**Student Name: Syed Sadhik**

**Student ID : 11712966**

**Email Address:syedsadhik2323@gmail.com**

**GitHub Link:** <https://github.com/Nandasagarre/sadhik/tree/master>

1 .PROBLEM STATEMENT:

Ten students (a,b,c,d,e,f,g,h,i,j) are going to get there pictured clicked by university camera. Only one student can enter the camera room while the other students wait outside the room. The students are waiting in a queue to enter the room. To pass time the students start to play a game. In this game the students give candies to each other in a random manner (assume the students never run out of candies). They decide that the student with highest candies will be allowed to enter. When the student with highest amount of candies enter the room, the student starts the game again. Initially the students do not know if there is any body in the room and they start their game and the student with highest candies enter. Write and implement the algorithm to schedule such and compute the waiting and turnaround time. Consider the arrival time and burst time as given by the user.

|  |  |  |
| --- | --- | --- |
|  |  |  |

2. ANALYSIS OF THE PROBLEM STATEMENT (DESCRIPTION):

Given that there are 10 subjects waiting in a queue waiting for their turn to get their job done. Where along with a game is played among them which reinitialize their sequence of their queue according to the one who wins the game.

If the above scenario is viewed in terms of the operating systems, the below conclusions can be drawn.

* There are 10 processes in the ready queue.
* Once the Photoshop room is nothing but the CPU.
* The game they played results in prioritizing the processes.
* With respect to that of the priority the process (student) will be allowed to use the resourceses of CPU(allowed to enter the room).

3.ALGORITHM

Figure 1.0 clearly explains the various data structures that are been used in the implementation of the code.

Figure 1.1 explains the algorithm of the scheduler.

