

The king's Feast

The screenshot shows the OnlineGDB IDE interface. On the left, there's a sidebar with links for 'IDE', 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. The main workspace has tabs for 'main.cpp' and 'input'. The code editor contains the following C++ code:

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     vector<long long> arr(n);
10    for (int i = 0; i < n; i++) {
11        cin >> arr[i];
12    }
13    long long maxFood = *max_element(arr.begin(), arr.end());
14
15    cout << maxFood << endl;
16    return 0;
17 }
```

The 'input' tab shows the input data:

```
5
2 7 1 9 5
9
```

The output window at the bottom shows:

```
...Program finished with exit code 0
```

At the bottom, there's a taskbar with icons for various applications like File Explorer, Task View, and Google Chrome. The system tray shows the date and time as 06-10-2025 11:09.

The Lost Soldier

S Online C++ Compiler - online.com X +

onlinegdb.com/online_c++_compiler

OnlineGDB

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

main.cpp

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8     vector<int> arr(n);
9     for (int i = 0; i < n; i++) cin >> arr[i];
10
11    long long expectedSum = (long long)n * (n + 1) / 2;
12    long long actualSum = 0;
13    for (int num : arr) actualSum += num;
14
15    cout << expectedSum - actualSum << endl;
16    return 0;
17 }
```

close ad [x]

SONY BRAVIA CINEMA IS COMING HOME

Optum

input

```
5
0 1 2 4 5
3
```

...Program finished with exit code 0

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

Very humid Now

Search ENG IN 11:12 06-10-2025

Potion Mixing (Two Sum)

The screenshot shows the OnlineGDB IDE interface. The code editor displays a C++ file named main.cpp containing the following code:

```
1 #include <iostream>
2 #include <vector>
3 #include <unordered_map>
4 using namespace std;
5 int main() {
6     int n, target;
7     cin >> n;
8     vector<int> arr(n);
9     for(int i=0;i<n;i++) cin >> arr[i];
10    cin >> target;
11    unordered_map<int,int> mp;
12    pair<int,int> ans = {-1,-1};
13    for(int i=0;i<n;i++){
14        int complement = target - arr[i];
15        if(mp.count(complement)){
16            ans = {mp[complement], i};
17            break;
18        }
19        mp[arr[i]] = i;
20    }
21    if(ans.first != -1)
22        cout << "(" << ans.first << "," << ans.second << ")" << endl;
23    else
24        cout << "No solution" << endl;
25 }
```

The terminal window at the bottom shows the input and output of the program. The input is:

```
4
3 2 4 7
6
```

The output is:

```
(1,2)
```

At the very bottom, the message "...Program finished with exit code 0" is displayed.

The Secret Message

The screenshot shows the OnlineGDB IDE interface. The code editor contains the following C++ code:

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int main() {
6     int n;
7     cin >> n;
8     vector<int> arr(n);
9     for (int i = 0; i < n; i++) cin >> arr[i];
10    reverse(arr.begin(), arr.end());
11    cout << "[";
12    for (int i = 0; i < n; i++) {
13        cout << arr[i];
14        if (i != n - 1) cout << ",";
15    }
16    cout << "]" << endl;
17    return 0;
18 }
19
```

The input provided is:

```
4
1 2 3 4
[4,3,2,1]
```

The output displayed is:

```
...Program finished with exit code 0
Press ENTER to exit console.
```

The browser address bar shows: `onlinegdb.com/online_c++_compiler#`

The system tray at the bottom right shows: 6 27°C Cloudy, ENG IN, 11:38, 06-10-2025.

The King's Parade

The screenshot shows the OnlineGDB IDE interface. The code editor window displays a C++ file named `main.cpp` with the following content:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int main() {
5     int n;
6     cin >> n;
7     vector<int> arr(n);
8     for (int i = 0; i < n; i++) cin >> arr[i];
9     bool sorted = true;
10    for (int i = 1; i < n; i++) {
11        if (arr[i] < arr[i - 1]) {
12            sorted = false;
13            break;
14        }
15    }
16    if (sorted)
17        cout << "true" << endl;
18    else
19        cout << "false" << endl;
20    return 0;
21 }
```

The terminal window below shows the input `4` followed by the output `true`.

The left sidebar of the IDE includes links for **My Projects**, **Classroom**, **Learn Programming**, **Programming Questions**, **Sign Up**, and **Login**.

The bottom status bar shows system icons for battery, signal, and date/time.

The Treasure Island

The screenshot shows the OnlineGDB IDE interface. The main window displays a C++ code editor with the file name `main.cpp`. The code reads a 2D vector from standard input and prints the maximum sum of a row to standard output.

```
#include <iostream>
#include <vector>
#include <climits>
using namespace std;

int main() {
    int rows, cols;
    cin >> rows >> cols;
    vector<vector<int>> grid(rows, vector<int>(cols));
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            cin >> grid[i][j];
        }
    }
    int maxRow = -1;
    long long maxSum = LLONG_MIN;
    for (int i = 0; i < rows; i++) {
        long long sum = 0;
        for (int j = 0; j < cols; j++) {
            sum += grid[i][j];
        }
        if (sum > maxSum) {
            maxSum = sum;
            maxRow = i;
        }
    }
    cout << "Row " << maxRow + 1 << " (sum=" << maxSum << ")" << endl;
    return 0;
}
```

The status bar at the bottom shows the input data:

```
3 3
1 2 3
4 5 6
7 8 9
Row 3 (sum=24)
```

Output text:
...Program finished with exit code 0
Press ENTER to exit console.

