



Xi'an Jiaotong-Liverpool University

西交利物浦大學

## Lab 1

# Getting Started

# Hello World!

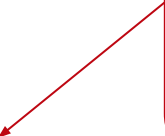
---

- Our first program

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

Our first C code is to simply print a string Hello World! in the screen



# Hello World!

---

- Our first program

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

For now, always start with these two lines

and always end with these

# Our First Lab Example (1)

---

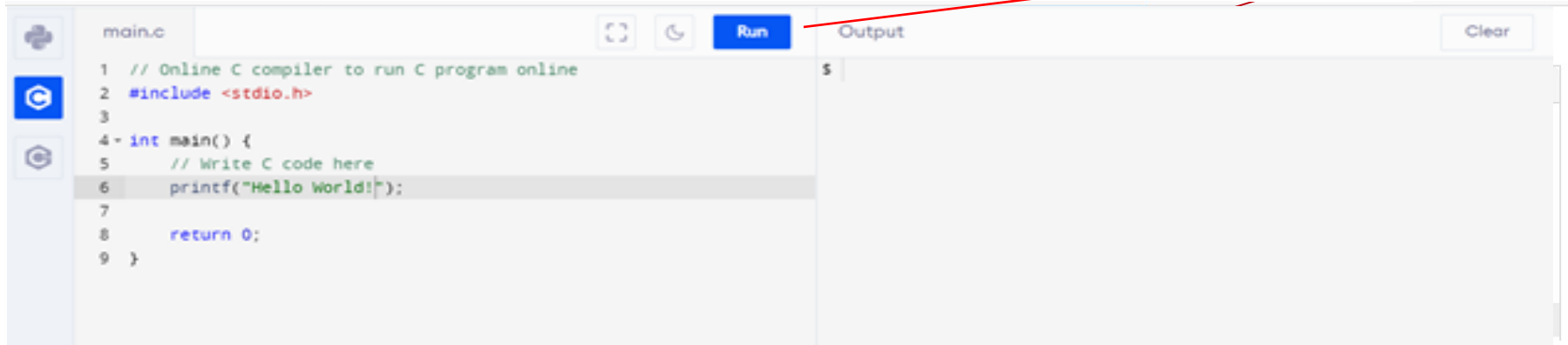
1. Use Firefox/Chrome, go to <https://codecast.france-ioi.org/v6/>
2. Type the code

```
#include <stdio.h>

int main() {
    printf("Hello World!");
    return 0;
}
```

3. Click compile

this button



## Our First Lab Example (2)

---

4. If the code compiles (no errors), run the program

click this run button

The screenshot shows a code editor interface with three main panels: Variables, Source, and Terminal. The Variables panel on the left shows 'main()' with up and down arrows. The Source panel in the center displays C code with line numbers 1 through 7. The code is: 

