



# Xử Lý File Trong C++

---

File

# Xử lý file là gì?

Xử lý file là tên gọi của tác vụ thao tác tập tin lưu trữ dữ liệu liên quan bằng một ngôn ngữ lập trình, cụ thể ở đây là C++. Điều này cho phép bạn lưu trữ dữ liệu trong kho lưu trữ vĩnh viễn, ngay cả sau khi chương trình này thực hiện xử lý file cho cùng một kết thúc thực thi của nó. C++ cung cấp thư viện **`fstream, ofstream, ifstream`** để xử lý file.



# File Text

## File text.

---



# Mở file

## Tính Chất:

- Trước khi thao tác với File bạn cần phải mở nó lên, tuy nhiên khi mở file lên bạn cần chỉ rõ mode mở file.
- Trong đó file\_name là tên của file bạn muốn mở, có thể mở file với đường dẫn tương đối hoặc tuyệt đối.
- Nếu bạn thao tác với file nằm chung thư mục với file mã nguồn thì bạn chỉ cần sử dụng đường dẫn tương đối thôi.

## Cú pháp:

```
open(file_name,mode)
```



# Mở File

## Tính Chất:

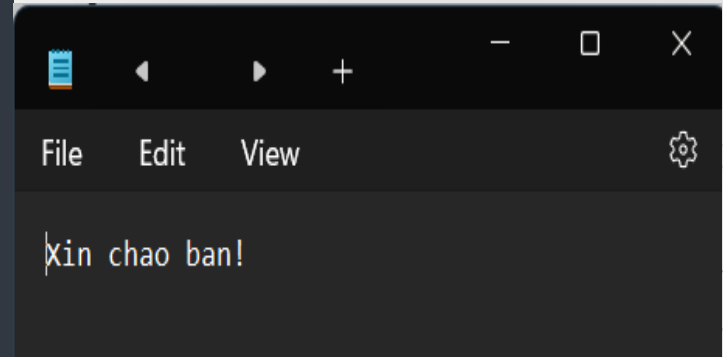
- Mỗi hàm open của 3 lớp ofstream, ifstream, istream đều có mode mặc định:
- Ofstream => ios::out
- ifstream => ios::in
- Istream => ios::in | ios::out



# Mở File Để Ghi

Chú ý: Nếu bạn mở file lên để ghi, nếu file này đã tồn tại thì nó sẽ ghi trên file này, ngược lại nếu file chưa tồn tại thì nó sẽ tạo một file mới.

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      ofstream fileOut;
6      fileOut.open("data.txt");
7      fileOut << "Xin chao ban!\n";
8      fileOut.close();
9      return 0;
10 }
```



# Mở File Để Ghi

- Sử dụng mode `ios::app` để ghi nội dung kế tiếp vào cuối file:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     ofstream fileOut;
6     fileOut.open("data.txt", ios::app | ios::out);
7     fileOut << "Tam biet cac ban!\n";
8     fileOut.close();
9     return 0;
10 }
```

File Edit View

Xin chao ban!  
Tam biet cac ban!

# Mở File ra để đọc:

- Khi bạn mở file lên để đọc thì nếu file bạn mở chưa tồn tại sẽ gây ra lỗi, bạn cần kiểm tra sự tồn tại của file trước khi đọc

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ifstream in;
    in.open("data.txt");
    if(in.is_open()){
        string s; in >> s;
        cout << s << endl;
    }
    else {
        cout << "File bị lỗi trong quá trình mở\n";
    }
    in.close();
    return 0;
}
```

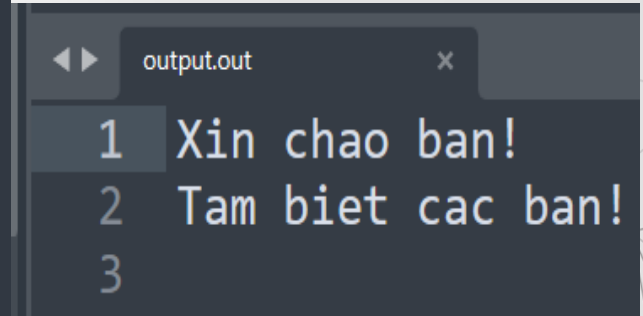




# Mở File ra để đọc:

- Để đọc hết mọi dòng trong file:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     ifstream in;
6     in.open("data.txt");
7     if(in.is_open()){
8         string s;
9         while(getline(in,s)){
10             cout << s << endl;
11         }
12     }
13     else {
14         cout << "File bi loi trong qua trinh mo\n";
15     }
16     in.close();
17     return 0;
18 }
```

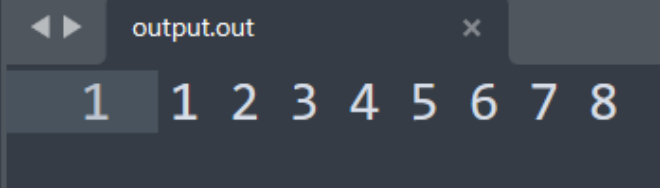


```
output.out
1 Xin chao ban!
2 Tam biet cac ban!
3
```