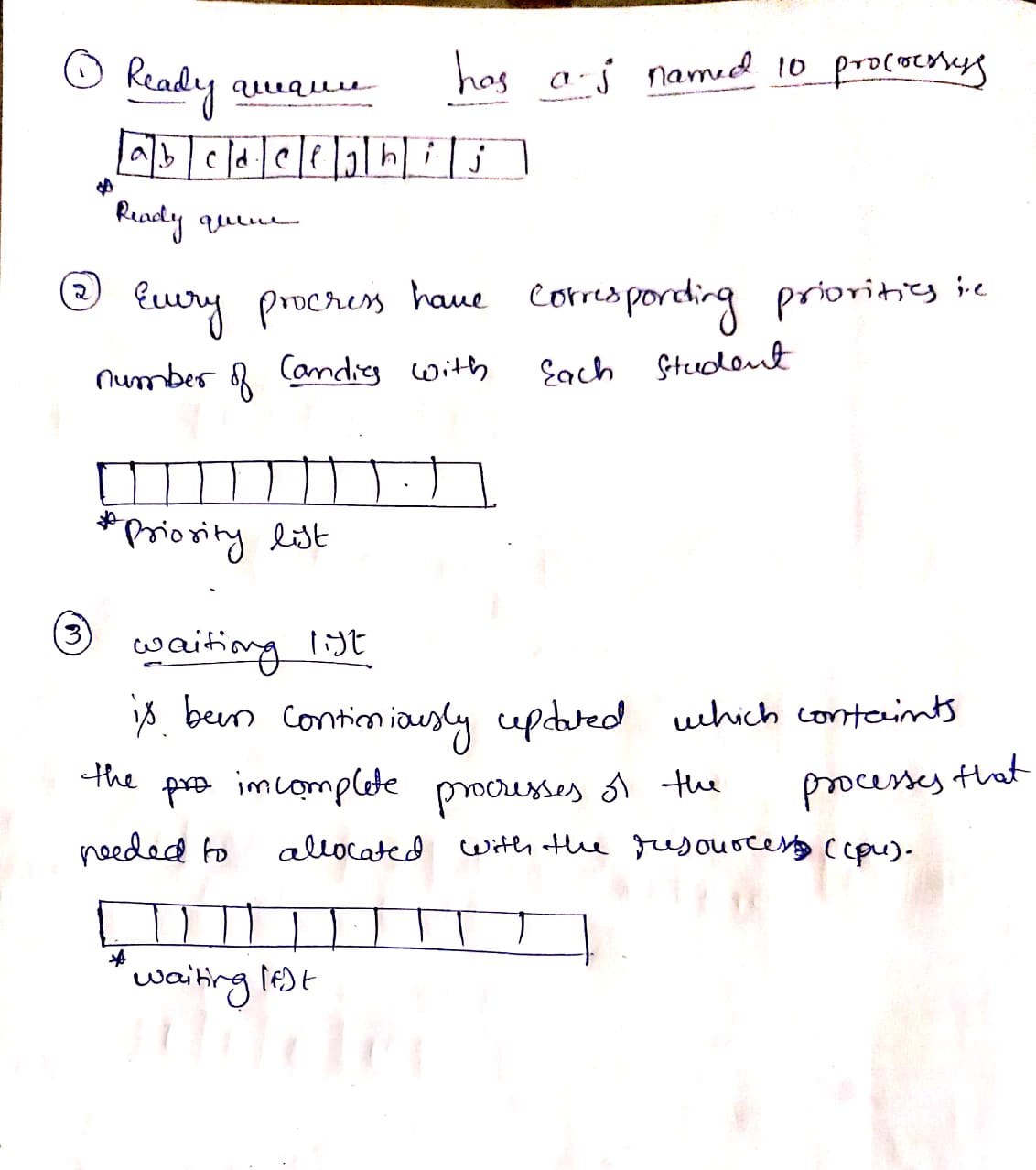


Fig 1.0 shows the different list that are been used

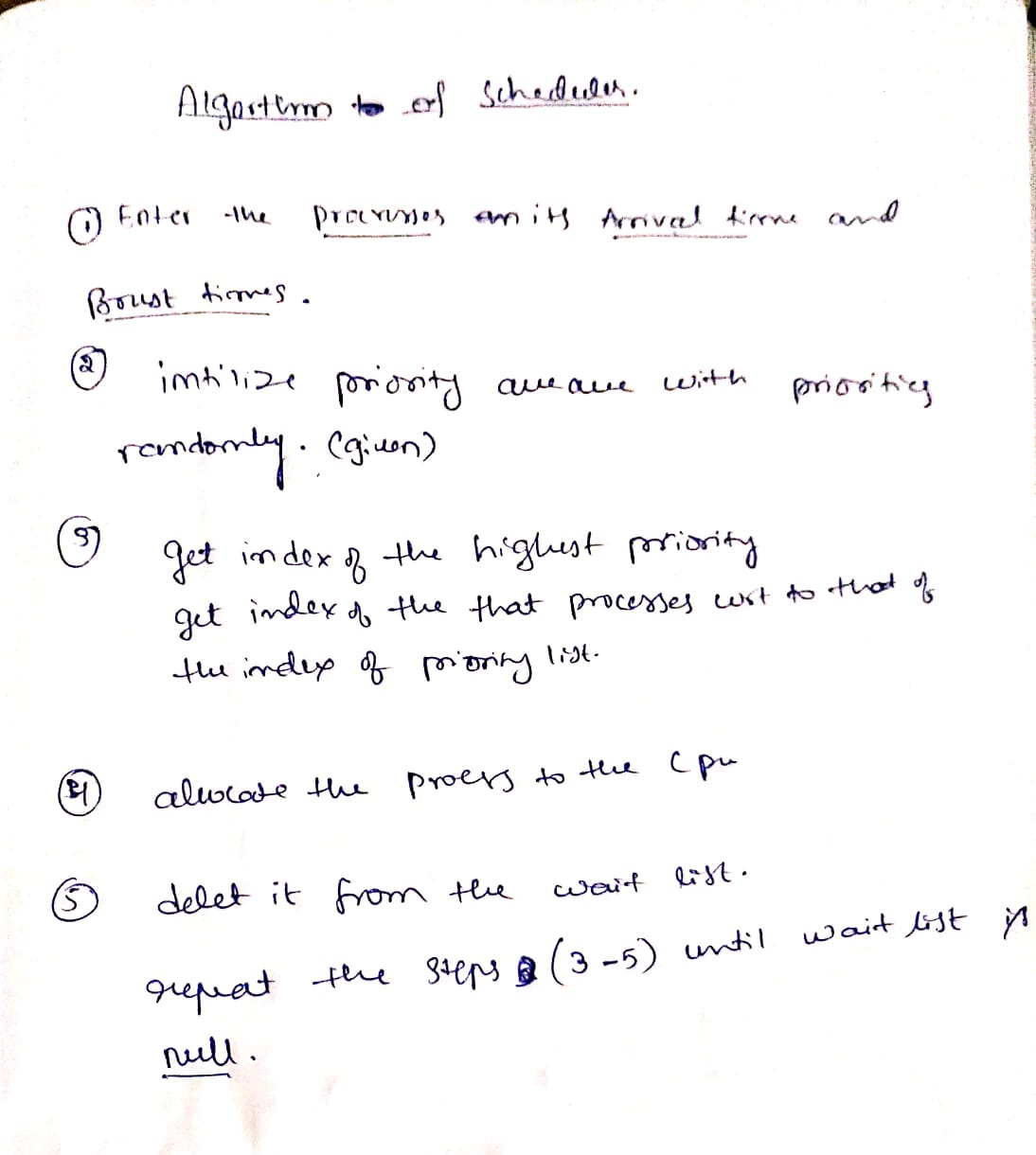


Fig 1.1 explains the algorithm used

4.Code:

5.The constrains are highlighted in the code snippet and the explanation is also provided just below it.

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

using namespace std;

/\* the variables used :

n=number of procresses, time= instanteneous time unit, req= resource requriments(n)

p[]=procresses, at[]=arrival time, bt[]=brust time, wait[]=waiting quaque, pri[]=priority queue,com[]= completed procresses

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*by default the procresses are a,b,c,d,e,f,g,h,i,j\*\*\*\*\*\*\*\*\*which are pre loaded into r[] list \*/

int main()

{ int n,i,k,j=1;

int at[10], bt[10], pri[15],CT[10],TAT[10],WT[10],com[10]; // arrival time;brust time;,priorty queue;

char r[10]; //ready queue

char p[11]={'a','b','c','d','e','f','g','h','i','j','\0'}; //procresses list

char wait[15]; //waitlist

// user enters the procresses, its respective arrival and brust times here

printf("Number of procresses=");

scanf("%d",&n);

int z=n; //internal computation purpose

if(n>10) //manages exceptions

{ printf("out of capacity");

printf("minimum 10 is allowed;\ntry again");

exit(0);

}

//\*\*This piece of code is used to restrict the number of processes greater than 10

for(i=0; i<n;i++) // ------------complexity O(n) for loop---

{

printf("Enter Arrival Time For Process %c:",p[i]);

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

}

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

{ printf("Enter Brust Time For Process %c:",p[i]);

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

wait[i]=p[i];

//loading all avalble procresses into wait list intially;

}

/\*since in the given problem it was metioned that user u'd be allowed to enter the arrival and brust times only

hence \*\*\*it is assumed that the assigning of priorities is totally the programmer call;

and \*\*\*\*\*\*also the problem statement demands for the computation of waiting and turn around time\*/

//assigning the procresses with priority;

for(i=0;i<n;i++) //complexity O(n);

{ pri[i]=i+3;

printf("The priority of procress: %c%d\n",wait[i],pri[i]);

}

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

{

printf("wait list initially%d",wait[i]);

}

for(i=9;i<=0;i--)

{

printf("updated wait list after assignf priorities%d",wait[i]);

