NAME – Anisha Rawat

COURSE- BSC IT

STUDENT ID – 20422001

SUBJECT : OPERATING SYSTEM

IMPLEMENTATION OF FCFS SCHEDULING ALGORITHM

PROGRAM

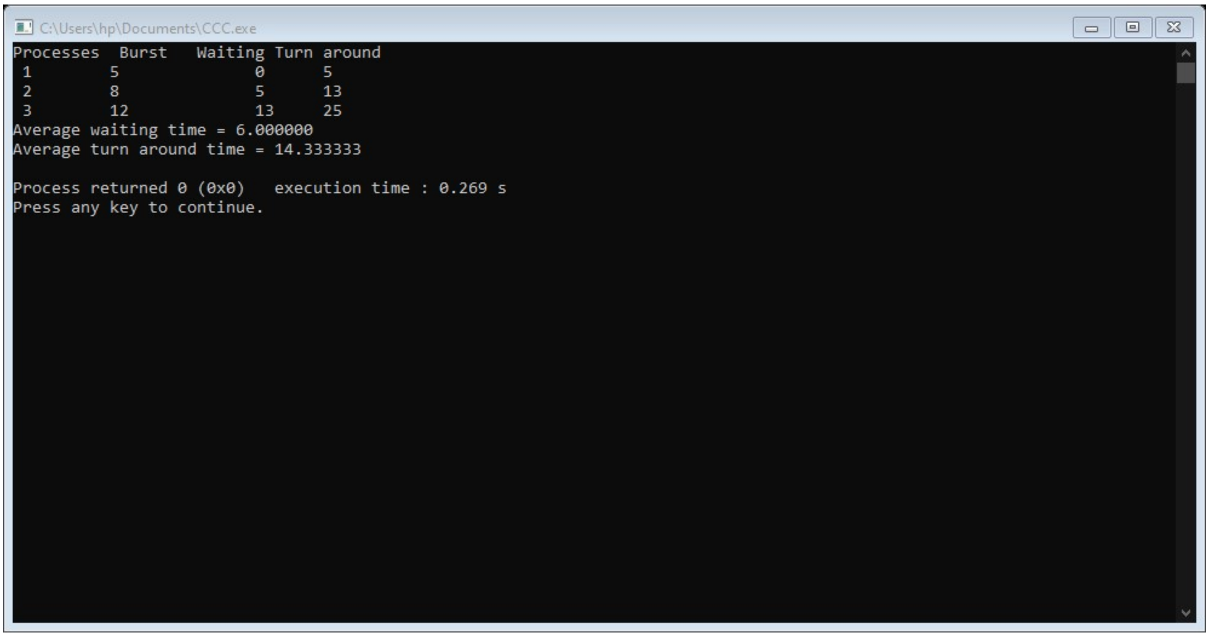
```
#include <stdio.h> int
waitingtime(int proc[], int n, int
burst_time[], int wait_time[])
{    wait_time[0] = 0;
   for (int i = 1; i < n ; i++ )    wait_time[i] =
burst_time[i-1] + wait_time[i-1] ;    return 0;
}
int turnaroundtime( int proc[], int n, int
burst_time[], int wait_time[], int tat[]) {
   int i;
   for ( i = 0; i < n ; i++)    tat[i] =
burst_time[i] + wait_time[i];
return 0;
}
int avgtime( int proc[], int n, int burst_time[])
{    int wait_time[n], tat[n], total_wt = 0, total_tat
= 0;
   int i;
   waitingtime(proc, n, burst_time, wait_time);
turnaroundtime(proc, n, burst_time, wait_time, tat);
printf("Processes  Burst   Waiting Turn around \n");
   for ( i=0; i<n; i++) {      total_wt = total_wt + wait_time[i];      total_tat
= total_tat + tat[i];       printf(" %d\t  %d\t\t %d \t%d\n", i+1,
burst_time[i], wait_time[i], tat[i]);    }

   printf("Average waiting time = %f\n", (float)total_wt / (float)n);
printf("Average turn around time = %f\n", (float)total_tat / (float)n);
return 0;
}
int main() {    int proc[] = { 1, 2, 3};
int n = sizeof proc / sizeof
proc[0];    int burst_time[] = {5, 8,
12};    avgtime(proc, n,
burst_time);
   return 0;
}
```

