bool is_valid(int x, int y, int n, int m) {
    if(x<=0 || y<=0 || x>n || y>m) return 0;
    return 1;
}

int main() {
    int t;
    scanf("%d", &t);
    while(t--){
        for(int i=0; i<250005; i++) adj[i].clear();
        int n, m;
        scanf("%d %d", &n, &m);
        for(int i=1; i<=n; i++){
            for(int j=1; j<=m; j++){
                scanf("%d", &mat[i][j]);
            }
        }
        // for(int i=1; i<=n; i++) {
        //     for(int j=1; j<=m; j++)
        //     }
        // for(int i=1; i<=n; i++) {
        //     for(int j=1; j<=m; j++)
        //     }
    }
}
Choosing the best developer assessment solution, boils down to one crucial factor: The quality of the screening process itself. Proctoring or anti-cheating mechanisms cement the foundation of an excellent screening process. A proctored assessment ensures that candidates taking a test do not cheat.

HackerEarth’s developer assessment platform is backed by powerful proctoring settings that make your tests completely cheat-proof. Here's how you can increase the quality of your screening process using these tips:
No more copy and paste

Encourage candidates to come up with answers based on their own logical and programming skills. HackerEarth assessments have a feature to disable candidates from copying code from the Web or another local computer and pasting it on the code editor. In addition, the platform also disallows candidates from copying questions from the question library and sharing it on the web.

Additionally, the platform has the ability to automatically compare code written by two candidates and check if the code has been plagiarised.
Take candidate snapshots

Prevent impostors from taking advantage of an online assessment by ensuring you have a visual proctor in place. Using a webcam, the HackerEarth platform takes regular snapshots of the candidate during a test. A thumbnail preview of the webcam is displayed on the bottom right of the screen to indicate that candidates are being monitored. Candidate snapshots are captured randomly at an interval of 20-30 secs (3 snapshots/ min).

During the test, this mechanism can detect if

- Additional people are present
- The candidate is absent
- If the candidate has attempted the test throughout the total duration of the test
After the test, you can view a candidate's report to determine how many snapshots have violated these guidelines.

Additionally, you can view the snapshots to determine any discrepancies in the ones that have been flagged.
The snapshots are divided into the following categories:

**Multiple candidates:**

If there is more than one candidate present in a snapshot, then it is stored in the **Multiple candidates category**. Multiple candidates are differentiated by green-colored bounding boxes.
**Candidate missing:**

If the candidate is missing in a snapshot, then it is stored in the **Candidate missing category**.

**Needs review:**

If the snapshot of the candidate attempting the test is different from the snapshot taken at the beginning of the test, then these are stored in the **Needs review category**. This is done by comparing the reference and the selected image. If a snapshot does not match with the reference image, then it is stored in this category.
All snapshots:

This category contains all the snapshots that are taken during a test.

- Reference image: Snapshot of candidate taken during the starting of the test.
- Selected image: Snapshot of another person taken while attempting the test.

Check for questions available on the internet

The HackerEarth platform has a feature to detect if a question is available to the public on the internet.

What’s more?
On clicking the said question, it lists the source URL where the particular question was found.
Disable tab switching

Establish a controlled test environment for candidates to remain focused on the test and answer honestly. Prevent a candidate from switching to another window or tab to browse answers.

HackerEarth’s platform issues a warning each time a candidate tries to leave the test environment. This is measured by the number of tab switches made by the candidate. If a candidate makes more than 5 tab switches, they are automatically logged out of the test environment.

Logout on leaving test environment

**Important:**
Ensure that your candidates are aware that:

1. Candidates will be given a warning if they click anywhere outside the test environment, after which their participation will be terminated automatically.
2. This will also happen if they click on system-generated popups.

Would you like to enable this setting?

---

NO  YES
Enable full-screen mode

Provide an environment devoid of distractions. Using the HackerEarth platform, you can enable the full screen mode for candidates taking a test. The platform issues a warning to candidates if they exit the full screen mode.

Restrict IP addresses

Sometimes it’s better to know if the candidates are taking the test from the right location, especially if there’s a recruitment drive or if you’ve invited a candidate to take the test from a specified place.
HackerEarth also has a **feature to restrict IP addresses** to contain the test-takers in a single location. With this feature, a warning is displayed if the candidate tries to log in from an IP that is outside the restricted range.

If you need help using proctoring settings in your assessments, write back to us at support@hackerearth.com. If you’re new to HackerEarth and want to create accurate skill-based developer assessments, **sign up for our 14-day free trial**.