NATIONAL INSTITUTE OF TECHNOLOGY MIZORAM

OS LAB ASSIGNMENT

Name: NIRAJ KUMAR

Department: CSE

Enrollment No. : BT19CS031

Implement the Round Robin CPU scheduling algorithm. Read inputs from a file (format given below). Take the time quantum from the user.

Inputs:

The program will read from an input file containing a list of processes along with other data require for scheduling.

The input file will look like this:

P1 0 20.0

P2 2 15.0

P3 6 27.0

p4 4 36.0

- **The file containing the information on the processes will have each process on a separate line. The processes will be in the file in the order in which they arrive at the OS.**
- **Each line will have a process name that will be a string.**

- **Following the name will be the arrival time of the process**
- **Following arrival time will be the total burst time.**

Outputs:

The program must print out the time taken by each process to complete (turnaround time) and the wait time and compute the average turnaround time for all processes.

```
#include<stdio.h>
#include<stdlib.h>
#include<iostream>
#include<string>
#include <cstdlib>
#include <fstream>
#include <sstream>
#include <algorithm>
#include <iomanip>
#include<vector>
using namespace std;

int n; // n denotes total no. of process , we are keeping
          it global variable

// Process class containing info about the process
```

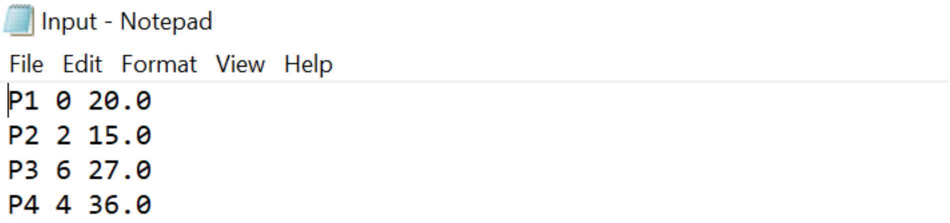
```
class Process
{
private:
string name; //Name of process
int process_id;
int arrival_time;
int burst_time;
int rbt;      //Remaining burst time
int completion_time;
int turn_around_time;
int wait_time;
int flag;    //to check if the process is inside ready
queue or not

public:

//constructor to initialise member variables

Process(string name,int arrival_time,int burst_time,int
process_id)
{
this->name = name;
this->arrival_time = arrival_time;
this->burst_time = burst_time;
this->process_id = process_id;
this->rbt = burst_time;
this->completion_time = 0;
this->turn_around_time = 0;
this->wait_time = 0;
this->flag = 0;
}


int getTurnAroundTime()
{
    return turn_around_time;
}

int getWaitTime()
{
    return wait_time;
}
```

```
int getArrivalTime()
{
    return arrival_time;
}

int getProcessID()
{
    return process_id;
}

void setTurnAroundTime()
{
 turn_around_time = completion_time - arrival_time;
}

void setWaitTime()
{
 wait_time = turn_around_time - burst_time;
}

void displayDetails()
{
cout << name << "\t" << arrival_time << "\t\t" <<
burst_time << "\t\t"
     << completion_time << "\t\t" << turn_around_time <<
"\t\t" <<
        wait_time << endl;
}

// Friend functions

friend void calculateCompletionTime( Process **,int );
friend void updatereadyqueue( Process ** ,
vector<Process*> * , int );

};

//comparator based on arrival time

bool compare1( Process *p1 , Process *p2 )
{
 if(p1->getArrivalTime() != p2->getArrivalTime())
    return p1->getArrivalTime() < p2->getArrivalTime();
 else
    return p1->getProcessID() < p2->getProcessID();
}

//comparator based on process id

bool compare2(Process *p1, Process *p2)
{
    return p1->getProcessID() < p2->getProcessID();
}

//this functions maintains ready queue on the basis of
arrival time

void updatereadyqueue(Process **p , vector<Process*> *rq ,
int time_counter)
{
 for (int i=0 ; i < n ; i++)
 if(p[i]->arrival_time <= time_counter && p[i]->rbt != 0
&& p[i]->flag == 0 )
 {
  rq->push_back(p[i]);
  p[i]->flag = 1;
 }
}

//Calculating completion time for all the process

void calculateCompletionTime(Process **p,int tq)
{
 int time_counter = 0;
 vector <Process*> ready_queue;

 for(int count = 1 ; count <= n; )
 {
     updatereadyqueue(p , &ready_queue , time_counter);
     if(!ready_queue.empty())
     {
         if(ready_queue[0]->rbt >= tq)
         {
```

```
         // Increase the value of time counter i.e. shows
         // how much time a process has been processed

            time_counter += tq;

        // decreasing the burst time of the processed
process
            ready_queue[0]->rbt -= tq;
         }

         // If burst time is smaller than or equal to
         // quantum. Last cycle for this process
         else
         {
            // Increase the value of time counter by the
amount of
            // burst time remaining
            time_counter += ready_queue[0]->rbt;

            //remaining burst time of process set to zero
            ready_queue[0]->rbt = 0;
         }

         //Context switching part


         if(ready_queue[0]->rbt == 0)
            {
            //current process completed
            count++;
            ready_queue[0]->completion_time =
time_counter;
            ready_queue[0]->setTurnAroundTime();
            ready_queue[0]->setWaitTime();

            // pop out the processed process from
beginning of queue
            //and inserting new process to queue according
to arrival time
            ready_queue.erase(ready_queue.begin());
            updatereadyqueue(p,&ready_queue,time_counter);
            }
         else
            {   //current process still need to be
processed but first we add new
                //process and add the current process to
last of ready queue and pop
                //out the current process from beginning
of queue

updatereadyqueue(p,&ready_queue,time_counter);
                ready_queue.push_back(ready_queue[0]);
                ready_queue.erase(ready_queue.begin());
            }
      }
     else
      // if the queue is empty then we need to add time
gap
      //in our gantt chart
      time_counter++;
}
}


int main()
{

int tq ;
n=4;

cout<< "Enter the value of time quantum = ";
cin>>tq;

// Array to store arrival and burst time for all the
process

int at[n],bt[n];

string lines[4];

//File Handling , taking input from file

ifstream fio;
```

