GECG10069 (561085) F25: Introduction to Programming (C++)

Lab 11: Data Manipulation

What you will learn from Lab 11

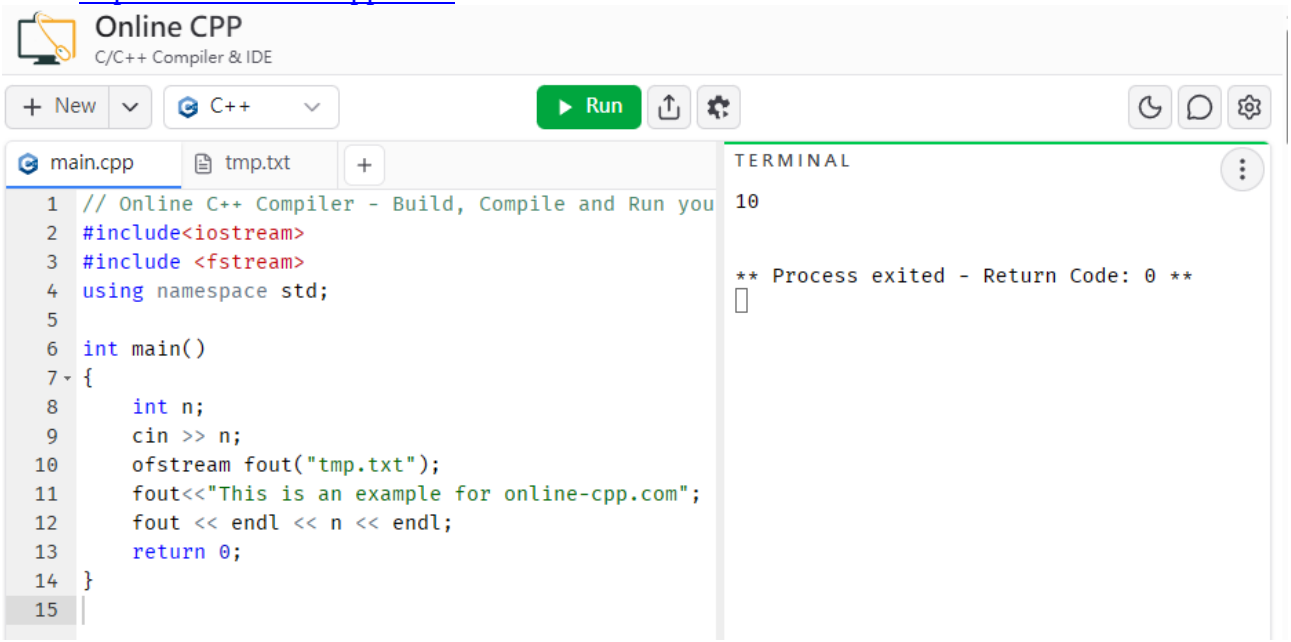
Output format and how to output file

TASK 11-1: OUTPUT FILE ON ONLINE-CPP

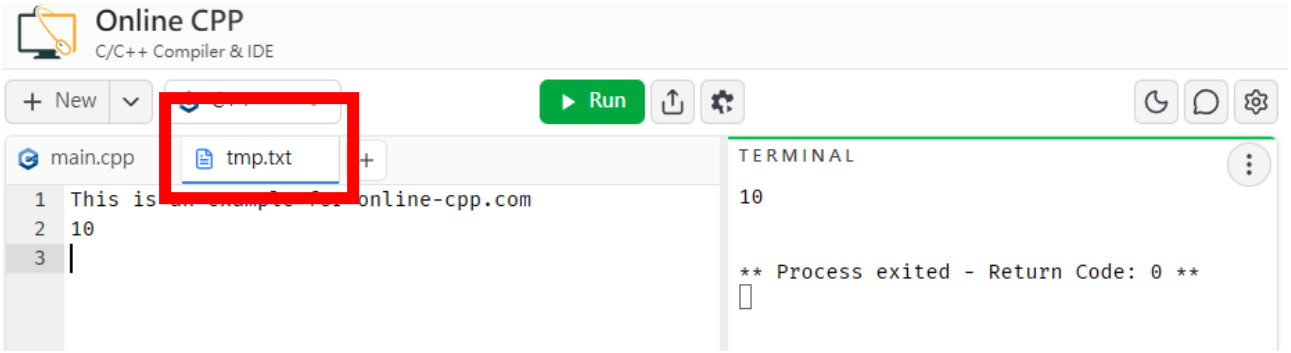
```
#include<iostream>
#include <fstream>
using namespace std;

int main()
{
    int n;
    cin >> n;
    ofstream fout("tmp.txt");
    fout <<"This is an example for online-cpp.com" << endl;
    fout << n << endl;
    return 0;
}
```