# Mở File ra để đọc:

- Để đọc mảng số nguyên từ file:

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main() {
5     ifstream in;
6     in.open("data.txt");
7     if(in.is_open()){
8         int n; in >> n;
9         vector<int> v(n);
10        for(int i = 0; i < n; i++){
11            in >> v[i];
12        }
13        for(int x : v){
14            cout << x << " ";
15        }
16    }
17    else {
18        cout << "File bi loi trong qua trinh mo\n";
19    }
20    in.close();
21    return 0;
22 }
```



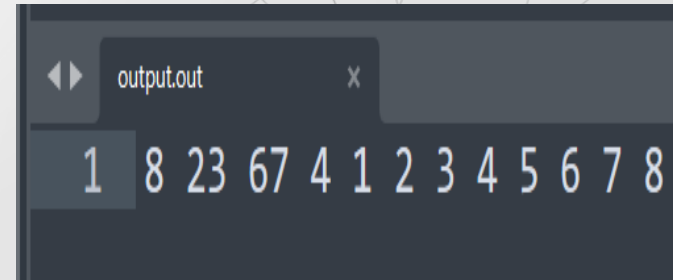
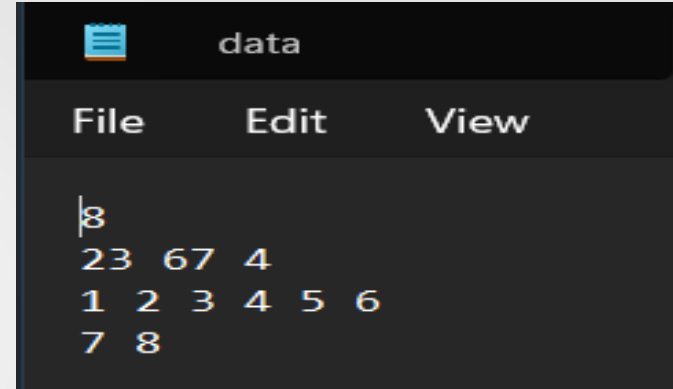
output.out

1 1 2 3 4 5 6 7 8

# Mở File ra để đọc:

- Đọc hết mọi số trong file khi không biết trước số lượng phần tử:

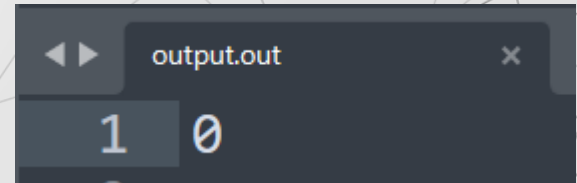
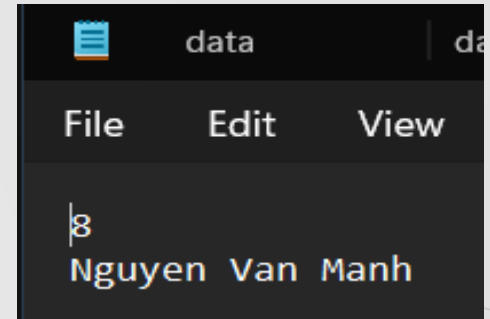
```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      ifstream in;
6      in.open("data.txt");
7      if(in.is_open()){
8          int tmp;
9          while(in >> tmp){
10             cout << tmp << " ";
11         }
12     }
13     else {
14         cout << "File bị lỗi trong quá trình mở\n";
15     }
16     in.close();
17     return 0;
18 }
```



# Mở File ra để đọc:

- Chú ý getline cũng sẽ bị trôi lệnh khi đọc file:

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      ifstream in;
6      in.open("data.txt");
7      if(in.is_open()){
8          int tmp; in >> tmp;
9          string s;
10         getline(in, s);
11         cout << s.size() << endl;
12     }
13     else {
14         cout << "File bi loi trong qua trinh mo\n";
15     }
16     in.close();
17     return 0;
18 }
```





