**Student Name: Ravi Prakash Meena**

**Student ID: 11804942**

**Email Address: raviprakashmeena25@gmail.com**

**GitHub Link:** <https://github.com/RaviPrakashMeena/Operating-Sytem>

**Code:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<string.h>**

**void main()**

**{**

**char name[10][5];**

**char temp[5];**

**int i;**

**int j;**

**int priority1[10];**

**int waiting\_time[10];**

**int totalwt = 0;**

**int priority2[10];**

**int temp\_1;**

**int n;**

**float averagewt;**

**printf("Enter no of students:");**

**scanf("%d",&n);**

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

**{**

**printf("student %d name:",i+1);**

**scanf("%s",&name[i]);**

**printf("enter time for P1:");**

**scanf("%d",&priority1[i]);**

**printf("enter no of gifts:");**

**scanf("%d",&priority2[i]);**

**}**

**for(i=0;i<n-1;i++)**

**{**

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

**{**

**if(priority2[i]>priority2[j])**

**{**

**temp\_1=priority2[i];**

**priority2[i]=priority2[j];**

**priority2[j]=temp\_1;**

**temp\_1=priority1[i];**

**priority1[i]=priority1[j];**

**priority1[j]=temp\_1;**

**strcpy(temp,name[i]);**

**strcpy(name[i],name[j]);**

**strcpy(name[j],temp);**

**}**

**}**

**}**

**waiting\_time[0]=0;**

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

**{**

**waiting\_time[i]=waiting\_time[i-1]+waiting\_time[i-1];**

**totalwt=totalwt+waiting\_time[i];**

**}**

**averagewt=(float)totalwt/n;**

**printf("G\_name\t P\_time\t No.Gifts\tW\_time\n");**

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

**{**

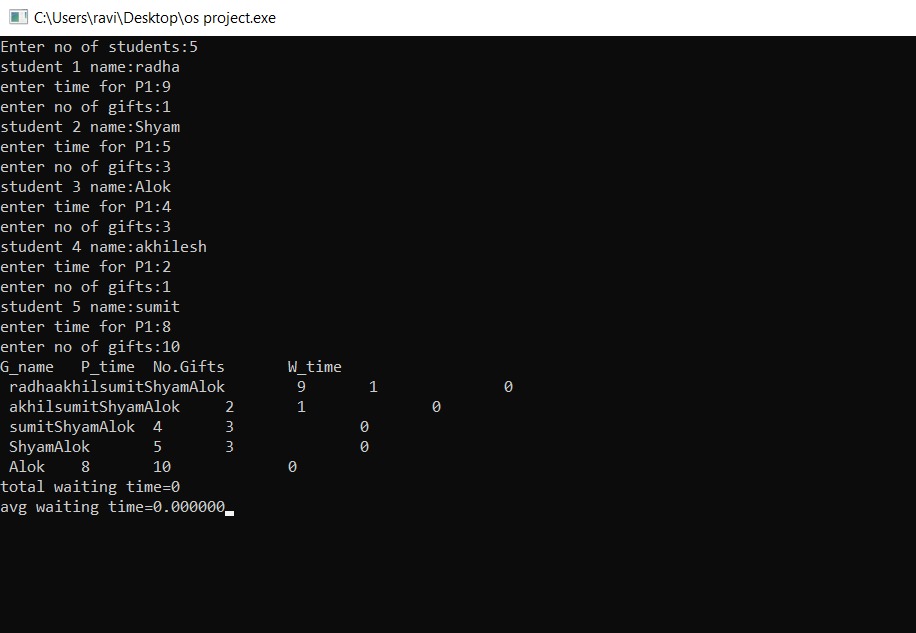
**printf(" %s\t %d\t %d\t \t%d\n" ,name[i],priority1[i],priority2[i],waiting\_time[i]);**

**}**

**printf("total waiting time=%d\navg waiting time=%f",totalwt,averagewt);**

**getch();**

**}**



1.Explain the problem in term of operating system concept(Max 200 words)

This problem is base on priority scheduling. **Priority Scheduling** is a method of scheduling processes that is based on priority. In this algorithm, the scheduler selects the tasks to work as per the priority. The processes with higher priority should be carried out first, whereas jobs with equal priorities are carried out on a round-robin or FCFS basis. Priority depends upon memory requirements, time requirements, etc.

**Preemptive Scheduling**

In Preemptive Scheduling, the tasks are mostly assigned with their priorities. Sometimes it is important to run a task with a higher priority before another lower priority task, even if the lower priority task is still running. The lower priority task holds for some time and resumes when the higher priority task finishes its execution.

**Characteristics of Priority Scheduling**

* A CPU algorithm that schedules processes based on priority.
* It used in Operating systems for performing batch processes.
* If two jobs having the same priority are READY, it works on a FIRST COME, FIRST SERVED basis.
* In priority scheduling, a number is assigned to each process that indicates its priority level.
* Lower the number, higher is the priority.
* In this type of scheduling algorithm, if a newer process arrives, that is having a higher priority than the currently running process, then the currently running process is preempted.

**In the problem the one which is having highest no. of gifts will get first priority and gets its process executed first by giving waiting time = 0sec.**

2.Write the algorithm for proposed solution of the assigned problem.

**Algorithm:**

Step 1: Assign all the process to ready queue.

Step 2: Assign the process to the CPU according to the priority, higher priority process i.e. the process having maximum gifts will get the CPU first than lower priority process. It will happen inside the priority queue.

Step 3: If two processes have similar priority then SJF is used to break the tie

Or we can use exception handling in that case like pass .

Step 4: Repeat the step 1 to 3 until ready queue is empty.

Step 5: Calculate Waiting time and Turnaround time of individual Process.

Step 6: Calculate Average waiting time and Average Turnaround time.

**Description (purpose of use):**

4.Explain all the constraints given in the problem. Attach the code snippet of the implemented constraint.

1st ‘for’ loop will be used for input and output from the user like char type his name,no. of gifts and time. Having complexity as **O(n).**

2nd ‘for’ is a nested for loop in which first loop will help in traverse and incrementation for n no. of processes and also helps in making tabular matrix form.

3rd ‘for’ loop inside the nested body will work for priority comparing using if statement

Inside if statement we will be swapping priorities using third variable known as ‘temp’ variable.

And strcopy fn. To copy name of string type.

These two nested loops are having priority as **O(n^2).**

4th ‘for’ loop is used for calculating total waiting time of aquired processes.having complexity **O(n)**

5th ‘for’ loop will print all the details like name,its priority,no. of gifts and waiting time having complexity **O(n).**

**Total time complexity = O(n^2) for full execution.**

4.Explain all the constraints given in the problem. Attach the code snippet of the implemented constraint.

**Constraints:-**

This proposed problem doesn’t have much constraints except that you can give only 10 inputs because specified for 10 students so it will work for static memory i.e is fixed for 10 students only.

**Code snippet:-**

**char name[10][5];**

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

**{**

**printf("student %d name:",i+1);**

**scanf("%s",&name[i]);**

**printf("enter time for P1:");**

**scanf("%d",&priority1[i]);**

**printf("enter no of gifts:");**

**scanf("%d",&priority2[i]); }**

6. Explain the boundary conditions of the implemented code.

**Description:**

it will ask the user only upto 10 inputs i.e I should have value for n following order (0<n<=10).

Similarly for all the ‘for’ loops we are using.

7.Explain all the test cases applied on the solution of assigned problem**.**

**Description:**