}\*/

CT[0]=bt[0];

TAT[0]=0;

WT[0]=0;

//the calculation of waiting time and timearound time

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

{ CT[i+1]=CT[i]+bt[i+1]; //completion time;

TAT[i]=CT[i]-at[i];

j++;

if(TAT[i]<0) //expection handlings

{printf("you have not entered proper values\n");

printf("some of the values are turning into negtive wich is not accpetable;");

printf("try again with proper set of values");

exit(0);

}

//This piece of code is used to monitor the correct set of arrival and brust times

//The input sets resulting in negative values will be warned and execution will be stopped

else

{

WT[i]=TAT[i]-bt[i]; //wait time;

}

}

printf("--------------------------------------------summary------------------------------------------------------------\n\n");

printf(" AT BT CT TAT(ct-at) WT(tat-bt) of\n");

for(i=n-1;i>=0;i--)

{

//printf("The turnaround time of %c is %d units\n",wait[i],TAT[i]);

//printf("The wait time of %c is %d unitd\n",wait[i],WT[i]);

printf("procress: %c %d | %d | %d | %d | %d\n",wait[i],at[i],bt[i],CT[i],TAT[i],WT[i]);}

printf("importat note:\nhere the quantum time is equal brut time.\n the procress is can take its required time to run to its completion\n");

printf("The priority is given to the process in a unusual way by the programer\n");

printf("\*\*\*\*notice the sequence of exuction of the procresses under summmary;\nits according to the priority");}

6.Boundary conditions:

The exaction of the program kills itself because of the following reasons:

* If the number process are greater than 10 or less than 1
* If the set of entered values resulting in negative values of turn-around time

7.Test cases 1:

If Processes >10

-----output of the code---------------------------------

Number of procresses=11

out of capacity

minimum 10 is allowed;

try again

--------------------------------

Process exited after 2.221 seconds with return value 0

Press any key to continue . . .

Test case 2:

If entering bad set of inputs

--------------output of code--------------------------------

Number of procresses=4

Enter Arrival Time For Process a:0

Enter Arrival Time For Process b:5

Enter Arrival Time For Process c:8

Enter Arrival Time For Process d:9

Enter Brust Time For Process a:1

Enter Brust Time For Process b:2

Enter Brust Time For Process c:3

Enter Brust Time For Process d:4

The priority of procress: a3

The priority of procress: b4

The priority of procress: c5

The priority of procress: d6

you have not entered proper values

some of the values are turning into negtive wich is not accpetable;try again wit

h proper set of values

--------------------------------

Process exited after 12.72 seconds with return value 0

Press any key to continue . . .. .

Test case 3:

---output of the code for following test case inputs-----

Number of procresses=4

Enter Arrival Time For Process a:0

Enter Arrival Time For Process b:1

Enter Arrival Time For Process c:2

Enter Arrival Time For Process d:4

Enter Brust Time For Process a:5

Enter Brust Time For Process b:3

Enter Brust Time For Process c:4

Enter Brust Time For Process d:1

The priority of procress: a3

The priority of procress: b4

The priority of procress: c5

The priority of procress: d6

--------------------------------------------summary-----------------------------

-------------------------------

AT BT CT TAT(ct-at) WT(tat-bt) of

procress: d 4 | 1 | 13 | 9 | 8

procress: c 2 | 4 | 12 | 10 | 6

procress: b 1 | 3 | 8 | 7 | 4

procress: a 0 | 5 | 5 | 5 | 0

importat note:

here the quantum time is equal brut time.

the procress is can take its required time to run to its completion

The priority is given to the process in a unusual way by the programer

\*\*\*\*notice the sequence of exuction of the procresses under summmary;

its according to the priority

--------------------------------

Process exited after 16.79 seconds with return value 0

Press any key to continue . . .

**GitHub Link:** <https://github.com/Nandasagarre/sadhik/tree/master>

The above link lands to a page where summary of commits will be displayed.