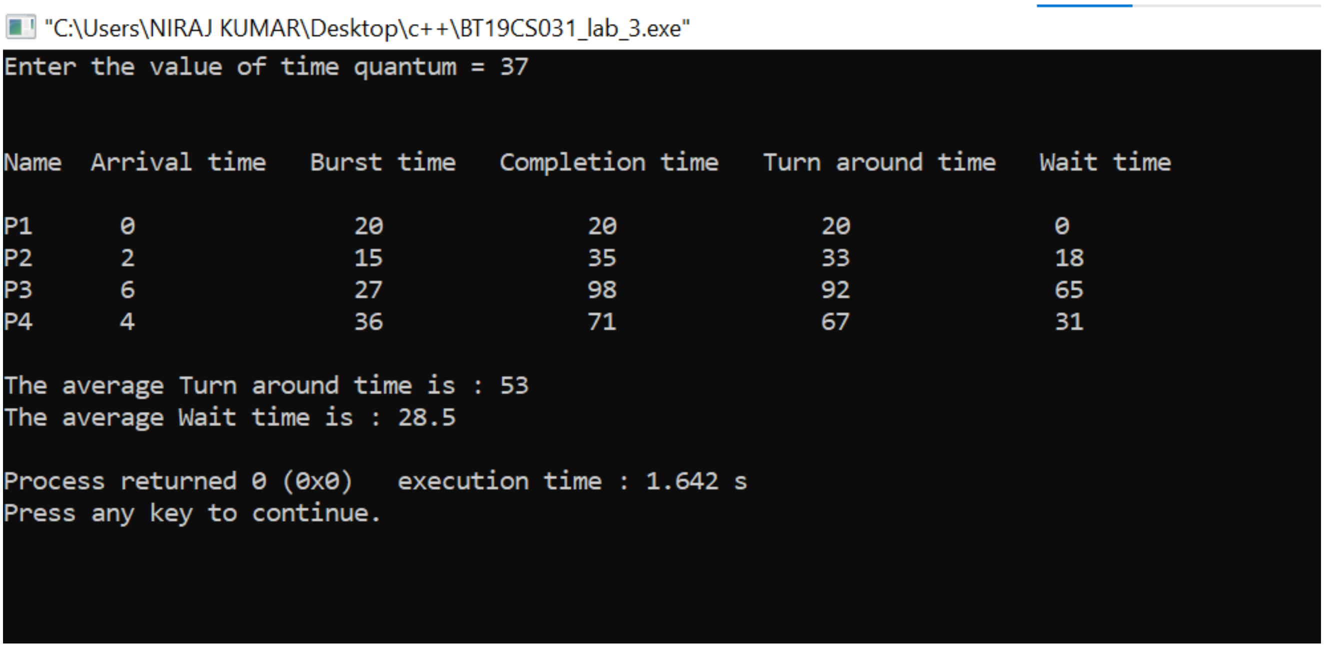
```
fio.open("C:\\Users\\NIRAJ
KUMAR\\Desktop\\c++\\Input.txt");

//from input file we update at[] and bt[] array
// we need to convert string data to integer values

for (int i=0;i<n;i++)
{
getline(fio,lines[i]);
stringstream gk1(lines[i].substr(3,1));
gk1 >> at[i];
stringstream gk2(lines[i].substr(5,4));
gk2 >> bt[i];
}
fio.close();

// Creating objects of Process class And initialising them

Process **p=(Process**)malloc( sizeof(Process*) * n );

p[0]=new Process("P1",at[0],bt[0],1);
p[1]=new Process("P2",at[1],bt[1],2);
p[2]=new Process("P3",at[2],bt[2],3);
p[3]=new Process("P4",at[3],bt[3],4);

//sorting on the basis of arrival time

stable_sort(p,p+n,compare1);

calculateCompletionTime(p,tq);

//sorting on the basis of process id

stable_sort(p,p+n,compare2);

//Displaying Details

cout<<"\n\nName  Arrival time   Burst time   Completion
time   Turn around time   Wait time\n\n";

for (int i=0;i<=n-1;i++)
 p[i]->displayDetails();
 //calculating average wait time and turn around time

double total_wait_time = 0,total_turn_around_time = 0;
for (int i = 0 ; i < n ; i++ )
{
total_turn_around_time += p[i]->getTurnAroundTime();
total_wait_time += p[i]->getWaitTime();
}

cout<<"\nThe average Turn around time is :
"<<total_turn_around_time/n<<endl;
cout<<"The average Wait time is :
"<<total_wait_time/n<<endl;

return 0;
}
```