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello World!");
5     return 0;
6 }
7
```

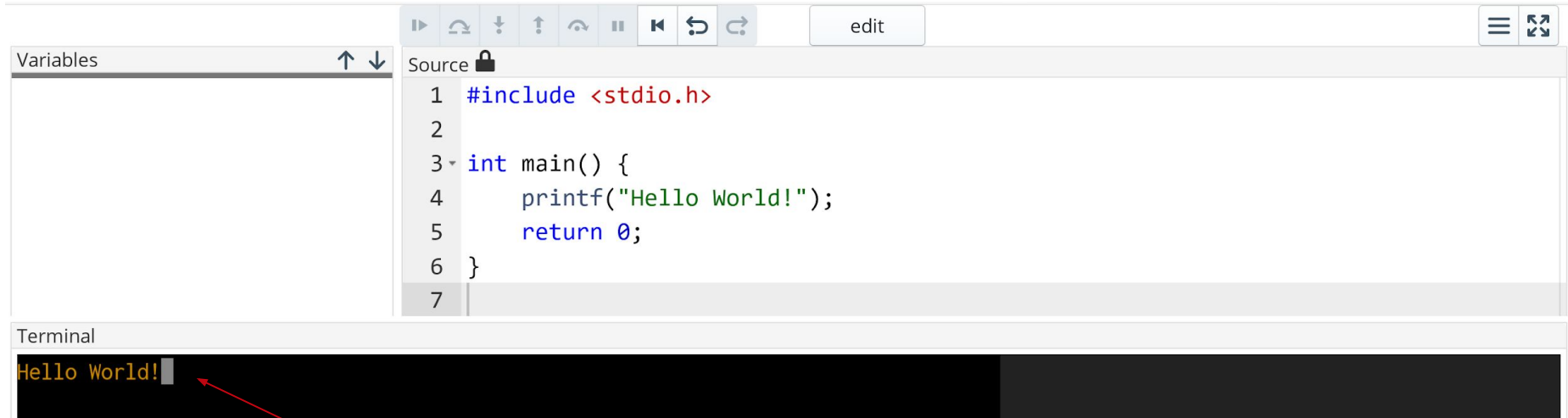
 Lines 4 and 5 are highlighted in green. Above the Source panel is a toolbar with icons for running, undo, redo, and other editing functions, along with an 'edit' button. A red callout box with the text 'click this run button' points to the first icon in the toolbar, which is a play button. The Terminal panel at the bottom is currently empty.

```
Variables
main()
Source
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello World!");
5     return 0;
6 }
7
Terminal
```

## Our First Lab Example (3)

---

5. The program runs and prints in the console/terminal



The screenshot shows a code editor interface. At the top, there is a toolbar with various icons for running and debugging the code, and an 'edit' button. Below the toolbar, the editor is divided into two main sections. The top section, labeled 'Source', contains a C program with the following code:

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello World!");
5     return 0;
6 }
7
```

The bottom section, labeled 'Terminal', shows the output of the program: 'Hello World!'. A red arrow points from a text box below the terminal to the output text.

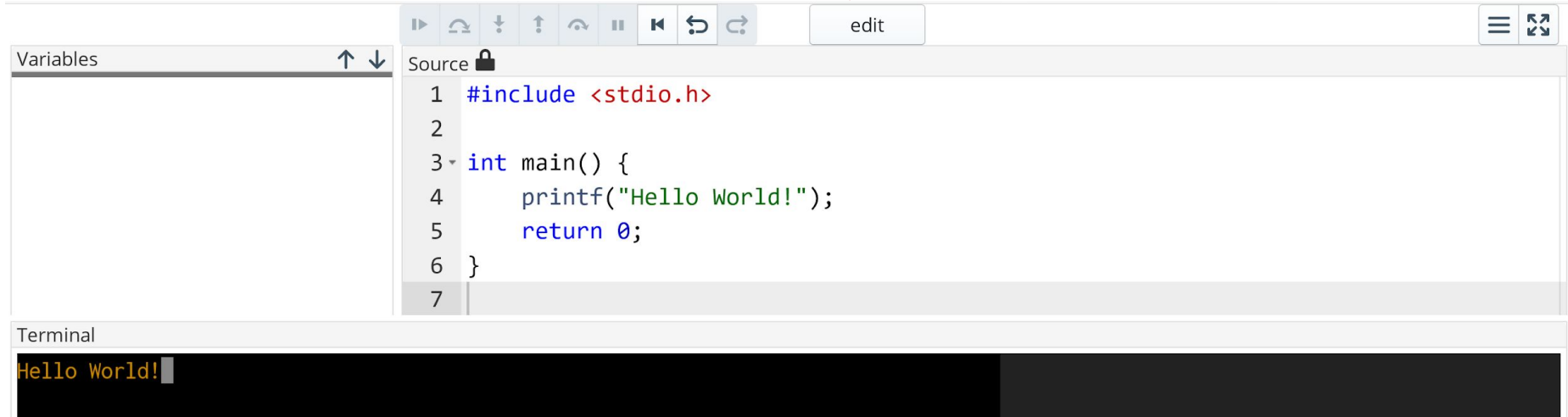
The string in printf is printed here

## Our First Lab Example (4)

---

6. Click edit if you want to continue write and debug your C code

this button



From now on, you will **write** and **test** your code using this **Codecast** !

