The map represents a two-dimensional grid of boxes

C – trail
H – bush, path slower than forest trail
N – impassable object (rock)
D – dragon
P – princess

Your task is to implement the following function in the **C programming language:**

```
int *save_the_princess (char **map, int n, int m, int t, int *journey_length);
```

This feature is to find a way to proceed so that the knight saves all the kidnapped princesses as soon as possible and neutralize the dragon. The map is **n** squares high and **m** squares wide (1 ≤ n, m ≤ 100). Parameter **t** (0 ≤ t ≤ 106) determines when from the beginning of your search the dragon wakes up and eats the first princess. Since the dragon flies very fast, rest assured that if you fail to destroy him until he wakes up, you will no longer save the princess.

It takes one unit of time to cross the forest path and the dragon wakes up in the t-th unit of time, when it will be late. **And don't forget that you must first destroy the dragon before you can save the princesses** (although you would also go through the box where the princess is before). After all, what would it be like to fight a dragon with the crowd of princesses as spectators ...

Return the found path as a sequence of coordinates (pairs of integers x, y, where 0 ≤ x <m and 0 ≤ y <n) of all visited fields. You are always at the beginning with a field with coordinates 0.0 and at the last field visited must be one of the kidnapped princesses. If there are several roads of the same duration, you can return any of them. Set the output argument **journey_length** to the number of coordinates you returned as a return value.

Implement the above functionality as efficiently as possible. Use the **Dijkstra algorithm with a binary heap (Min Heap) to find the shortest path**.

Example of using the function:

```
Int i, *path = save_the_princess (map, n, m, t, &journey_length);
for (i=0;i<journey_length;++i)
        printf("%d %d\n",path[i*2],path[i*2+1]);
```

IMPORTANT NOTICE:
The knight moves only in four directions (up, down, left, right). It takes one unit of time to go through the field by default. Thus, it also takes one unit of time to cross the squares with the dragon and the princess. All fields through which the knight passes are included in the final time. Thus, if it starts on the field (0,0), passes through the field (1,0) and ends on the field (1,1), while all three fields contain a forest path, then it takes 3 units of time. Destruction of the dragon is an immediate action, the duration of which is negligible due to the duration of the passage through the square (i.e. we include it with zero duration). Since knight has a number of other princesses that still need to be rescued in other parts of the world (there are several kingdoms in the entrance where it wants to save the princesses), you must

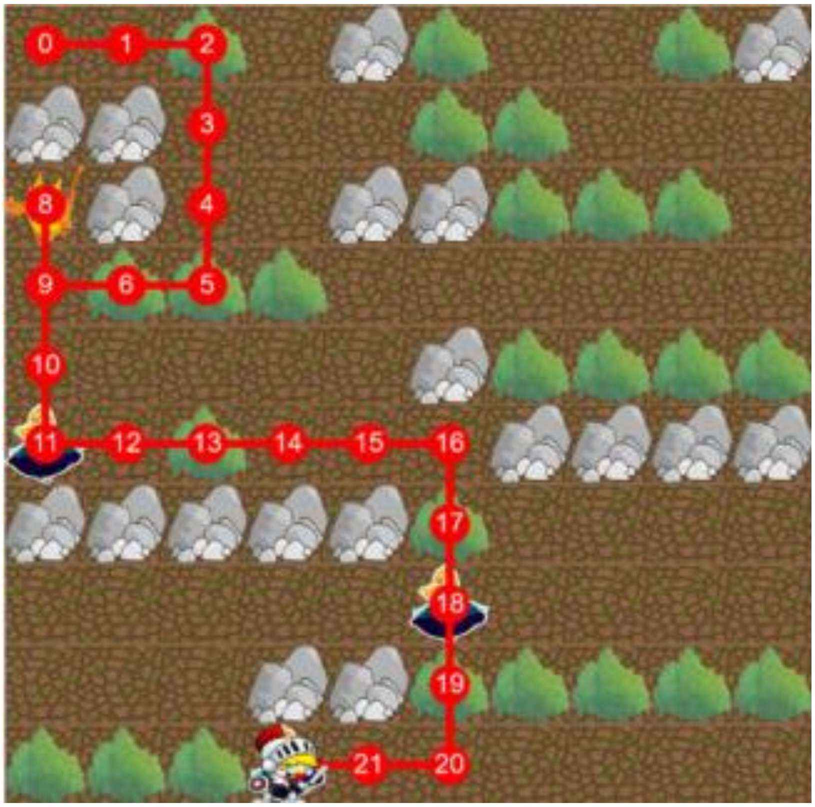
rescue the princesses as soon as possible (including destroying the dragon) so you can move on to the next kingdom. **A maximum of 3 princesses are marked on the map**

Input from file:

10 ->n, map height
10 ->m, map width
12 ->t, time, how long until the dragon wakes up

```
10 10 12
CCHCNHCCHN
NNCCCHHCCC
DNCCNNHHHC
CHHHCCCCCC
CCCCCNHHHH
PCHCCCNNNN
NNNNNHCCCC
CCCCCPCCCC
CCCNNHHHHH
HHHPCCCCCC
```

The test output can also be visualized in the tool
https://popolvar.surge.sh/



OUTPUT

```
0 0
1 0
2 0
2 1
2 2
2 3
1 3
0 3
0 2
0 3
0 4
0 5
1 5
2 5
3 5
4 5
5 5
5 6
5 7
5 8
5 9
4 9
3 9
```