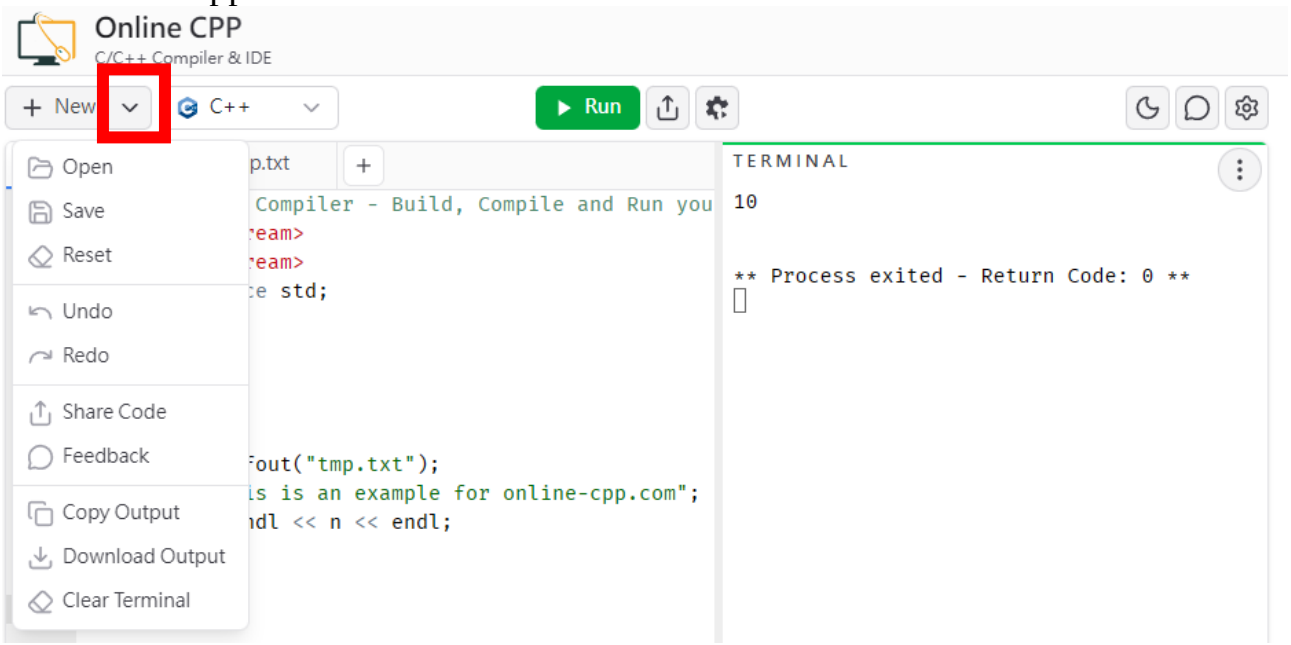
Web: https://www.online-cpp.com/



First click run, TERMINAL on the right used for cin/cout (the cin value can be enter after run the program). After running, the output file will appear on the tab.



