Week-1 Matrix

```
#include <stdio.h>
#include <conio.h>
void add int a[3][3], intb [3][3]
  {
  int i,j
 for(i=0; i<3; i++){
  for (j=0; j<3; j++){
printf("%d" \t", a[i][j]+ b[i][j]);
 }
 printf("\n");
 }
}

void sub(int a [3] [3], int b [3][3]){
     int i,j;
 for (i=0 ; i<3 ; i++) {
  for (j=0; j<3 ; j++) {
print ("%d "\t", a[i][j]-b[i] [j]);
 }
print ("\n");

for(i=0; i<3; i++) {
  for (j=0, j<3; j++) {
  printf("%d", t, ~~res~~ result [i] [j]
   }
  printf("\n");
  }

 void MUL (int a[3] (3), int b[3][3]) {
  for (int i=0; i<3; i++){
     for (int j=0; j<3; j++)
     for (int k=0; k<3; k++){
```

PAGE NO.
DATE / /

```
        result [i][j] =0;
        result [i][j]= result [i][j]+ a[i][k] +b[k][j]
        }
     }

  void row sum (int a [3][3])
     {
     int i,j,r;
   for (int i=0; i<3; i++){
           r=0
    for ( int j=0; j<3; j++){
         r= r+a[i][j];
     }
  printf ("row %d sum is %d", i, r)
     }
    }

  void column sum(int a [i][j])
    {
    int i, j, r;
    for (int i=0; i<3; i++) {
              r=0;
    for (int j=0; j<3; j++){
            r= r + a[j][i];
       }
    printf ("colown %d sum is %d"; j, r)
    }
  }
```

PAGE NO.
DATE / /

Week-2 FCFS

```
#include <stdio.h>
#include <stdlib.h>

void waitingtime(int proc[], int n, int bursttime[], 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];
 }
}

void turnaroundtime(int proc[], int n, int wait time[], int tat[])
 {
 for(int i=0; i<n; i++)
 tat[i]=burst_time[i]+ wait_time[i];
 }

void avgtime(int proc[], int n, int burst_time[])
 {
 int wait time [n], tat [n], totatal_wt=0, total_tat=0;
 waitingtime(proc, n, burst time, wait_time);
 turnaroundtime(proc, n, burst time, wait_time, tat);
 for(int i= 0; i<n; i++)
 {
 total_wt + wait_time[i];
  total_tat + = tat[i];
 printf("\n process : %d\n burst time %d\n wait time: %d\n
    Turnaround time: %d", proc[i], burst time[i], wait[i], tat[i]);
  }
 printf("\n Avg wait time: %d\n Avg. turn arand time, total tat/n);
  }
```

```
void main()
 {
int proc[10], burst time [10], n;
 printf("\n Enter size of n:");
scanf("%d", &n);
for(int i=0; i<n; i++)
 {
printf("\n enter the processor number:");
scanf("%d", &proc[i]);
printf("\n enter burst time:");
scanf("%d", &burst_time[i]);
 }
 avg_time(proc, n, burst_time);
 }
```

Output -

```
enter the size of n: 5
enter the processor number: 5
enter the burst time: 2
enter the size of n: 6
enter the ~~burst~~ burst time: 5
enter the processor number: 7
enter the burst time: 8
enter the processor ~~time~~ number: 5
enter the burst time: 6
```

```
Avg wait time: 9
Avg. turnaround time: 13
process returned 50(0x32)
execution time: 116.985 s
```

```
process: 5
burst time: 2
wait Time: 0
Turnaround time: 2
process: 6
burst time: 5
```

```
wait Time: 2
Turnaround Time: 15
process: 5
burst time: 6
Wait time: 15
Turnaround time: 21
```

shortest Job first

PAGE NO.
DATE / /

SJF

```
# include <stdio.h>
# include <stdlib.h>
void waiting time (int proc[], int n ; int burst time)
 {
 wait_time [0]=0;
 for (int i= 1 ; i<n ; i++)
 {
  wait_time[i] = burst_time [i-1]+ wait_time [i-1];
  }
 }
void turnaroundtime (int proc [], int n, int wait_time)
 {
  for (int i=0; i<n; i++)
 tat [i] = burst_time [i] +wait time[i];
 }
void avgtime (int proc [], int n , int burst_time[])
 {
 int wait_time [n], tat[n], total wt=0, total tat =0, k;
  for (int i=0 ; i<n; i++)
  {
  ~~for(int f~~  if (burst time [j] < burst_time [i])
       {
       k = burst time[i];
      burst time [i]= burst_time [j];
       burst time [j]=k;
         k = proc [i];
      proc [i] = proc[j];
        proc [j]=k;
     }
    }
```

```
waiting time (proc, n, burst time, wait time);
turn around time (proc, n, burst time, wait time, tat);
 for(int i=0; i<n; i++)
  {
  total_wt + = wait time [i];
  total tat + = tat [i];
 printf("\n process : %d\n proc[i], burst time [i] wait time [i] tat[i]
 }
printf("\n avg wait time: %d\n avg turnaround time : %d", total tat/n);
 }
 void main()
  {
  int proc [10], burst time [10], n;
  printf("\n enter the size of n:");
    scanf("%d", &n);
    for(int i=0; i<n; i+)
    {
   printf ("\n enter processor number:");
     scanf("%d", &proc[i]);
   printf("\n enter burst time:");
     scanf("%d", & burst time [i]);
   }
   avg time (proc, n, burst time);
    }
```

