1. FCFS PROGRAM

```cpp
#include<iostream>

using namespace std;

int main()

{ int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;

cout<<"Enter total number of processes(maximum 20):";

cin>>n;

cout<<"\nEnter Process Burst Time aka DURATION \n";

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

{

cout<<"P["<<i+1<<"]:";

cin>>bt[i];

}

wt[0]=0; //waiting time for first process is 0

//calculating waiting time

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

{

wt[i]=0;

for(j=0;j<i;j++)

wt[i]+=bt[j];

}

cout<<"\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time";

//calculating turnaround time

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

```cpp
{

tat[i]=bt[i]+wt[i];

avwt+=wt[i];

avtat+=tat[i];

cout<<"\nP["<<i+1<<"]"<<"\t\t"<<bt[i]<<"\t\t"<<wt[i]<<"\t\t"<<tat[i];

}

avwt/=i;

avtat/=i;

cout<<"\n\nAverage Waiting Time:"<<avwt;

cout<<"\nAverage Turnaround Time:"<<avtat;

return 0;

}
```

OR

```cpp
//C++ program for implementation of FCFS

// scheduling

#include<iostream>

using namespace std;

// Function to find the waiting time for all

// processes

void findWaitingTime(int processes[], int n,

int bt[], int wt[])

{



// waiting time for first process is 0
```

```cpp
wt[0] = 0;

// calculating waiting time

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

wt[i] = bt[i-1] + wt[i-1] ;

}

// Function to calculate turn around time

void findTurnAroundTime( int processes[], int n,

int bt[], int wt[], int tat[])

{

// calculating turnaround time by adding

// bt[i] + wt[i]

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

tat[i] = bt[i] + wt[i];

}

//Function to calculate average time

void findavgTime( int processes[], int n, int bt[])

{

int wt[n], tat[n], total_wt = 0, total_tat = 0;

//Function to find waiting time of all processes

findWaitingTime(processes, n, bt, wt);

//Function to find turn around time for all processes

findTurnAroundTime(processes, n, bt, wt, tat);

//Display processes along with all details

cout << "Processes "<< " Burst time "

<< " Waiting time " << " Turn around time\n";
```

```cpp
// Calculate total waiting time and total turn
// around time
for (int i=0; i<n; i++)
{
total_wt = total_wt + wt[i];
total_tat = total_tat + tat[i];
cout << " " << i+1 << "\t\t" << bt[i] <<"\t "
<< wt[i] <<"\t\t " << tat[i] <<endl;
}
cout << "Average waiting time = "
<< (float)total_wt / (float)n;
cout << "\nAverage turn around time = "
<< (float)total_tat / (float)n;
}
// Driver code
int main()
{

//process id's
int processes[] = { 1, 2, 3};
int n = sizeof processes / sizeof processes[0];
//Burst time of all processes
int burst_time[] = {10, 5, 8};
findavgTime(processes, n, burst_time);
return 0;
```

}

OUTPUT:

Enter total number of processes(maximum 20):5

Enter Process Burst Time aka DURATION

P[1]:4

P[2]:3

P[3]:2

P[4]:1

P[5]:5

Process Burst Time Waiting Time Turnaround Time

P[1] 4 0 4

P[2] 3 4 7

P[3] 2 7 9

P[4] 1 9 10

P[5] 5 10 15

Average Waiting Time:6

Average Turnaround Time:9