INPUT FILE -



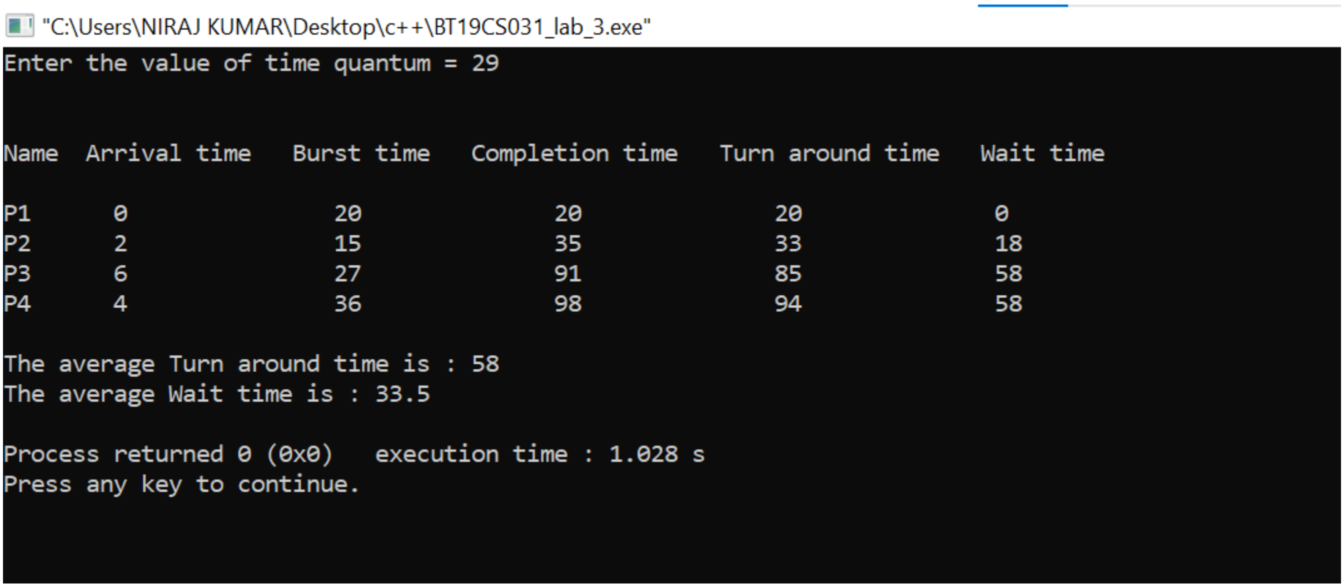
```
P1 0 20.0
P2 2 15.0
P3 6 27.0
P4 4 36.0
```