# The First Lab Exercise

---

- We now continue with the Lab Exercise
  - Just like the Lab Example, you will use Codecast to write and test your code
  - However, since it is graded, you will also need to submit your code using the [LMO VPL](#).
- 
- In the next slide, you will see the Lab Exercise
  - Lab Exercise slides will consist of
    - Description of the problem
    - One or more test cases
    - Hints and skeleton code (optional)
- 
- After that, we will guide you to solve and submit the Lab Exercise



## Exercise Learning C in Linux

---

- Complete the code to print in 2 lines:   Hello World!  
  I'm learning "C in Linux" coding
  - Hint: Use the escape sequence \" and new line character \n
- Skeleton code:

```
#include <stdio.h>

int main() {

    return 0;
}
```

# Our First Lab Exercise (1)

---

1. Use Firefox/Chrome, open Codecast in <https://codecast.france-ioi.org/v6/>
2. Copy paste the skeleton code and complete the code in Codecast using `printf`
  - use the **escape sequence** `\"` in the string to print the double quote
  - use the **new line character** `\n` in the string to change line
  - IMPORTANT: from now on, always ends the last `printf` with `\n`
  - for example:

```
printf("Always end last \"printf\" with a newline\n");
```

the LMO VPL

- otherwise, the ICE VPL grader will mark your code incorrect !

## Our First Lab Exercise (2)

---

3. Continue write, test, edit your code in Codecast until it displays the correct 2 lines as specified in the Lab Exercise **description**
4. After you are satisfied with your code, go to LMO and click **LMO** VPL Lab Exercise **below**



Exercise - Learning C in Linux

# Our First Lab Exercise (3)

---

5. You will see the following screen, click Edit



## Exercise - Learning C in Linux

📄 Requested files: Ex.1.1.c ([Download](#))

Type of work: 👤 Individual work

Complete the code to print in 2 lines:

Hello World!

I'm learning "C in Linux coding"

