

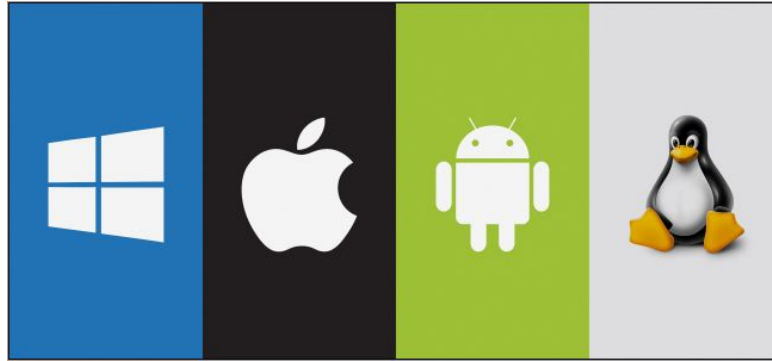
[salamusk.onrender.com](https://salamusk.onrender.com)

# Developer and releases



Bram Moolenaar

# Why Vim?



Cross platforms

# Why Vim?

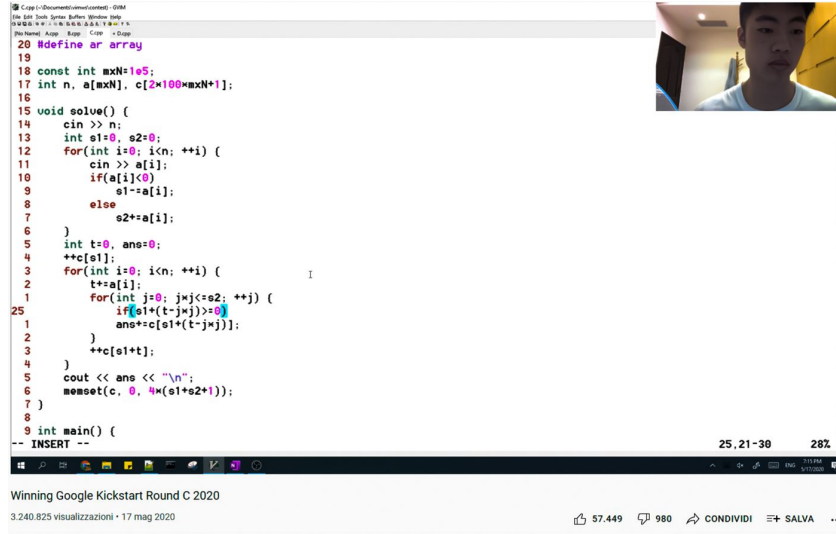


Ubiquitous



[salamusk.onrender.com/VIM/index](https://salamusk.onrender.com/VIM/index)

# Why Vim?



```
20 #define ar array
19
18 const int mxN=1e5;
17 int n, a[mxN], c[2*100*mxN+1];
16
15 void solve() {
14     cin >> n;
13     int s1=0, s2=0;
12     for(int i=0; i<n; ++i) {
11         cin >> a[i];
10         if(a[i]<0)
9             s1-=a[i];
8         else
7             s2+=a[i];
6     }
5     int t=0, ans=0;
4     ++c[s1];
3     for(int i=0; i<n; ++i) {
2         t+=a[i];
1         for(int j=0; j*j<=s2; ++j) {
25             if(s1+(t-j*j)>=0)
1                 ans+=c[s1+(t-j*j)];
2             ++c[s1+t];
3         }
4     }
5     cout << ans << "\n";
6     memset(c, 0, 4*(s1+s2+1));
7 }
8
9 int main() {
-- INSERT --
```

Winning Google Kickstart Round C 2020  
3,240,825 visualizzazioni · 17 mag 2020

57,449 980 CONDIVIDI SALVA ...

Look at this guy winning Google Kickstart using Vim shortcuts!

# Vim introduction and modes

- Normal mode
- Command mode
- Insert mode
- Visual mode
- Replace mode

# What's the file .vimrc ?

All the plugins and macros that allow the customization of the workspace are stored in this file.

- Basic version
- Awesome version

[salamusk.onrender.com/tuto](https://salamusk.onrender.com/tuto)



# Basic movements: h, j, k, and l

In contrast to regular text editor, you use keys h, j, k, and l instead of arrow keys to move the cursor.

