

Roll No: 20 MCA060

Practical :- 5

Q) FCFS :

→ #include <iostream>  
using namespace std;  
struct process

{

string process\_name;  
int AT; // arrival time  
int BT; // Burst time  
int ST; // Start time  
int ET; // end time  
int TAT; // Turn Around Time  
float ratio; // TAT/BT  
void input()

{

cout << "Enter Process Name : ";  
cin >> process\_name;  
cout << "Enter Arrival Time : ";  
cin >> AT;  
cout << "Enter Burst Time : ";  
cin >> BT;

}

void output()

{

$$TAT = ET - AT;$$

$$ratio = (float) TAT / BT;$$

cout << process\_name << "I" << AT << "J" << BT << "I" << ST << "J" << ET << "A" << TAT << "D" << ratio << endl;

}

}

int main()

{

int n;

cout << "Enter Number of Process : ";

cin >> n;

process p[n];

// input data;

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

{

p[i].input();

}

// sort according to Arrival Time

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

{

for (int j=i+1; j<n; j++)

{

if (p[i].AT > p[j].AT)

{

process temp = p[i];

p[i] = p[j];

p[j] = temp;

}

}

// calculate start & End time

int start = p[0].AT + p[0].BT;

p[0].ST = p[0].AT;

int end = p[0].AT + p[0].BT;  
p[0].ET = end;

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

    p[i].ST = start;  
    start = start + p[i].BT;  
    end = end + p[i].BT;  
    p[i].ET = end;

}

//sort name vice

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

    for (int j=i+1; j<n; j++)  
    {

        if (p[i].process\_name > p[j].process\_name)

{

    process temp = p[i];  
    p[i] = p[j];  
    p[j] = temp;

}

}

:

int sum\_TAT = 0;

float sum\_ratio = 0;

cout << "In Process Id AT BT ST ET TAT TQ/TS In";

cout << "-----m";

```
for (int i = 0; i < n; i++)
```

{

```
    p[i].output();
```

```
    sum_tat = sum_tat + p[i].TAT;
```

```
    sum_ratio = sum_ratio + p[i].ratio;
```

{

```
cout << "Sum of TAT : " << sum_tat;
```

```
cout << "Avg of TAT : " << (float)sum_tat/n;
```

```
cout << "Sum of TG/TS : " << sum_ratio;
```

```
cout << "Avg of TG/TS : " << (float)sum_ratio/n;
```

```
return 0;
```

{

## \* SJF in

```
→ #include <iostream>
using namespace std;
struct process
{
    string process_name;
    int AT; // arrival time
    int BT; // burst time
    int BT_ORI; //
    int ST; // start time
    int ET; // end time
    int TAT; // turn around time
    float ratio; // TAT/BT
```

void input()

{

```
cout << "Enter Process Name : ";
cin >> process_name;
cout << "Arrival time : ";
cin >> AT;
cout >> << "Burst time : ";
cin >> BT;
BT_ORI = BT;
```

}

void output()

{ TAT = ET - AT;

ratio = (float) TAT / BT\_ORI;

```
cout << process_name << " " << AT << " " <<
```

```
BT_ORI << " " << ST << " " << ET << " " << TAT << " " <<
```

```
ratio << endl;
}
};

int check(process p[], int n, int t)
{
    int min = 0;
    for(int i=0; i<n; i++)
    {
        if(p[min].BT == 0)
        {
            min = i;
        }
        else if(t >= p[i].AT && p[i].BT < p[min].BT && p[i].BT != 0)
        {
            min = i;
        }
    }
    return min;
}

int main()
{
    int n;
    cout << "Enter Number of Process : ";
    cin >> n;
    process p[n];
    //input data
    for(int i=0; i<n; i++)
    {
        p[i].input();
    }
    int t = 0; // calculate start & end time
```

```
for (int i=0; i<n; i++)
```

```
    int min = check(p, n, t);
```

```
    if (t >= p[min].AT)
```

```
{
```

```
    p[min].ST = t;
```

```
    t = t + p[min].BT;
```

```
    p[min].ET = t;
```

```
    p[min].BT = 0;
```

```
}
```

```
else
```

```
    t++;
```

```
}
```

```
int sum_tat = 0;
```

```
float sum_ratio = 0;
```

```
cout << "In Process (J AT) (BT) (ST) (ET) (TAT)  
      (TG/TB) \n";
```

```
cout << " - - - - - - - - - \n";
```

```
for (int i=0; i<n; i++)
```

```
{
```

```
    p[i].output();
```

```
    sum_tat = sum_tat + p[i].TAT;
```

```
    sum_ratio = sum_ratio + p[i].ratio;
```

```
}
```

```
cout << "Sum of TAT : " << sum_tat;
```

```
cout << "AVG : " << (float) sum_tat / n;
```

```
cout << "Sum of TAT/BT : " << sum_ratio;
```

```
cout << "AVG : " << sum_ratio / n;
```

```
>>show();
```

```
}
```