The Spiral Library

The screenshot shows the OnlineGDB IDE interface. The code editor contains a C++ program named `main.cpp` which prints a spiral matrix. The code uses nested loops and vectors to achieve this. The IDE also displays the input and output sections, showing sample data and the resulting spiral matrix.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int m, n;
7     cin >> m >> n;
8     vector<vector<int>> a(m, vector<int>(n));
9     for (int i = 0; i < m; i++) {
10         for (int j = 0; j < n; j++) {
11             cin >> a[i][j];
12         }
13     }
14     vector<int> res;
15     int t = 0, b = m-1, l = 0, r = n-1;
16     while(t <= b && l <= r) {
17         for(int j=l;j<=r;j++) res.push_back(a[t][j]);
18         for(int i=t;i<=b;i++) res.push_back(a[i][r]);
19         if(t<=b) for(int j=r;j>=l;j--) res.push_back(a[b][j]);
20         if(l<=r) for(int i=b;i>=t;i--) res.push_back(a[i][l]);
21     }
22     cout << "[";
23     for(int i=0;i<res.size();i++){
24         if(i) cout << ",";
25         cout << res[i];
26     }
27     cout << "]\n";
28     return 0;
29 }
```

Input:

```
3 3
1 2 3
4 5 6
7 8 9
[1,2,3,6,9,8,7,4,5]
```

Output:

```
...Program finished with exit code 0
Press ENTER to exit console.
```

The Royal Diagonal

The screenshot shows a web-based IDE interface for OnlineGDB. The code in the editor is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n, m;
7     cin >> n >> m;
8     vector<vector<int>> mat(n, vector<int>(m));
9     for (int i = 0; i < n; i++) {
10         for (int j = 0; j < m; j++) {
11             cin >> mat[i][j];
12         }
13     }
14     int primary = 0, secondary = 0;
15     for (int i = 0; i < n; i++) {
16         primary += mat[i][i];
17         secondary += mat[i][m - i - 1];
18     }
19     cout << "Primary diagonal sum = " << primary << ", Secondary diagonal sum = " << secondary << endl;
20     return 0;
}
```

The input provided was:

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

The output displayed was:

```
Primary diagonal sum = 15, Secondary diagonal sum = 15
...Program finished with exit code 0
```

The status bar at the bottom right shows the date and time as 06-10-2025 12:17.

The Messenger's Path

Online C++ Compiler - onlinegdb.com

onlinedb.com/online_c++_compiler

OnlineGDB

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

- My Projects
- Classroom new
- Learn Programming
- Programming Questions
- Sign Up
- Login

main.cpp

```
1 #include <iostream>
2 #include <vector>
3 #include <queue>
4 using namespace std;
5
6 int main() {
7     int n, m; cin >> n >> m;
8     vector<vector<int>> g(n, vector<int>(m));
9     for (auto& row : g) for (int& cell : row) cin >> cell;
10    if (g[0][0] || g[n-1][m-1]) return cout << "false", 0;
11    queue<pair<int, int>> q; q.push({0, 0});
12    vector<vector<bool>> v(n, vector<bool>(m));
13    int dx[] = {0, 0, 1, -1}, dy[] = {1, -1, 0, 0};
14    while (!q.empty()) {
15        auto [x, y] = q.front(); q.pop();
16        if (x == n-1 && y == m-1) return cout << "true", 0;
17        for (int i = 0; i < 4; ++i) {
18            int nx = x + dx[i], ny = y + dy[i];
19            if (nx >= 0 && ny >= 0 && nx < n && ny < m && !g[nx][ny] && !v[nx][ny])
20                v[nx][ny] = true, q.push({nx, ny});
21    } cout << "false";}
```

close ad [x]

Google Ads



With Google Ads, we got 10x more leads than usual

Raghunandan Saraf, CEO
Saraf Furniture

Learn more

input

```
3 3
0 0 0
0 1 0
0 0 0
true
```

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online

Sports headline
F1 players standi...

Search

18:04
ENG US 07-10-2025

The Rainwater Pond

The screenshot shows the OnlineGDB IDE interface. The code editor contains a C++ program named main.cpp. The code implements a depth-first search (DFS) algorithm to count the number of rainwater ponds in a grid. The input consists of two integers (n, m) followed by an n x m grid of integers. The output is the total number of ponds.

```
#include <iostream>
using namespace std;

int n, m, grid[100][100];

void dfs(int x, int y) {
    grid[x][y] = 0;
    for (int d = 0; d < 4; d++) {
        int nx = x + "2011"[d] - '1', ny = y + "1120"[d] - '1';
        if (nx >= 0 && ny >= 0 && nx < n && ny < m && grid[nx][ny])
            dfs(nx, ny);
    }
}

int main() {
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> grid[i][j];

    int ponds = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            if (grid[i][j]) ponds++, dfs(i, j);

    cout << ponds << endl;
}
```

The terminal window at the bottom shows the input and output. The input is:

```
3 3
1 0 1
0 1 0
1 0 1
5
```

The output is:

```
input
```

Tower of Temples(Hanoi)