## Requested files

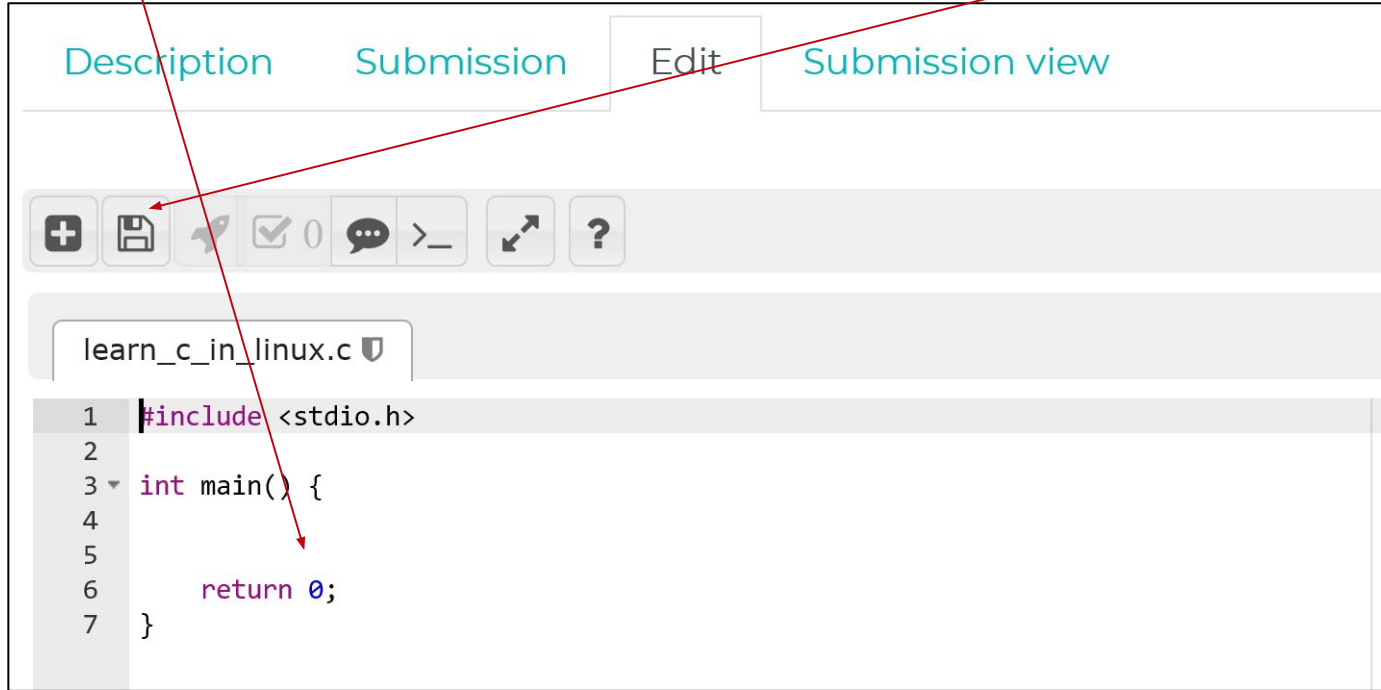
### Ex.1.1.c

```
1 #include <stdio.h>
2
3 int main() {
4
5
6     return 0;
7 }
```