★ RR :

```
#include <iostream>
using namespace std;
struct process {
    string process_name;
    int AT; // arrival Time
    int BT; // burst Time
    int BT_ORI;
    int ST; // start Time
    int ET; // end Time
    int TAT; // turn around Time
    float ratio; // TAT/BT
    void input()
    {
        cout << "Enter Process Name : ";
        cin >> process_name;
        cout << "Arrival Time : ";
        cin >> AT;
        cout << "Burst Time : ";
        cin >> BT;
        BT_ORI = BT;
    }
    void output()
    {
        TAT = ET - AT;
        ratio = (float)TAT / BT_ORI;
        cout << process_name << " " << AT << " " <<
        BT_ORI << " " << ST << " " << ET << " " << TAT << " " << ratio;
    }
};
```

```
};

bool check (process p[], int n)
{
    bool flag = false;
    for (int i=0; i<n; i++)
    {
        if (p[i].BT > 0)
            flag = true;
        break;
    }
    return flag;
}
```

```
int main()
{
    int n;
    cout << "Enter Number of Process : ";
    cin >> n;
    process p[n];
    // input data
    for (int i=0; i<n; i++)
    {
        p[i].input();
    }
    // sort according to Arrival Time
    for (int i=0; i<n; i++)
    {

```

```
for (int j = i+1; j < n; j++)  
{  
    if (p[i].AT > p[j].AT)  
    {  
        process temp = p[i];  
        p[i] = p[j];  
        p[j] = temp;  
    }  
}  
int q;  
cout << "Enter Quantum Time : ";  
cin >> q;  
// calculate start & end time  
int t = p[0].AT;  
int loop = 0;  
while (check(p, n))  
{  
    loop++;  
    for (int i = 0; i < n; i++)  
    {  
        if (t >= p[i].AT)  
        {  
            if (loop == 1)  
                p[i].ST = t;  
            if (p[i].BT <= q && p[i].BD)  
            {  
                t = t + p[i].BT;  
                p[i].ET = t;  
                p[i].BT = 0;  
            }  
        }  
    }  
}
```

else if ( $p[i].BT > q$ )

{

$p[i].BT = p[i].BT - q;$

$t = t + q;$

}

}  
else  
{

$t = t + q;$

}

}

// Sort name vice  
for (int i=0; i<n; i++)  
{

for (int j=i+1; j<n; j++)

{

if ( $p[i].process\_name > p[j].process\_name$ )

{

process temp =  $p[i];$

$p[i] = p[j];$

$p[j] = temp;$

}

}

int sum\_tat = 0;

float sum\_ratio = 0;

cout << "In Process \t TAT \t BT \t SFT \t "

TAT \t TAT/BT \n";

```
cout << "-----\n";
```

```
for (int i=0; i<n; i++)
```

```
{
```

```
    p[i].output();
```

```
    sum_tat = sum_tat + p[i].TAT;
```

```
    sum_ratio = sum_ratio + p[i].ratio;
```

```
}
```

```
cout << "Sum of TAT : " << sum_tat;
```

```
cout << "AVG : " << (float)sum_tat / n;
```

```
cout << "Sum of TAT/BT : " << sum_ratio;
```

```
cout << "AVG : " << sum_ratio / n;
```

```
return 0;
```

```
}
```