# Word movement: w, e, b

To navigate the text in terms of words, you can use keys w, b, and e (also W, B, E in real Vim).

w moves to the start of next word; e moves to the end of the word; and b moves to beginning of the word.

# Insert text repeatedly, e.g. 3iYes

You can insert text multiple times. For example, an underline of a header might consist of 30 -s.

-----

With 30i- Esc, there's no need to press - 30 times.

Let's try it out: insert go three times.

# Find a character, f and F

To find and move to the next (or previous) occurrence of a character, use f and F, e.g. fo finds next o.

You can combine f with a number.

For example, you can find 3rd occurrence of 'q' with 3fq, que?

# Go to matching parentheses, %

In text that is structured with parentheses or brackets, ( or { or [, use % to jump to the matching parenthesis or bracket. Here is (a sample) text to try that.

# Number powered movement, e.g. 5w

Moving within the text is not limited to individual keys; you can combine movement keys with a number. For example, 3w is the same as pressing w three times.

# Go to start/end of line, 0 and \$

To reach the beginning of a line, press 0. For the end of a line, there's \$

# Find word under cursor, \* and #

Find the next occurrence of the word under cursor with \*, and the previous with #.



# Goto line, gg and G

`gg` takes you to the beginning of the file; `G` to the end.

To jump directly to a specific line, give its line number along with `G`.

Now go to the beginning of this screen with `gg` and then back to end with `G`.

# Search, /text with n and N

Searching text is a vital part of any text editor.

In Vim, you press `/`, and give the text you are looking for.

You can repeat the search for next and previous occurrences with `n` and `N`, respectively.

For advanced use cases, it's possible to use **regexps** that help to find text of particular form (In real Vim).

Let's try a simple text search. Search for text and find the subsequent matches with `n`.

# Insert new line, o and O

To insert text into a new line, press `o` or `O`.

After new line is created, the editor is set to insert mode.

Write a bit and get back to normal mode.

# Removing a character, x and X

x and X delete the character under the cursor and to the left of the cursor, respectively.

Try pressing x to remove the last word.

# Replacing letter under cursor, r

When you need to replace only one character under your cursor, without changing to insert mode, use `r`.

# Deleting, d

`d` is the delete command.

You can combine it with movement, e.g. `dw` deletes the first word on the right side of the cursor.

e.g. `dd` deletes the line on the cursor.

It also copies the content, so that you can paste it with `p` to another location (on real Vim).

# Repetition with .

To repeat the previous command, just press .

First, remove two words with d2w.

After that, remove the rest of the words in this line with .

# Visual mode, v

Besides insert and normal mode, Vim has also visual mode.

In visual mode, you select text using movement keys before you decide what to do with it.

Let's see how.

Go to visual mode with v. Then select a word with e. After you've selected the text, you can delete it with d.

This sentence has not seen the light.