OUTPUT FOR MULTIPLE VALUES OF TIME QUANTUM



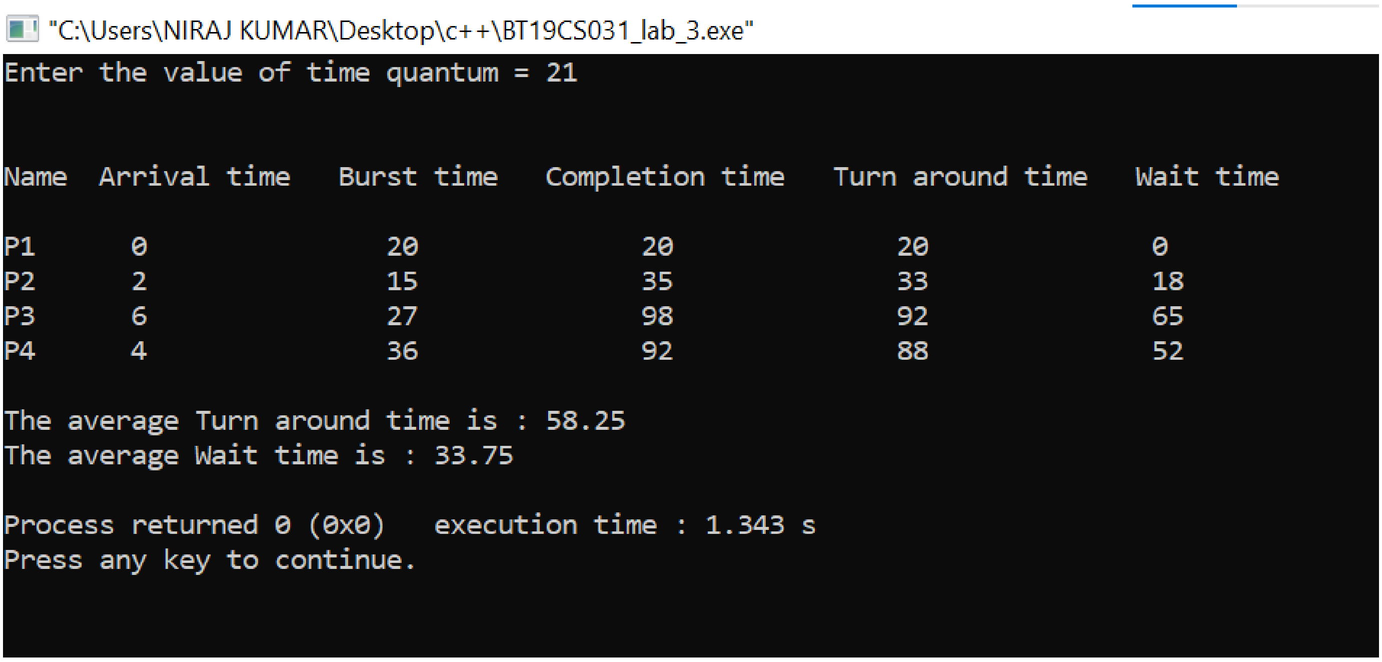
"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 37


Name  Arrival time   Burst time   Completion time   Turn around time   Wait time

P1      0              20            20              20                0
P2      2              15            35              33                18
P3      6              27            98              92                65
P4      4              36            71              67                31

