst time for each process: 6 2 8 3 4

Enter arrival time for each process: 2 5 1 0 4
```

```
Process | Burst Time | Arrival Time | Completion Time | Waiting Time | Turnaround Time | Response Time
--------------------------------------------------------------------------------------------------------

   p4        3            0              3              0             3              0
   p3        8            1             11              2            10              2
   p1        6            2             17              9            15              9
   p5        4            4             21             13            17             13
   p2        2            5             23             16            18             16
--------------------------------------------------------------------------------------------------------


Average Completion time: 15.00
Average Waiting time: 8.00
Average Turnaround time: 12.60
Average Response time: 8.00

Gantt chart:
|  p4  |      p3       |     p1     |   p5   | p2 |
0      3               11           17       21   23
```

```
FAIZAN CHOUDHARY
20BCS021

First Come First Serve Scheduling Algorithm

Enter the number of processes: 6

Enter process names: p1 p2 p3 p4 p5 p6

Enter burst time for each process: 3 1 2 1 2 3

Enter arrival time for each process: 5 7 6 1 1 8
```

```
Process | Burst Time | Arrival Time | Completion Time | Waiting Time | Turnaround Time | Response Time
--------------------------------------------------------------------------------------------------------

   p4        1            1              2              0             1              0
   p5        2            1              4              1             3              1
   p1        3            5              8              0             3              0
   p3        2            6             10              2             4              2
   p2        1            7             11              3             4              3
   p6        3            8             14              3             6              3
--------------------------------------------------------------------------------------------------------


Average Completion time: 8.17
Average Waiting time: 1.50
Average Turnaround time: 3.50
Average Response time: 1.50

Gantt chart:
|  |p4| p5 |  |  p1  | p3 |p2|  p6  |
0  1  2    4  5      8   10 11     14
```