The screenshot shows the OnlineGDB IDE interface. The main window displays a C++ program named 'main.cpp' with the following code:

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     int n;
7     cin >> n;
8
9     long long moves = (1LL << n) - 1; // 2^n - 1
10    cout << moves << endl;
11
12    return 0;
13 }
```

The 'input' field contains the numbers 3 and 7, which are the expected inputs for the Hanoi Tower problem. The output window shows the program's response:

```
...Program finished with exit code 0
Press ENTER to exit console.
```

The browser address bar shows the URL: onlinegdb.com/online_c++_compiler#. The bottom status bar includes weather information (27°C, Light rain), system icons (Search, Task View, File Explorer, Control Panel, Mail, Google Chrome), and system status (ENG IN, 12:29, 06-10-2025).

The Magical Staircase

A screenshot of the OnlineGDB IDE interface. The main window shows a code editor with a C++ file named 'main.cpp'. The code implements a recursive solution to calculate the number of ways to climb a staircase of n steps. The user has inputted '4' at the bottom of the code editor. The output window below shows the result: '...Program finished with exit code 0' and 'Press ENTER to exit console.' The browser address bar shows the URL 'onlinegdb.com/online_c%2B%2B_compiler#'. The browser status bar at the bottom right indicates the date and time as '06-10-2025 12:31'.

```
main.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cin >> n;
7
8     if(n==0) { cout<<0<<endl; return 0; }
9     if(n==1) { cout<<1<<endl; return 0; }
10
11    long long a = 1, b = 1; // ways to reach step 1 and 2
12    for(int i=2;i<=n;i++){
13        long long c = a + b;
14        a = b;
15        b = c;
16    }
17
18    cout << b << endl;
19    return 0;
20 }
21
```

input
4
5

...Program finished with exit code 0
Press ENTER to exit console.

OnlineGDB online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE My Projects Classroom new Learn Programming Programming Questions Sign Up Login
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online 7 27°C Light rain ENG IN 06-10-2025 12:31

The Sorcerer's Spell

A screenshot of the OnlineGDB IDE interface. The main window shows a code editor with a file named 'main.cpp' containing C++ code to reverse a string. The code uses recursion to reverse the string 'abc' and prints 'cba'. Below the code editor is a terminal window showing the execution results. The interface includes a navigation bar, a sidebar with user navigation links, and a system tray at the bottom.

OnlineGDB

online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

My Projects

Classroom new

Learn Programming

Programming Questions

Sign Up

Login

main.cpp

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 string reverseString(const string &s, int index) {
6     if(index < 0) return "";
7     return s[index] + reverseString(s, index - 1);
8 }
9
10 int main() {
11     string str;
12     cin >> str;
13
14     string rev = reverseString(str, str.size() - 1);
15     cout << rev << endl;
16
17     return 0;
18 }
19
```

input

```
abc
cba
```

...Program finished with exit code 0

Press ENTER to exit console.

About • FAQ • Blog • Terms of Use • Contact

Us • GDB Tutorial • Credits • Privacy

© 2016 - 2025 GDB Online

7 27°C Light rain

Search

ENG IN

12:32

06-10-2025

The Dragon's Roar

A screenshot of the OnlineGDB IDE interface. The code editor shows a C++ file named main.cpp with the following content:

```
1 #include <iostream>
2 using namespace std;
3
4 void printNumbers(int n) {
5     if(n == 0) return;      // base case
6     printNumbers(n - 1);   // recursive call
7     cout << n << " ";    // print after recursion
8 }
9
10 int main() {
11     int n;
12     cin >> n;
13
14     printNumbers(n);
15     cout << endl;
16
17     return 0;
18 }
19
```

The input field contains the number 5. The output console shows the numbers 1, 2, 3, 4, 5. The status bar at the bottom right indicates the date and time as 06-10-2025 12:34.

The Hidden Chamber

The screenshot shows the OnlineGDB IDE interface. The code editor displays a C++ file named main.cpp with the following content:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int sumArray(const vector<int>& arr, int n) {
6     if(n == 0) return 0;// base case
7     return arr[n-1] + sumArray(arr, n-1);
8 }
9
10 int main() {
11     int n;
12     cin >> n;
13     vector<int> arr(n);
14     for(int i=0;i<n;i++) cin >> arr[i];
15
16     cout << sumArray(arr, n) << endl;
17     return 0;
18 }
19
```

The output window below shows the input "4\n1 2 3 4\n10" and the resulting output "...Program finished with exit code 0".

The browser address bar shows "onlinegdb.com/online_c++_compiler#". The status bar at the bottom right indicates "12:36 06-10-2025".

The Ancient Scroll

The screenshot shows the OnlineGDB IDE interface. The main window displays a C++ code editor with the file 'main.cpp' open. The code implements a linear search algorithm. The IDE includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language setting is set to C++. Below the code editor is a terminal window labeled 'input' containing the numbers 4, 2, 5, 7, 8, and 7, which are likely test inputs for the search function. The left sidebar contains links for My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login. At the bottom, there are links for About, FAQ, Blog, Terms of Use, Contact Us, GDB Tutorial, Credits, Privacy, and a copyright notice for 2016-2025.

