HQ11

// a set of homeworks represented in H[] array is given as input, along with the following information:

// $t_1, t_2, \dots, t_n$: time taken to solve homeworks (in hours)

// $m_1, m_2, \dots, m_n$: full marks of homeworks

// $d_1, d_2, \dots, d_n$: deadlines of homeworks

// T : particular time instance

1. given K, T, compute

$MM(K, T)$ = maximum marks obtained from homeworks $H[hw_K, \dots, hw_n]$ when we start to solve them at time T.

2. $MM(n, T) = H[hw_n].m$ // base case

$$MM(j, T) = \max_{\forall j < n} \begin{cases} H[hw_j].m + MM(j+1, T + H[hw_j].t & \text{if } H[hw_j].t + T \leq H[hw_j].d \\ MM(j+1, T) \end{cases}$$

3. Let M be the maximum marks obtained from homeworks $H[hw_K \dots hw_n]$ when started at time T.

Now,

$$M = \max \begin{cases} H[hw_K].m + MM(j, T + H[hw_K].t) \\ MM(j, T) \end{cases}$$

where $MM(j, T + H[hw_K].t)$ and $MM(j, T)$ returns the maximum score of homeworks $H[hw_j \dots hw_n]$ when we start at time $T + H[hw_K].t$ and T respectively where $j > k$.

also we can write the above equation as

$$M = \max \begin{cases} H[hw_K].m + m_1' \\ m_2' \end{cases}$$

where m_1' and m_2' are the ~~value~~ marks obtained for the above two cases respectively.

<u>claim</u>: m_1' and m_2' must be the maximum marks obtained from homeworks $H[hw_j \dots hw_n]$ when we start at time $T + H[hw_k].t$ and T

If m_1' and m_2' are not the maximum marks obtained and instead we had a greater marks m_1'' and m_2'' such that; $m_2'' > m_2'$ Then we would get a marks mxm such that

$$mxm = \max \begin{cases} H[hw_k].m + m_1'' \\ m_2'' \end{cases}$$

which would obviously be greater than m. But this contradicts our assumption that m is the maximum marks from $H[hw_k \dots hw_n]$ started at time T. Thus,

$$M = \max_{\forall j > k} \begin{cases} H[hw_k].m + MM(j, T + H[hw_k].t) & \text{if } H[hw_k].t + T \leq H[hw_k].d \\ MM(j, T) \end{cases}$$

where max is taken on all j such that $j > k$. If there is no such j then $M = H[hw_k].m$

Also for a single homework given $H[hw_1]$ we can simply take the marks of that homework.

4. 2D array $W[1 \dots n][0 \dots T]$ (where $T = \Sigma H[hw_1 \dots hw_n].t$)
$W[j][T]$ will store the value of $MM(j, T)$.

5. Initialize $W[n][T] = MM(n, T) = H[hw_n].m$

```
for i = n-1 .... 1
    for j = T-1 ... 0
        compute W[i][j] = MM(i,j)
```

Using the reccurance formula on the given array $H[]$.

6. Maximum marks obtained from homeworks $H[]$ when starting to solve them at time $T = 0$

(7) Space complexity $= O(n \times T)$

where $T = \sum_{i=1}^{n} (H[hw_i].t)$

Time Complexity $= O(n \times T)$

since there are $O(n \times T)$ computations to be done where each computation requires $O(1)$ time.

(8) To get the actual order of submitting homeworks we will also store pointers in 2D array W that points to other indices of W 2D array. We denote the pointer associated with $W[j][T]$ as $W[j][T].p$.

Let $MaxMarks_j$ be the max marks obtained from $H[hw_j \dots hw_n]$ when started at time T where $MM(j, T)$ will point to $\max(MM(k,T), MM(k, T + H[hw_j].t))$ where $k > j$ i.e., it will point to the next homework hw_k which should be solved in sequence.

The pointers can be computed while calculating values of $W[][]$

$W[j][T].p = W[k][T]$ if $MM(k,T)$ is maximum

and

$W[j][T].p = W[k][T + H[hw_j].t]$ when $MM(k, T + H[hw_j].t)$ is maximum

for base case i.e., $W[n][T].p = $ null

finally to get the order of submission of HW —

```
i = 1 , j = 0
initialise temp[1...n] = 0
while (WW[i][j].p != NULL)
    print (WW[i][j].p)
    W[i][j] = W[i][j].p
    temp[i] = 1
for (i=1 ... n)
    if (temp[i] ! = 1)
        Print (temp[i])
```

Thus we will print the requred sequence of HW for $H[HW_1, \ldots HW_n]$ when submitted on time. The for loop will print the remaining HW which are submitted late.