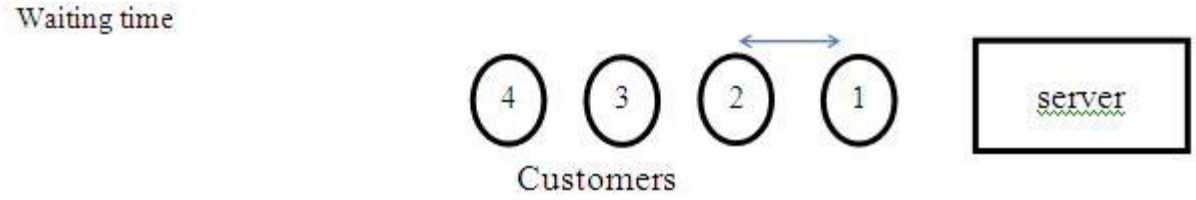
Problem statement

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.

Waiting time is the time interval for which one has to wait after placing a request for an action or service and before the action/service actually occurs.

In operations, it is the time between the actual processes. The main aim of a company is to minimize the waiting time between the processes. At times, waiting time depends on how important a process is. This will tell how fast the request will be serviced or which request will be serviced first in case of multiple requests. The customer behaviour varies depending on the waiting time for a service. e.g. – There maybe balking customers who may not join a queue due to its length. Or customers maybe reneging, leaving the queue after some time without taking the service.



E.g. – Jobs waiting for their turn to be processes in a CNC machine. Here waiting time for a job in queue is the time taken for the job in front to be loaded, machined and removed from a CNC machine

Vehicles waiting at the toll plaza to get a toll receipt. Here waiting time for a vehicle in queue is the time for the vehicle in front to reach the poll booth, obtain the receipt and leave.

Hence, this concludes the definition of Waiting Time along with its overview.

Definition - What does *Turnaround Time (TAT)* mean?

Turnaround time (TAT) is the time interval from the time of submission of a process to the time of the completion of the process. It can also be considered as the sum of the time periods spent waiting to get into memory or ready queue, execution on CPU and executing input/output. Turnaround time is an important metric in evaluating the scheduling algorithms of an operating system.

Turnaround Time (TAT)

In simple terms, turnaround time is the total time needed for an application to provide the required output to the user. From a batch system perspective, turnaround time can be considered the time taken in batch formation and printing of results. The concept of turnaround time overlaps with lead time and contrasts with the concept of cycle time. Turnaround time is expressed in terms of units of time for a specific system state and at times for a given algorithm. Turnaround time varies for different applications and different programming languages.

Many factors influence turnaround time, such as:

- Memory needed for the application
- Execution time needed for the application
- Resources needed for the application
- Operating environment

Turnaround time is an important component in the design of microprocessors, especially for multiprocessor systems. Faster turnaround designs are preferred by hardware design companies, as they lead to faster performance and computing speeds.

Burst time

Every process in a computer system requires some amount of time for its execution. This time is both the CPU time and the I/O time. The CPU time is the time taken by CPU to execute the process. While the I/O time is the time taken by the process to perform some I/O operation. In general, we ignore the I/O time and we consider only the CPU time for a process. So, **Burst time is the total time taken by the process for its execution on the CPU.**

Arrival time

Arrival time is the time when a process enters into the ready state and is ready for its execution.

Code for problem statement

```
#include <iostream>
#include <vector>
#include <random>
#include <windows.h>


using namespace std;

int main()
{
    int num_of_student, temp;
    int waiting_time = 0;
    int turn_around_time = 0;

    cout << "Enter number of student: ";
    cin >> num_of_student;

    vector<int> BurstTime;

    cout << "Enter Burst Time of each student:\n";
    for (int i = 0; i < num_of_student; i++)
    {
        cout << "[" << i + 1 << "] Burst Time : ";
        cin >> temp;
        BurstTime.push_back(temp);
    }

    for (int i = 0; i < num_of_student; i++)
    {
        cout << "\ncurrent number of Candies each students having are:";

        int max_candy_position;
        int max_candy  = -1;

        for (int j = 0; j < num_of_student - i; j++)
        {
            cout << "\n[" << j + 1 << "]: ";
            temp = rand() % 10;
            cout << temp;

            if (max_candy < temp)
            {
                max_candy = temp;
                max_candy_position = j;
            }
            Sleep(1000);
        }
```

```
        cout << "\nmax number of candy: " << max_candy
             << ", position: " << max_candy_position + 1 << endl;

        cout << "So, Student whose Burst Time is " << BurstTime[max_candy_position]
            << " is going to be PHOTOGRAPHED NOW..." << endl;


        if (i == 0)
        {
            turn_around_time += BurstTime[max_candy_position];
        }
        else
        {
            waiting_time += BurstTime[max_candy_position];
            turn_around_time += waiting_time + BurstTime[max_candy_position];
        }


        BurstTime.erase(BurstTime.begin() + max_candy_position);

        Sleep(7000);
    }

    cout << "\n\ntotal waiting time: " << waiting_time << endl;
    cout << "total turn around time: " << turn_around_time << endl;

    return 0;
}
```