```
#include <iostream>
#include <vector>
using namespace std;

int search(const vector<int>& arr, int key) {
    for(int i=0;i<arr.size();i++) {
        if(arr[i]==key) return i;
    }
    return -1; // not found
}

int main() {
    int n, key;
    cin >> n;
    vector<int> arr(n);
    for(int i=0;i<n;i++) cin >> arr[i];
    cin >> key;
    cout << search(arr, key) << endl;
    return 0;
}
```

input

```
4
2 5 7 8
7
```

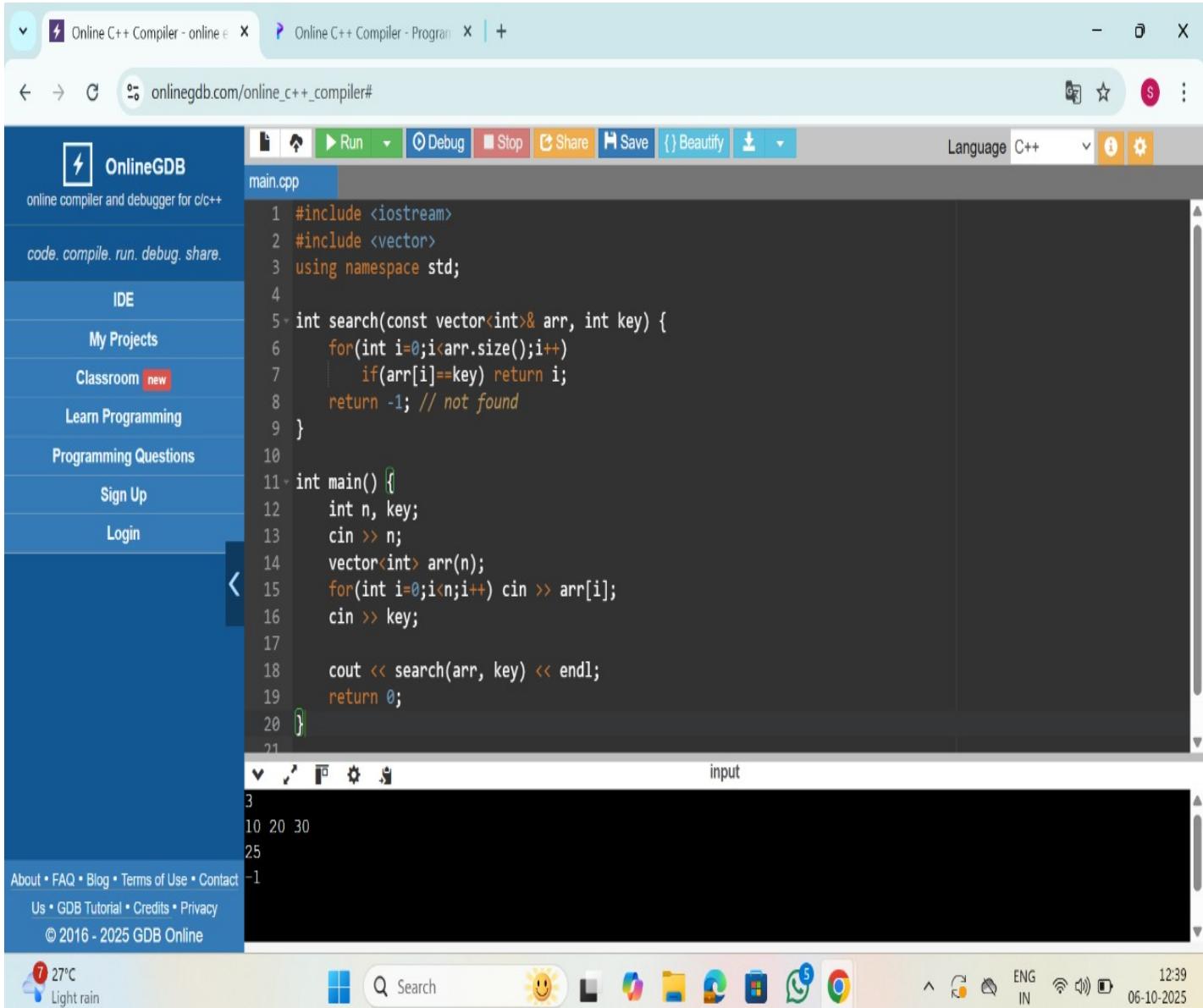
About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online

27°C Light rain

Search

12:38 06-10-2025

The Farmer's Basket



A screenshot of the OnlineGDB IDE interface. The main window shows a C++ code editor with the file 'main.cpp' containing a binary search algorithm. The code is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int search(const vector<int>& arr, int key) {
6     for(int i=0;i<arr.size();i++)
7         if(arr[i]==key) return i;
8     return -1; // not found
9 }
10
11 int main() {
12     int n, key;
13     cin >> n;
14     vector<int> arr(n);
15     for(int i=0;i<n;i++) cin >> arr[i];
16     cin >> key;
17
18     cout << search(arr, key) << endl;
19     return 0;
20 }
```

The 'input' field contains the numbers 3, 10, 20, 30, 25, and -1, which are the expected output of the search function.

The IDE sidebar includes links for 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. The status bar at the bottom shows weather information (27°C, Light rain), system icons, and the date/time (12:39, 06-10-2025).