The average Turn around time is : 53
The average Wait time is : 28.5

Process returned 0 (0x0)   execution time : 1.642 s
Press any key to continue.
```



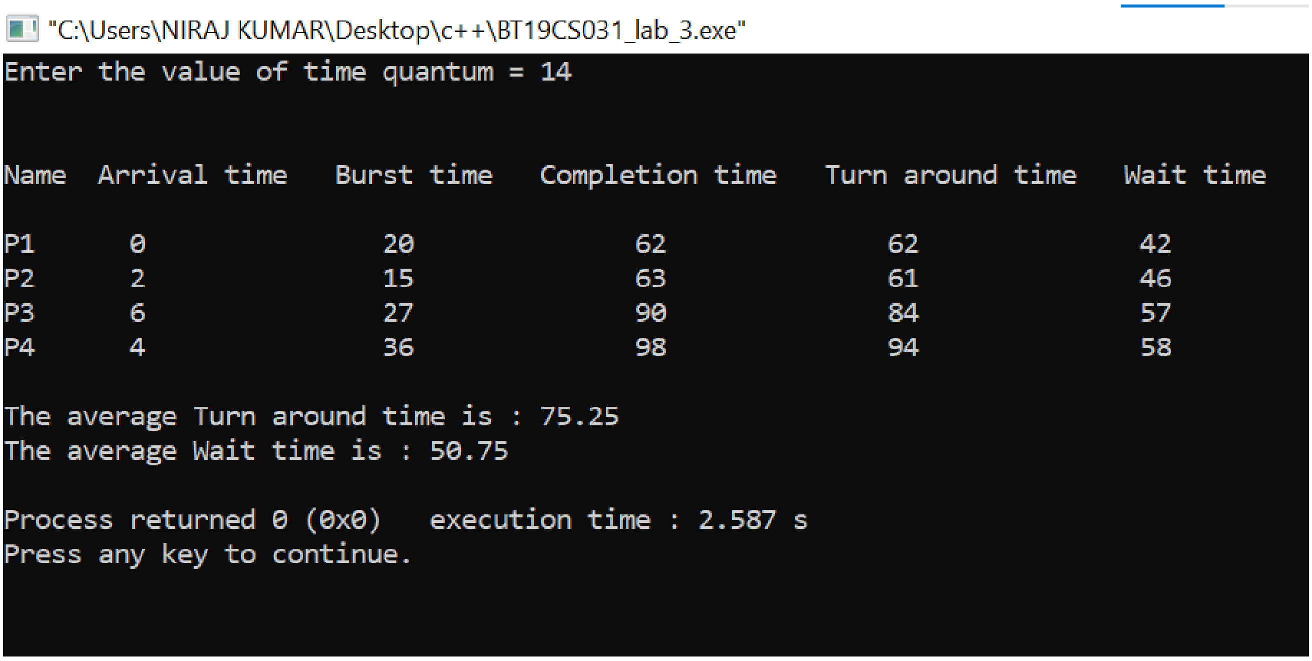
"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 29


Name  Arrival time   Burst time   Completion time   Turn around time   Wait time

P1      0              20            20              20                0
P2      2              15            35              33                18
P3      6              27            91              85                58
P4      4              36            98              94                58

The average Turn around time is : 58
The average Wait time is : 33.5

Process returned 0 (0x0)   execution time : 1.028 s
Press any key to continue.
```



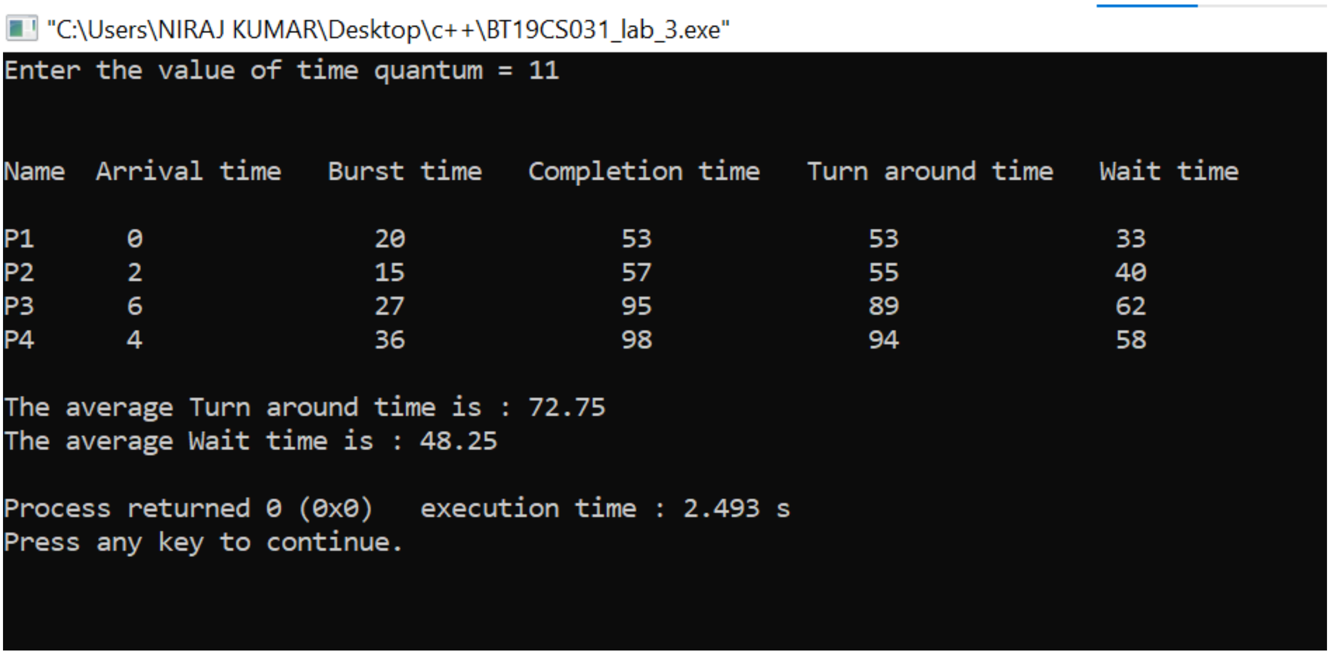
"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 21