To save the .cpp code



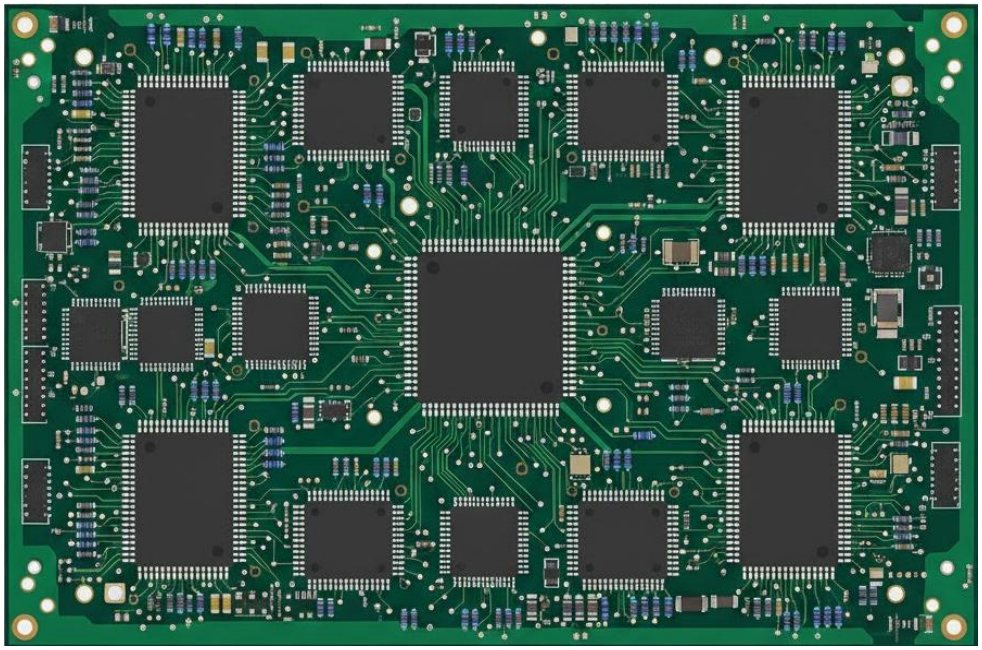
EXERCISE 11-1 : PCB SPEC CHECKER

Description -

Please read the given csv file, check whether the PCB design satisfies two given specifications, and print the results.

Specification 1: The average IC Power must be no more than 3 W.

Specification 2: The total IC area should be no greater than half of the PCB area.



Input CSV Format

- The first line contains three integers, representing the PCB's width, PCB's Height, and number of ICs on the PCB.

PCB_Width, PCB_Length, IC_Num

- Then, for N ICs, the next N lines each describe one IC's information:

IC_Power (double), IC_Area (integer)

Output Format

- Your Program should print two lines:

Average IC Power = aa.aa, SPEC1 PASS/FAIL Total IC Area = xxx, SPEC2 PASS/FAIL

Sample Test Cases -

Sample Input - 1 (ex_11_1.csv)
200,100,4 2.5,2000

3.5,1500 2.0,3000 4.0,3500
Sample Output - 1
Average IC Power = 3.00, SPEC1 PASS Total IC Area = 10000, SPEC2 PASS
Sample Input - 2 (ex_11_2.csv)
300 , 300 , 8 4.0 , 5000 3.5, 5000 2.0 ,5000 3.0, 5000 3.5 , 5000 4.0, 5000 2.5, 5000 2.5 , 5000
Sample Output - 2
Average IC Power = 3.12, SPEC1 FAIL Total IC Area = 40000, SPEC2 PASS

Requirements / Notes

- You **must** use stringstream to parse each line from the CSV.
- Use stod() and stoi() for number conversion.
- PCB area = PCB_Width × PCB_Length.
- Output numeric values with fixed << setprecision(2) for power and integers for area.
- `Input size ≤ 100 ICs.`

EXERCISE 10-2 : JSON TRANSLATOR

Description -

Read three lines of user information from standard input — **Name**, **Age**, and **Gender** — and output to a simple JSON-like format, named sample.json. Note that the age should be printed as a hexadecimal base.

Input Format :

- Three lines:

```
<Name>
<Age>
<Gender>
```

Output Format :

```
{
  "name": "<Name>",
  "age": <Age>,
  "gender": "<Gender>"
}
```

Sample Testcases -

Sample Input - 1
Tralalero Tralala 999 male
Sample Output - 1
{   "name": "Tralalero Tralala",

```
  "age": 0x3e7,
  "gender": "male"
}
```

Sample Input - 2

```
Ballerina Cappuccino
10
female
```

Sample Output - 2

```
{
  "name": "Ballerina Cappuccino",
  "age": 0xa,
  "gender": "female"
}
```