The Secret Door

A screenshot of the OnlineGDB IDE interface. The main window shows a C++ code editor with the file 'main.cpp' containing a binary search algorithm. The code is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int binarySearch(const vector<int>& arr, int key) {
5     int low = 0, high = arr.size() - 1;
6     while(low <= high) {
7         int mid = low + (high - low) / 2;
8         if(arr[mid] == key) return mid;
9         else if(arr[mid] < key) low = mid + 1;
10        else high = mid - 1;
11    }
12    return -1;
13 }
14 int main() {
15     int n, key;
16     cin >> n;
17     vector<int> arr(n);
18     for(int i = 0; i < n; i++) cin >> arr[i];
19     cin >> key;
20     cout << binarySearch(arr, key) << endl;
21     return 0;
22 }
```

The IDE has a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The status bar at the bottom shows the input '5\n1 3 5 7 9' and the system tray with icons for weather, search, and various apps.

The Archer's Range

A screenshot of the OnlineGDB IDE interface. The main window shows a C++ code editor with the file 'main.cpp' open. The code implements a binary search algorithm to find the first occurrence of a key in a sorted array. The code is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int firstOccurrence(const vector<int>& arr, int key) {
5     int low = 0, high = arr.size() - 1;
6     int result = -1;
7     while(low <= high) {
8         int mid = low + (high - low) / 2;
9         if(arr[mid] == key) { result = mid;
10            high = mid - 1;
11        }
12        else if(arr[mid] < key) low = mid + 1;
13        else high = mid - 1;
14    }
15    return result;
16 }
17 int main() {
18     int n, key;
19     cin >> n;
20     vector<int> arr(n);
21     for(int i = 0; i < n; i++) cin >> arr[i];
22     cin >> key;
23     cout << firstOccurrence(arr, key) << endl;
24     return 0;
25 }
```

The input field contains the number 5, and the output field shows the result 1, indicating the first occurrence of the value 2 in the array [2, 2, 2, 3]. The IDE has a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C++. The left sidebar includes links for My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login.

The Treasure Chest

A screenshot of the OnlineGDB IDE interface. The main window shows a C++ code editor with the file 'main.cpp' containing a binary search algorithm. The code is as follows:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n, key;
7     cin >> n;
8     vector<int> arr(n);
9     for(int &x: arr) cin >> x;
10    cin >> key;
11
12    int low=0, high=n-1, res=-1;
13    while(low<=high){
14        int mid=low+(high-low)/2;
15        if(arr[mid]==key) res=mid, low=mid+1;
16        else if(arr[mid]<key) low=mid+1;
17        else high=mid-1;
18    }
19    cout << res << endl;
20 }
21
```

The IDE has a toolbar with various icons for file operations like Open, Save, and Run. The language is set to C++. Below the code editor is an input field containing '5 1 2 2 2 3'. The bottom status bar shows system information including battery level, signal strength, and the date/time '06-10-2025 12:47'.

The first index where element is greater than or equal to target

The screenshot shows the OnlineGDB IDE interface. The code editor displays a C++ file named main.cpp with the following content:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4
5 int main() {
6     int n, target;
7     cin >> n;
8     vector<int> arr(n);
9     for(int &x: arr) cin >> x;
10    cin >> target;
11
12    int low=0, high=n-1, res=n;
13    while(low<=high){
14        int mid=low+(high-low)/2;
15        if(arr[mid]>=target) res=mid, high=mid-1;
16        else low=mid+1;
17    }
18    cout << res << endl;
19 }
20
```

The input window contains the following data:

```
6
1 2 4 6 6 8
6
3
```

The output window shows the result:

```
...Program finished with exit code 0
```

The status bar at the bottom right shows the date and time as 06-10-2025 12:49.

The first index where element is strictly greater than target

Screenshot of OnlineGDB IDE showing C++ code for finding the upper bound of a target value in a sorted array.

Code:

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int main() {
6     vector<int> arr;
7     int x;
8     cout << "Enter array elements (space-separated): ";
9     while (cin >> x) arr.push_back(x);
10    cin.clear();
11    cin.ignore(1000, '\n');
12    int target;
13    cout << "Enter target: ";
14    cin >> target;
15    sort(arr.begin(), arr.end());
16    int l = 0, r = arr.size();
17    while (l < r) {
18        int m = l + (r - l) / 2;
19        if (arr[m] <= target) l = m + 1;
20        else r = m;
21    }
22    cout << "Upper bound index = " << l << endl;
23 }
```

Input:

```
Enter array elements (space-separated): 1 2 4 6 6 8
Enter target value: 6
Upper bound index = 5
```

Advertisement:

Our sales are up by 300%

Raghunandan Saraf, CEO
Saraf Furniture

Learn more

OnlineGDB Sidebar:

- IDE
- My Projects
- Classroom new
- Learn Programming
- Programming Questions
- Sign Up
- Login

Bottom Navigation:

- About • FAQ • Blog • Terms of Use • Contact
- Us • GDB Tutorial • Credits • Privacy
- © 2016 - 2025 GDB Online

System Status:

- 25°C Mostly cloudy
- Search bar
- Taskbar icons: File Explorer, OneDrive, Google Chrome, Microsoft Edge, File Manager, Task View, Taskbar notifications (7), and a red circle icon.
- System tray: Battery, Network, Volume, Language (ENG US), and Date/Time (07-10-2025).

The smallest element >=target

The screenshot shows the OnlineGDB IDE interface. The code editor window displays the following C++ program:

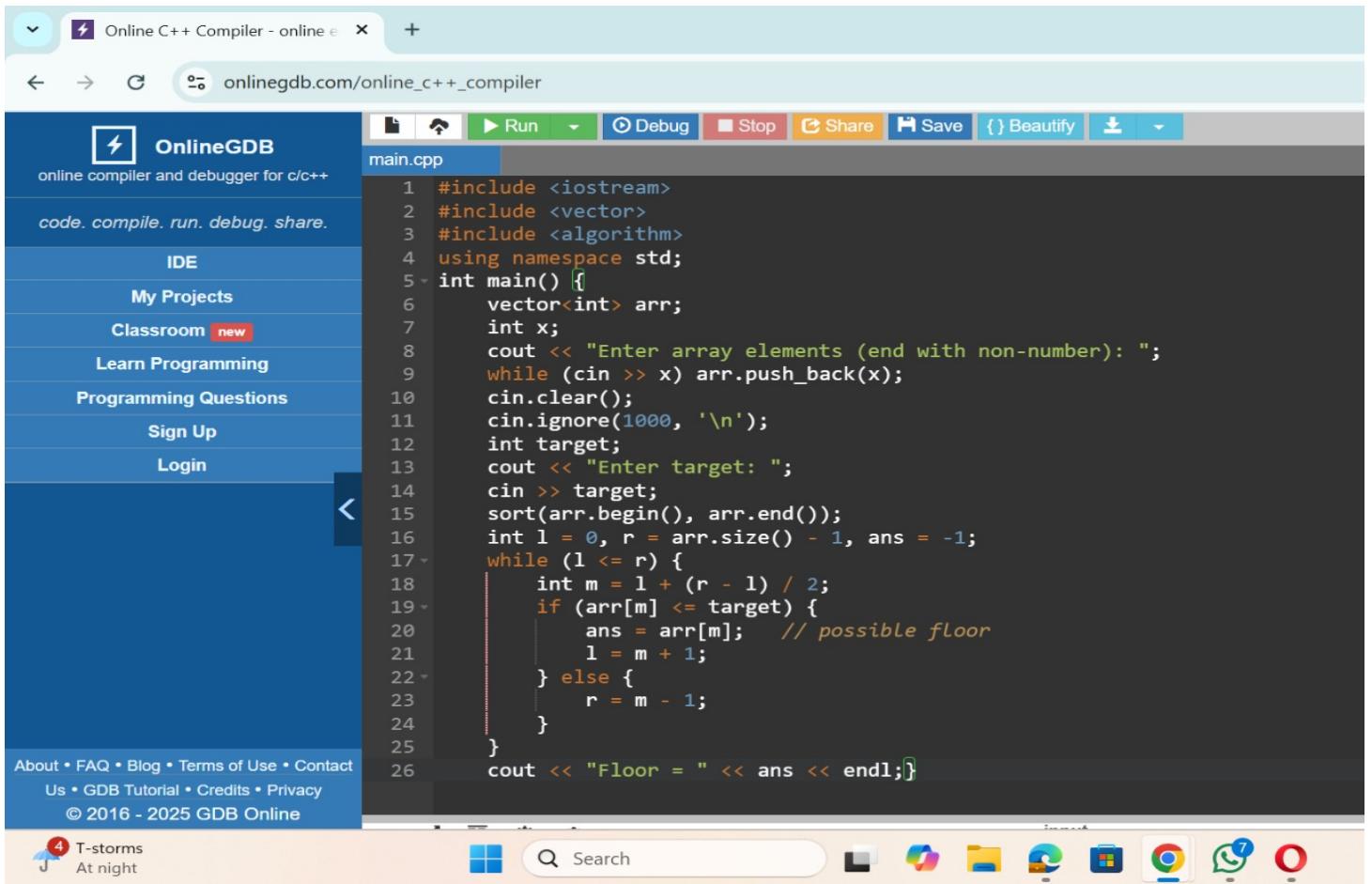
```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int main() {
6     vector<int> arr;
7     int x;
8     cout << "Enter array elements (end with non-number): ";
9     while (cin >> x) arr.push_back(x);
10    cin.clear();
11    cin.ignore(1000, '\n');
12    int target;
13    cout << "Enter target: ";
14    cin >> target;
15    sort(arr.begin(), arr.end());
16    int l = 0, r = arr.size() - 1, ans = -1;
17    while (l <= r) {
18        int m = l + (r - l) / 2;
19        if (arr[m] >= target) {
20            ans = arr[m]; // possible ceil
21            r = m - 1;
22        } else {
23            l = m + 1;
24        }
25    }
26    cout << "Ceil = " << ans << endl;}
```

The status bar at the bottom shows the output of the program:

```
Enter array elements (end with non-number): 1 2 4 6 6 8 q
Enter target: 5
Ceil = 6