ALGORITHM

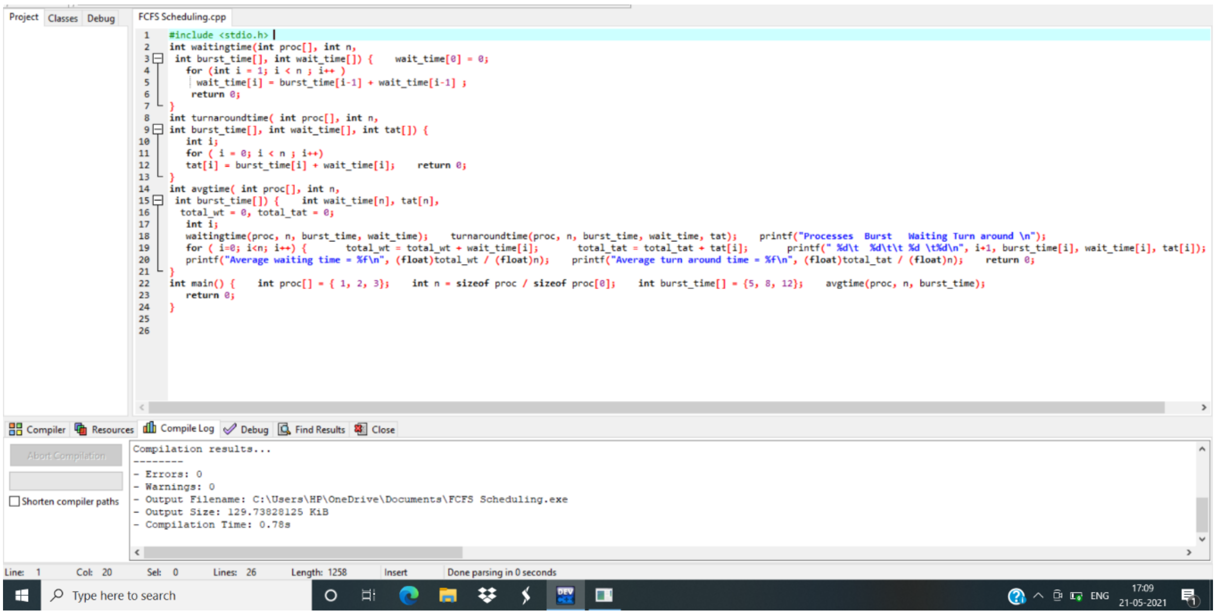
START

```
Step 1- In function int waitingtime(int proc[], int n, int burst_time[], int wait_time[])
  Set wait_time[0] = 0
  Loop For i = 1 and i < n and i++
    Set wait_time[i] = burst_time[i-1] + wait_time[i-1]
  End For
Step 2- In function int turnaroundtime( int proc[], int n, int burst_time[], int wait_time[], int tat[])
  Loop For  i = 0 and i < n and i++
    Set tat[i] = burst_time[i] + wait_time[i]
  End For
Step 3- In function int avgtime( int proc[], int n, int burst_time[])
  Declare and initialize wait_time[n], tat[n], total_wt = 0, total_tat = 0;
  Call waitingtime(proc, n, burst_time, wait_time)
  Call turnaroundtime(proc, n, burst_time, wait_time, tat)
  Loop For  i=0 and i<n and i++
    Set total_wt = total_wt + wait_time[i]
    Set total_tat = total_tat + tat[i]
    Print process number, burstime wait time and turnaround time
  End For
  Print "Average waiting time =i.e. total_wt / n
  Print "Average turn around time = i.e. total_tat / n
Step 4- In int main()
  Declare the input int proc[] = { 1, 2, 3}
  Declare and initialize n = sizeof proc / sizeof proc[0]
  Declare and initialize burst_time[] = {10, 5, 8}
  Call avgtime(proc, n, burst_time)
STOP
```

OUTPUT



```
Processes  Burst   Waiting Turn around
 1        5              0      5
 2        8              5      13
 3        12             13     25
Average waiting time = 6.000000
Average turn around time = 14.333333

Process returned 0 (0x0)   execution time : 0.269 s
Press any key to continue.
```



```c
#include <stdio.h>
int waitingtime(int proc[], int n,
 int burst_time[], int wait_time[]) {    wait_time[0] = 0;
   for (int i = 1; i < n ; i++ )
    wait_time[i] = burst_time[i-1] + wait_time[i-1] ;
    return 0;
}
int turnaroundtime( int proc[], int n,
int burst_time[], int wait_time[], int tat[]) {
   int i;
   for ( i = 0; i < n ; i++)
   tat[i] = burst_time[i] + wait_time[i];    return 0;
}
int avgtime( int proc[], int n,
 int burst_time[]) {    int wait_time[n], tat[n],
  total_wt = 0, total_tat = 0;
   int i;
   waitingtime(proc, n, burst_time, wait_time);    turnaroundtime(proc, n, burst_time, wait_time, tat);    printf("Processes  Burst   Waiting Turn around \n");
   for ( i=0; i<n; i++) {       total_wt = total_wt + wait_time[i];       total_tat = total_tat + tat[i];       printf(" %d\t  %d\t\t %d \t%d\n", i+1, burst_time[i], wait_time[i], tat[i]);
   printf("Average waiting time = %f\n", (float)total_wt / (float)n);    printf("Average turn around time = %f\n", (float)total_tat / (float)n);    return 0;
}
int main() {    int proc[] = { 1, 2, 3};    int n = sizeof proc / sizeof proc[0];    int burst_time[] = {5, 8, 12};    avgtime(proc, n, burst_time);
   return 0;
}
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\HP\OneDrive\Documents\FCFS Scheduling.exe
- Output Size: 129.73828125 KiB
- Compilation Time: 0.78s
```