Name   Arrival time   Burst time   Completion time   Turn around time   Wait time

P1       0              20            20                20                 0
P2       2              15            35                33                 18
P3       6              27            98                92                 65
P4       4              36            92                88                 52

The average Turn around time is : 58.25
The average Wait time is : 33.75

Process returned 0 (0x0)   execution time : 1.343 s
Press any key to continue.
```



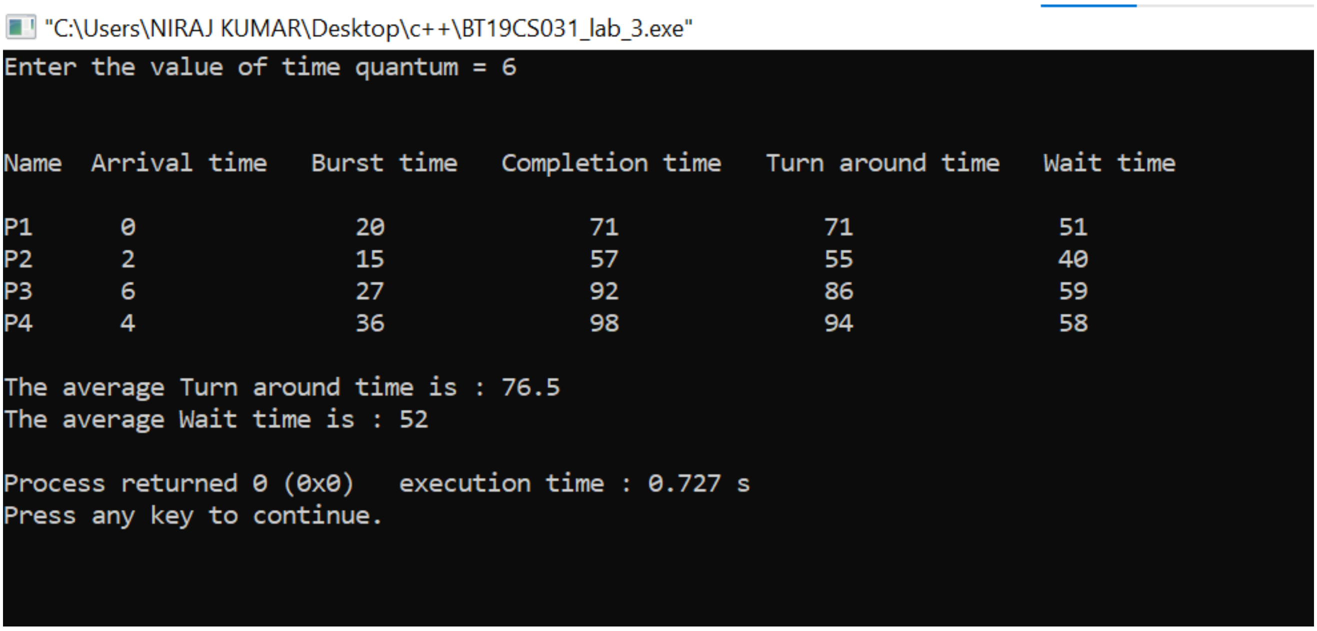
"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 14


Name   Arrival time   Burst time   Completion time   Turn around time   Wait time

P1       0              20            62                62                 42
P2       2              15            63                61                 46
P3       6              27            90                84                 57
P4       4              36            98                94                 58

The average Turn around time is : 75.25
The average Wait time is : 50.75

Process returned 0 (0x0)   execution time : 2.587 s
Press any key to continue.
```