...Program finished with exit code 0
Press ENTER to exit console.
```

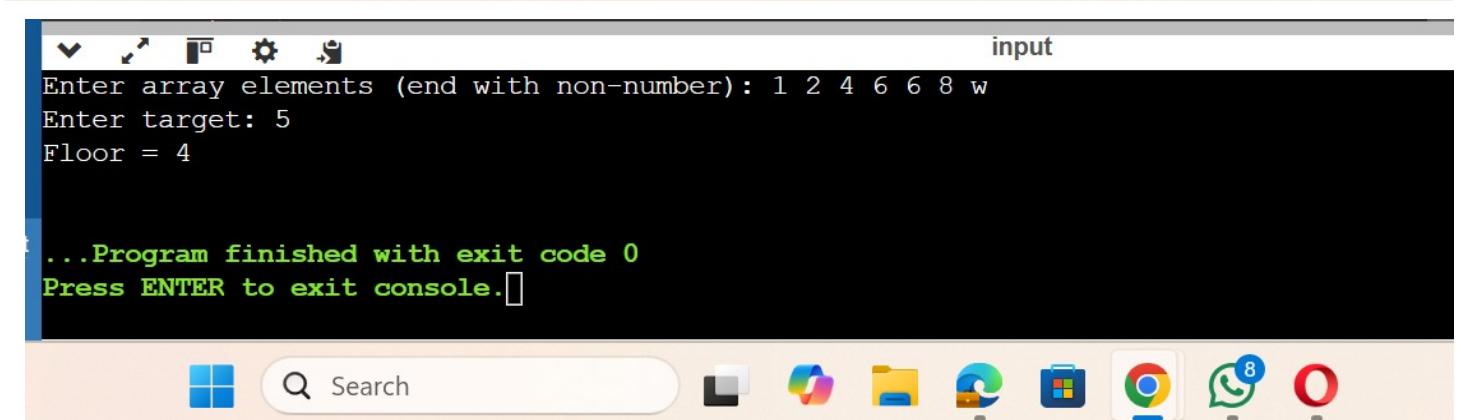
The largest element \leq target



The screenshot shows the OnlineGDB IDE interface. The code editor window displays a C++ file named main.cpp with the following content:

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5 int main() {
6     vector<int> arr;
7     int x;
8     cout << "Enter array elements (end with non-number): ";
9     while (cin >> x) arr.push_back(x);
10    cin.clear();
11    cin.ignore(1000, '\n');
12    int target;
13    cout << "Enter target: ";
14    cin >> target;
15    sort(arr.begin(), arr.end());
16    int l = 0, r = arr.size() - 1, ans = -1;
17    while (l <= r) {
18        int m = l + (r - 1) / 2;
19        if (arr[m] <= target) {
20            ans = arr[m]; // possible floor
21            l = m + 1;
22        } else {
23            r = m - 1;
24        }
25    }
26    cout << "Floor = " << ans << endl;
```

The browser address bar shows the URL: onlinegdb.com/online_c++_compiler. Below the browser is a taskbar with various icons.



The terminal window shows the output of the program. It prompts for array elements and a target value, then prints the floor value. The terminal also shows the exit code and a message to press Enter to exit.

```
Enter array elements (end with non-number): 1 2 4 6 6 8 w
Enter target: 5
Floor = 4

...Program finished with exit code 0
Press ENTER to exit console.[]
```

The taskbar at the bottom of the screen includes icons for File Explorer, File History, Task View, Microsoft Edge, Google Chrome, WhatsApp, and Opera.

The Treasure Map(Linear Search)

The screenshot shows a web-based IDE interface for OnlineGDB. The left sidebar contains links for the OnlineGDB logo, code compilation, sharing, and various user accounts. The main workspace is divided into sections for file operations, toolbars, and code editors. The code editor displays a C++ file named 'main.cpp' containing a linear search algorithm. The terminal window at the bottom shows the execution of the program, which prompts for matrix dimensions, elements, and a target value, then outputs 'Yes' indicating the target was found.

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int main() {
5     int n, m, target;
6     cout << "Enter rows and columns: ";
7     cin >> n >> m;
8     vector<vector<int>> matrix(n, vector<int>(m));
9     cout << "Enter matrix elements:\n";
10    for (int i = 0; i < n; i++) {
11        for (int j = 0; j < m; j++)
12            cin >> matrix[i][j];
13    cout << "Enter target: ";
14    cin >> target;
15    bool found = false;
16    for (int i = 0; i < n && !found; i++)
17        for (int j = 0; j < m; j++)
18            if (matrix[i][j] == target) {
19                found = true;
20                break;
21            }
22    cout << (found ? "Yes" : "No") << endl;
23    return 0;
24 }
```

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

input

```
Enter rows and columns: 3 3
Enter matrix elements:
1 2 3
4 5 6
7 8 9
Enter target: 5
Yes

...Program finished with exit code 0
Press ENTER to exit console.
```

The Magical Scrolls (Linear search return index)

The screenshot shows a web-based online compiler interface for C/C++. The code editor window displays a file named 'main.cpp' containing the following C++ code:

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int n, m, target;
    cout << "Enter rows and columns: ";
    cin >> n >> m;
    vector<vector<int>> matrix(n, vector<int>(m));
    cout << "Enter matrix elements:\n";
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> matrix[i][j];
        }
    }
    cout << "Enter target: ";
    cin >> target;
    int row = -1, col = -1;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (matrix[i][j] == target) {
                row = i;
                col = j;
                break;
            }
        }
        if (row != -1) break;
    }
    cout << "(" << row << "," << col << ")" << endl;
    return 0;
}
```

The browser's address bar shows the URL https://onlinegdb.com/online_c%2B%2B_compiler. Below the code editor, the browser's taskbar and system tray are visible, showing various open windows and system icons.

