**Aim:** To calculate CPU idle time percentage while processes do i/o along with execution.

**Algorithm:**

Shortest remaining time ( SRT ) scheduling algorithm selects the process for execution which has the smallest amount of time remaining until completion.

Let three processes be p0, p1 and p2. Their execution time is 10, 20 and 30 respectively.

p0 spends first 2 time units in I/O, 7 units of CPU time and finally 1 unit in I/O.  
p1 spends first 4 units in I/O, 14 units of CPU time and finally 2 units in I/O.  
p2 spends first 6 units in I/O, 21 units of CPU time and finally 3 units in I/O.

AT- Arrival Time, IO-input/output, BT-Burst Time

First process p0 will spend 2 units in IO, next 7 units in BT, then process p1 will spend 14 units in BT (as its 4 units of IO has been spent already when previous process was running) and ten process p2 will spend 21 units in BT (as its 6 units of IO has been spent already when previous processes were running) and atlast 3 units in IO (process p0,p1,p2’s last IO included.)

**Code:**

#include<stdio.h>

int main()

{

int n;

printf("Enter the number of processes: ");

scanf("%d",&n);

float total[n];

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

{

printf("Enter Total Execution Time for process %d: ",i+1);

scanf("%f",&total[i]);

}

float io1[n];

float io2[n];

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

{

io1[i]=(total[i]/10)\*2;

io2[i]=(total[i]/10);

}

/\*for(int i=0;i<n;i++)

{

printf("%d",io1[i]);

}\*/

float burst[n];

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

{

burst[i]=total[i]-io1[i]-io2[i];

}

float idle=io1[0]+io2[n-1];

float sumburst=0.0;

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

{

sumburst+=burst[i];

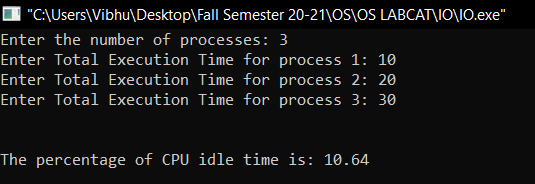
}

float cpuidle=(idle/(idle+sumburst))\*100;

printf("\n\nThe percentage of CPU idle time is: %.2f\n\n",cpuidle);

}

**Output(screenshots):**

****