Output -

```
enter the size : n = 3                  enter processor number : 3
enter processor number : 1              enter burst time : 8
enter burst time : 10
enter processor number : 2
 enter burst time : 5
```

Adora PAGE NO. DATE / /

week-3 priority.

```
#include <stdio.h>
#include <stdlib.h>

void waiting time (int proc[], int n, int burst time [], wait_time[])
 {
  wait_time [0]=0;
   for(int i= 1 ; i<n; i++)
     {
  wait_time [i] = burst_time [i-1] + wait_time[i-1];
     }
  }
  void turnaround time(int proc[], int n, int tat[])
   {
   for (int i =0 ; i<n; i++)
      tat [i] = burst_time [i]+ wait_time [i];
   }
   void avgtime [int proc[], int n, int burst_time[]) {
   int wait_time [n], tat[n], total_tat =0 ;
   waiting time (proc, n, burst_time, wait_time);
  turnaround time (proc, n, burst_time, wait_time, tat);
    for (int i=0; i<n; i++)
      {
      total_wt += wait_time [i];
      total_tat += tat [i];
      }
    void sort (int proc[], int burst time [], int n, int priority[]){
        int a, b, c;
     for( int i = 0 ; i<n; i++) {
       for (int j = i+1; j<n; j++) {
      if (priority [i] > priority (j]){
          a = burst_time (i];
burst_time [i]= burst_time [j];
burst_time [j]=a;
  b= proc [i];
 proc [i] = proc [j]
   proc[j]=b;
  c=priority [i]
   priority[i]= priority[j];
   priority [j]=c;
  }
 }
 }
}
void main() {
int proc [10], burst_time [10], n, priority [10];
 printf ("\n enter size of n:");
 scanf ("%d", &n);
  for (int i =0 ; i<n; i++) {
 printf ("Enter processor number:");
 scanf ("%d", & proc [i]);
 printf ("enter burst time:");
  scanf ("%d", &burst_time [i]);
 printf ("enter priority");
 scanf ("%d" & priority [i]); }
 printf ("\n");
 sort (proc, burst_time, n, priority);
  avgtime (proc, n, burst_time);
}
```

Output -

```
enter the size of n :3
enter the processor number : 1
enter the burst time : 12
enter the priority1
enter the processor number : 2
enter the burst time : 3
enter the priority2
enter the processor number : 3
enter the bursttime : 4
enter the priority3

process : 1  Burst Time : 12  WaitTime : 0   Turnaround : 12
process : 2  Burst Time : 3   WaitTime : 12  Turnaround : 15
process : 3  Burst Time : 4   WaitTime : 15  Turnaround : 19
Average wait time : 9        Average turnaround time : 15

process returned 50 (0x32)  execution time : 17.163 s
```

Adora PAGE NO. DATE / /

Round Robin

```
#include<stdio.h>
void main() {
 int n, proc[100], burst_time[100], wait_time [100], tq, i,
 burst_update [100], t=0, turnaround[100], tot_tt=0;
printf ("enter no. of process :");
  scanf("%d", &n);
     c=n;
  printf ("enter time quantum:");
  scanf("%d", & tq);
  printf ("Enter burst times:\n");
    for(i=0; i<n; i++ ){
       proc[i] = i+1;
  printf("enter burst time process %d:");
  scanf("%d", &burst_time [i]);
  burst_ update [i] = burst time[i]; }
      i=0;
    while(c!=0) {
      if (proc [i]=!=0){
      if (burst_update [i]>tq) {
       burst_update [i]= tq;
         t+= tq;
       }
      else {
      t += burst_update [i]=0;
      proc[i]=0;
      turnaround [i]=t;
          c--;
     wait_time [i]= turnaround [i]- burst_time[i];
     }
    }
       i=(i+1)%n;
    }.
for(i=0; i<n; i++) {
        tot_tt += turnaround [i];
        tot_wt += wait_time [i];
   }
printf("\n\nprocess\t\tBurst Time\t\t turnaround Time\n");
   for(i=0; i<n; i++)
printf("%d\t\t%d\t\t\t%d\t\t%d\n", i+1, burst_time[i]
                                wait_time [i]);
 printf("\n\nAverage Turn around time is : %d\n", tot_tt/n);
printf("Avg waiting Time is: %d \n", tot_wt/n);
   }
```

Output -

```
enter no. of process: 3
enter time quantum: 2
enter burst time of process 1 : 4
enter burst time of process 2 : 3
enter burst time of process 3 : 5

Process     Bursttime     wait time     Turnaround Time
  1            4              4              8
  2            3              6              9
  3            5              7              12

Avg  Turn Around time is : 9
Avg. waiting Time is : 5

process returned 27 (0x1B)   execution time : 7.609 s
```

[illegible]
22/6/2023