The Battle formation(Binary search)

The screenshot shows a Windows desktop environment with a browser window open to the OnlineGDB website. The browser title bar reads "Online C++ Compiler - online". The main content area displays a C++ IDE interface. On the left, a sidebar menu includes "My Projects", "Classroom", "Learn Programming", "Programming Questions", "Sign Up", and "Login". The central workspace shows a file named "main.cpp" with the following code:

```
1 #include <iostream>
2 #include <vector>
3 using namespace std;
4 int main() {
5     int n, m, target;
6     cout << "Enter rows and columns: ";
7     cin >> n >> m;
8     vector<vector<int>> matrix(n, vector<int>(m));
9     cout << "Enter matrix elements (sorted row-wise):\n";
10    for (int i = 0; i < n; i++) {
11        for (int j = 0; j < m; j++)
12            cin >> matrix[i][j];
13    cout << "Enter target: ";
14    cin >> target;
15    int low = 0, high = n * m - 1;
16    bool found = false;
17    while (low <= high) {
18        int mid = low + (high - low) / 2;
19        int val = matrix[mid / m][mid % m];
20        if (val == target) {
21            found = true;
22            break;
23        } else if (val < target)
24            low = mid + 1;
25        else
26            high = mid - 1;
27    }
28    cout << (found ? "True" : "False") << endl;
29    return 0;
}
```

Below the code editor is a terminal window titled "input" showing the execution of the program. The user enters the matrix size (3x3), matrix elements (1 3 5; 7 10 11; 16 20 30), and the target value (10). The output shows "True", indicating the target was found. The terminal also displays the standard exit message.

```
Enter rows and columns: 3 3
Enter matrix elements (sorted row-wise):
1 3 5
7 10 11
16 20 30
Enter target: 10
True

...Program finished with exit code 0
Press ENTER to exit console.[]
```

The Queen's jewels (Binary Search First occurrence)

The screenshot shows the OnlineGDB IDE interface. The main window displays a C++ program named `main.cpp`. The code implements a binary search algorithm to find the first occurrence of a target value in a $n \times m$ matrix. The user inputs n , m , and the matrix elements, followed by the target value. The program then performs a binary search on each row to find the first index where the value matches the target. If found, it prints the coordinates; otherwise, it prints $(-1, -1)$.

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    int n, m, target;
    cin >> n >> m;
    vector<vector<int>> mat(n, vector<int>(m));
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> mat[i][j];
    cin >> target;
    int l = 0, r = n * m - 1, ans = -1;
    while (l <= r) {
        int mid = l + (r - l) / 2;
        int val = mat[mid / m][mid % m];
        if (val >= target) {
            if (val == target) ans = mid; // possible first occurrence
            r = mid - 1;
        } else {
            l = mid + 1;
        }
    }
    if (ans != -1)
        cout << "(" << ans / m << ", " << ans % m << ")" << endl;
    else
        cout << "(-1,-1)" << endl;
}
```

The terminal output shows the input values and the resulting coordinates of the first jewel found.

```
3 3
1 2 2
3 4 4
5 6 7
4
(1,1)

...Program finished with exit code 0
```

The Hidden Scrolls(staircase search)

The screenshot shows the OnlineGDB IDE interface. The main window displays a C++ program named `main.cpp`. The code implements a staircase search algorithm to find a target value in a matrix. The input consists of matrix dimensions n and m , followed by the matrix elements and a target value. The output indicates whether the target was found.

```
int main() {
    int n, m, target;
    cin >> n >> m;
    vector<vector<int>> mat(n, vector<int>(m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            cin >> mat[i][j];
    }
    cin >> target;

    int i = 0, j = m - 1;
    bool found = false;
    while (i < n && j >= 0) {
        if (mat[i][j] == target) { found = true; break; }
        else if (mat[i][j] > target) j--;
        else i++;
    }

    cout << (found ? "True" : "False") << endl;
}
```

Input:

```
4 4
1 4 7 11
2 5 8 12
3 6 9 16
10 13 14 17
6
True
```

Output:

```
True
```

The Magic Portal (Binary Search 2D)

The screenshot shows the OnlineGDB IDE interface. The code editor displays a C++ program named `main.cpp` for a 2D binary search problem. The input section contains sample data, and the output section shows the program's response.

```
int main() {
    int n, m, target;
    cin >> n >> m;
    vector<vector<int>> mat(n, vector<int>(m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++)
            cin >> mat[i][j];
        cin >> target;
    }

    int i = 0, j = m - 1;
    bool found = false;
    while (i < n && j >= 0) {
        if (mat[i][j] == target) { found = true; break; }
        else if (mat[i][j] > target) j--;
        else i++;
    }

    cout << (found ? "Activated" : "Failed") << endl;
}
```

input

```
3 3
1 2 8
3 6 10
7 9 12
9
```

output

```
Activated
```

OnlineGDB online compiler and debugger for c/c++

code. compile. run. debug. share.

IDE

- My Projects
- Classroom new
- Learn Programming
- Programming Questions
- Sign Up
- Login

About • FAQ • Blog • Terms of Use • Contact
Us • GDB Tutorial • Credits • Privacy
© 2016 - 2025 GDB Online

24°C
Mostly cloudy

Search

ENG US