## Our First Lab Exercise (4)

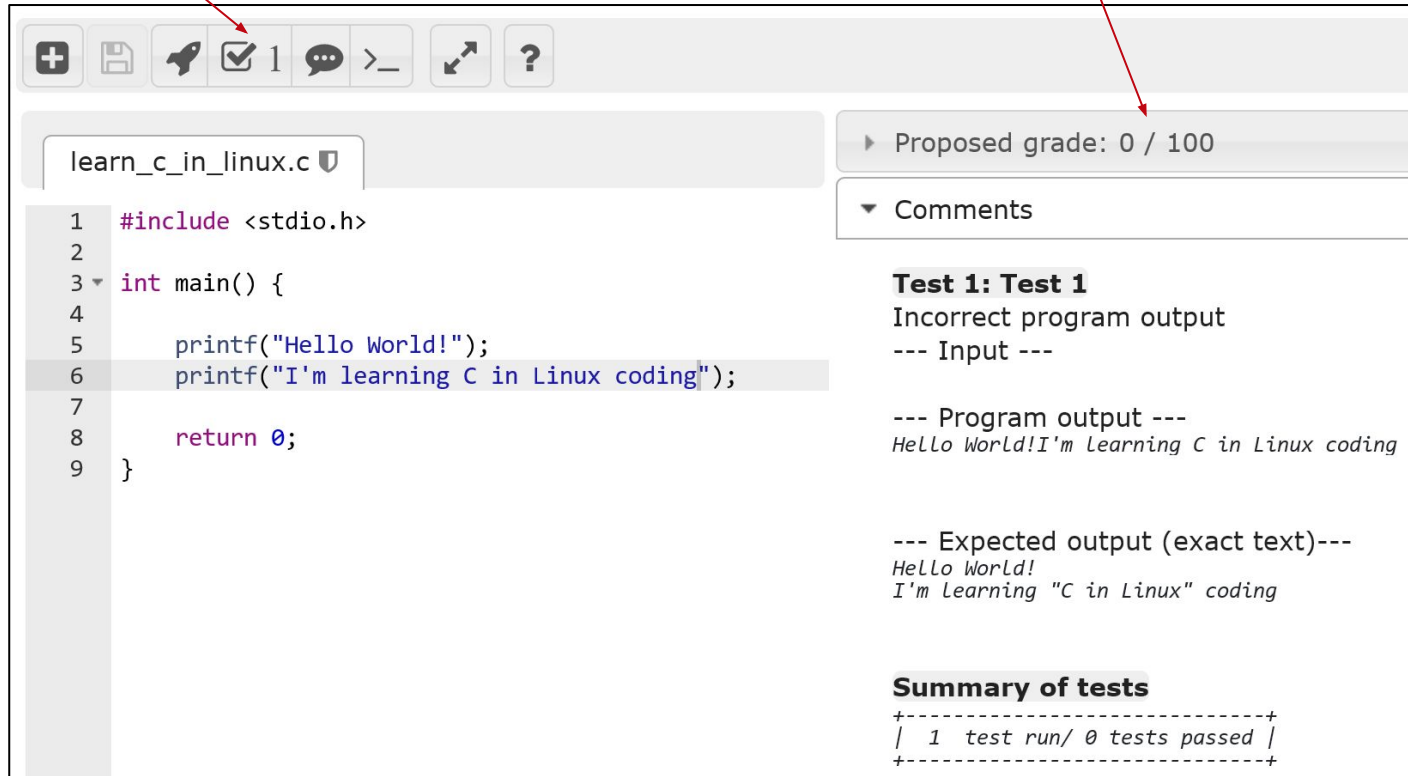
---

6. Copy Paste your code from Codecast, and then click Save



# Our First Lab Exercise (5)

7. Click Evaluate and then you will see the following result if you made mistake



The screenshot shows a web-based C programming environment. At the top, a toolbar contains icons for adding files, saving, running, evaluating (checked), comments, and help. A red arrow points from the text 'Click Evaluate' to the evaluate icon. Below the toolbar, the file name 'learn\_c\_in\_linux.c' is displayed. The code editor shows a C program with a syntax error: a missing closing quote in the second printf statement. A red arrow points from the text 'if you made mistake' to the 'Proposed grade: 0 / 100' status bar. The right sidebar displays the test results for 'Test 1: Test 1', indicating an 'Incorrect program output' and showing the difference between the actual and expected program outputs.

```
1  #include <stdio.h>
2
3  int main() {
4
5      printf("Hello World!");
6      printf("I'm learning C in Linux coding");
7
8      return 0;
9  }
```

Proposed grade: 0 / 100

Comments

**Test 1: Test 1**  
Incorrect program output  
--- Input ---  
  
--- Program output ---  
Hello World!I'm Learning C in Linux coding

--- Expected output (exact text)---  
Hello World!  
I'm Learning "C in Linux" coding

**Summary of tests**

1	test run/ 0 tests passed
---	--------------------------

## Our First Lab Exercise (6)

8. You can go back to code, debug, and test again in Codecast, and return to [LMO](#) VPL, Copy Paste, Save, Evaluate again; you get 100 if you pass all test cases

The screenshot displays the Codecast IDE interface. At the top is a toolbar with icons for adding files, saving, running, checking, commenting, and other functions. Below the toolbar, the file name 'learn\_c\_in\_linux.c' is shown. The main editor area contains the following C code:

```
1 #include <stdio.h>
2
3 int main() {
4
5
6
7
8     return 0;
9 }
```

Red arrows point from the text in the exercise description to specific UI elements: one to the 'Save' icon, one to the 'Run' icon, one to the 'Evaluate' button, and one to the 'code removed' box.

On the right side, a panel shows the 'Proposed grade: 100 / 100' and a 'Comments' section. Below this is a 'Summary of tests' section with the following output:

```
+-----+
| 1 test run/ 1 test passed |
+-----+
```

A red box labeled 'code removed' is positioned at the bottom right of the editor area.