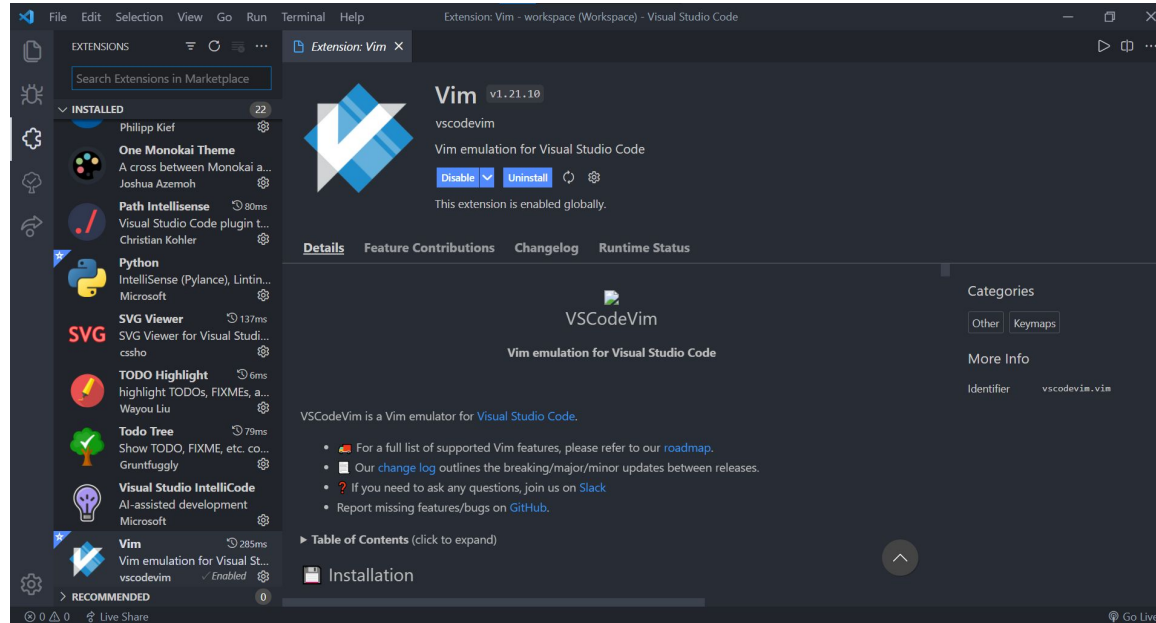
# How to create and use macros

- Enter command mode and press 'q'
- Press any key you want the macro to be assigned to
- Press any key you want the macro to reproduce
- Press 'q' and the key you assigned the macro to, to stop the macro
- Press '@' and 'shift+<macro-key>' to start the macro

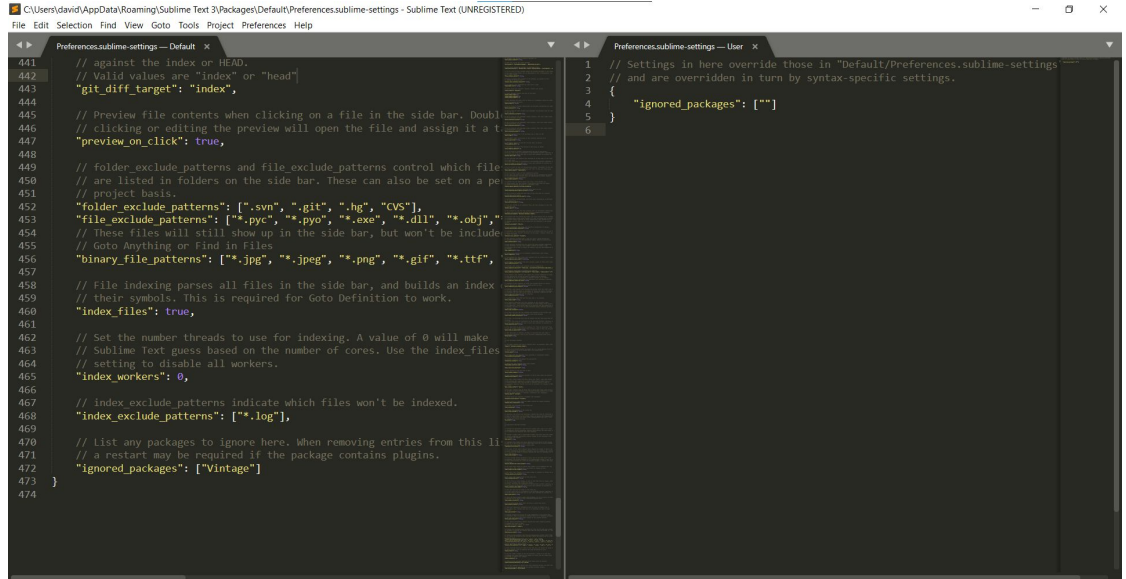
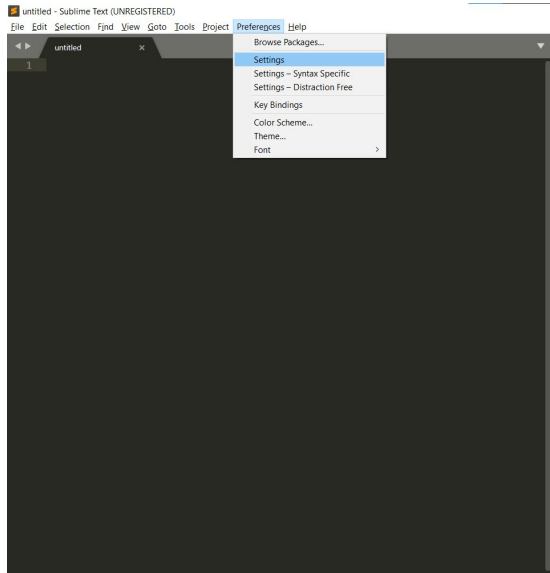
# Vim plugin

What are they?

# Plugin Visual Studio Code



# Plugin Sublime Text



# Fonti

<https://github.com/vim/vim/blob/master/README.md>