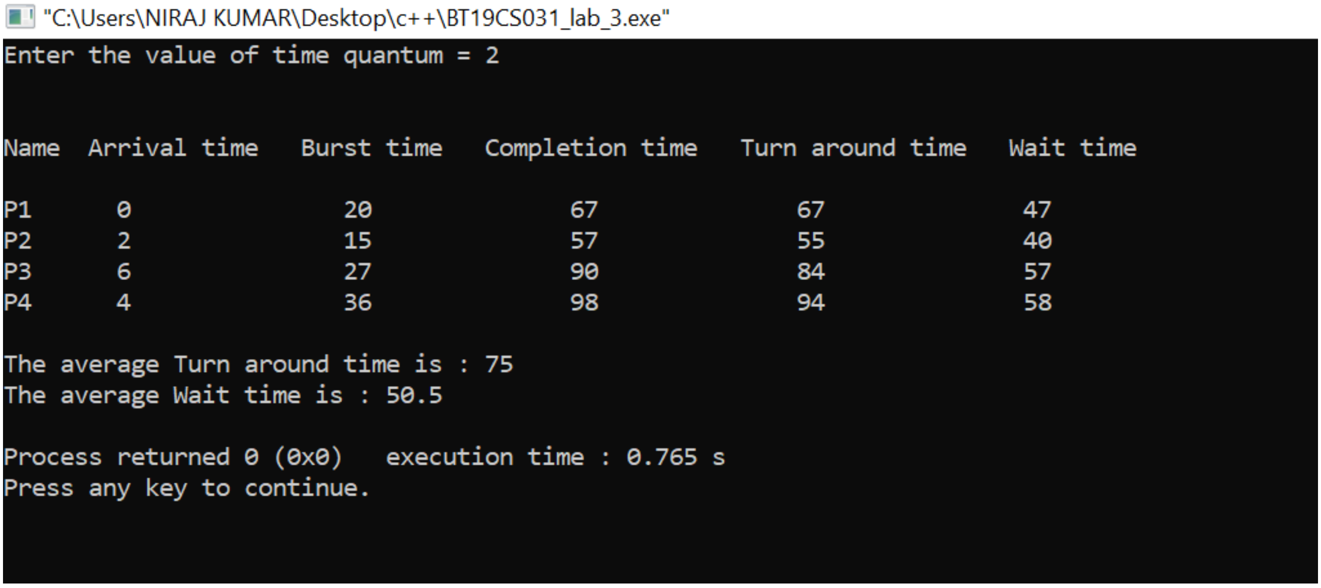
"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 11


Name  Arrival time   Burst time   Completion time   Turn around time   Wait time

P1       0             20             53               53                33
P2       2             15             57               55                40
P3       6             27             95               89                62
P4       4             36             98               94                58

The average Turn around time is : 72.75
The average Wait time is : 48.25

Process returned 0 (0x0)   execution time : 2.493 s
Press any key to continue.
```



"C:\Users\NIRAJ KUMAR\Desktop\c++\BT19CS031_lab_3.exe"

```
Enter the value of time quantum = 6


Name  Arrival time   Burst time   Completion time   Turn around time   Wait time

P1       0             20             71               71                51
P2       2             15             57               55                40
P3       6             27             92               86                59
P4       4             36             98               94                58

The average Turn around time is : 76.5
The average Wait time is : 52

Process returned 0 (0x0)   execution time : 0.727 s
Press any key to continue.
```



```
Enter the value of time quantum = 2


Name  Arrival time   Burst time   Completion time   Turn around time   Wait time

P1      0               20              67              67              47
P2      2               15              57              55              40
P3      6               27              90              84              57
P4      4               36              98              94              58

The average Turn around time is : 75
The average Wait time is : 50.5

Process returned 0 (0x0)   execution time : 0.765 s
Press any key to continue.
```