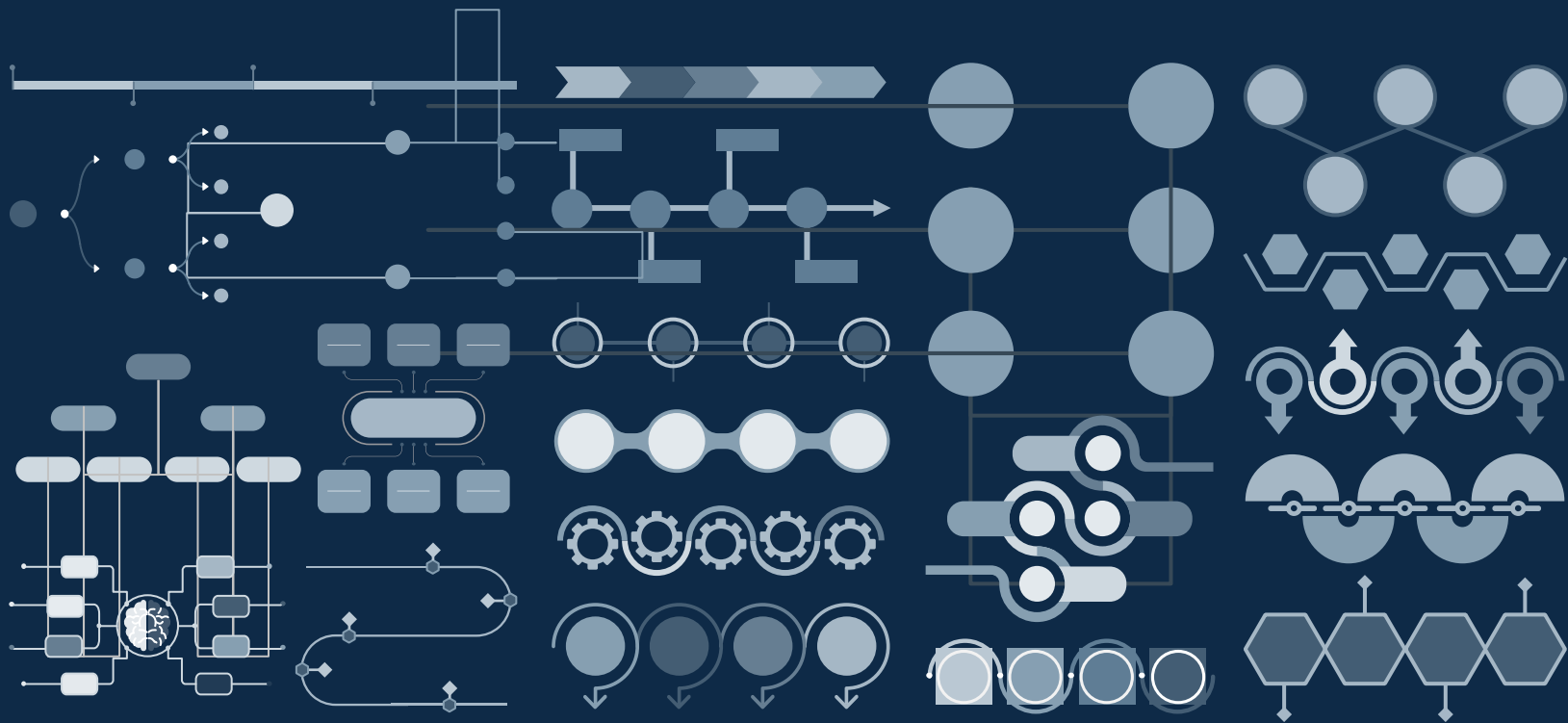
# THANKS

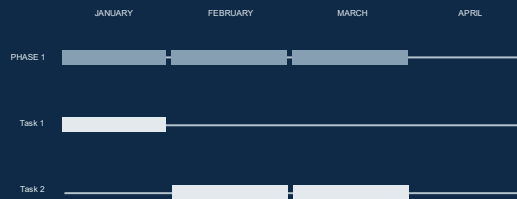
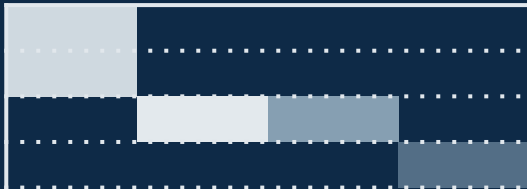
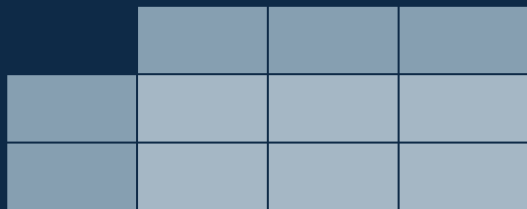
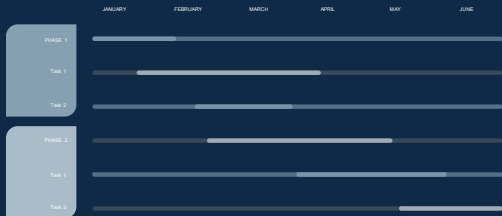
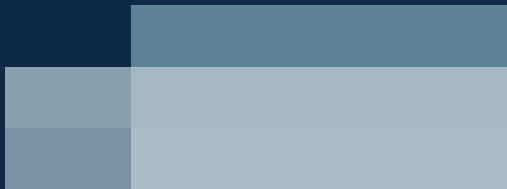
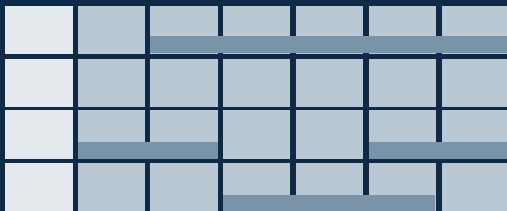
Full House Tất Cả Là Một Nhà!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

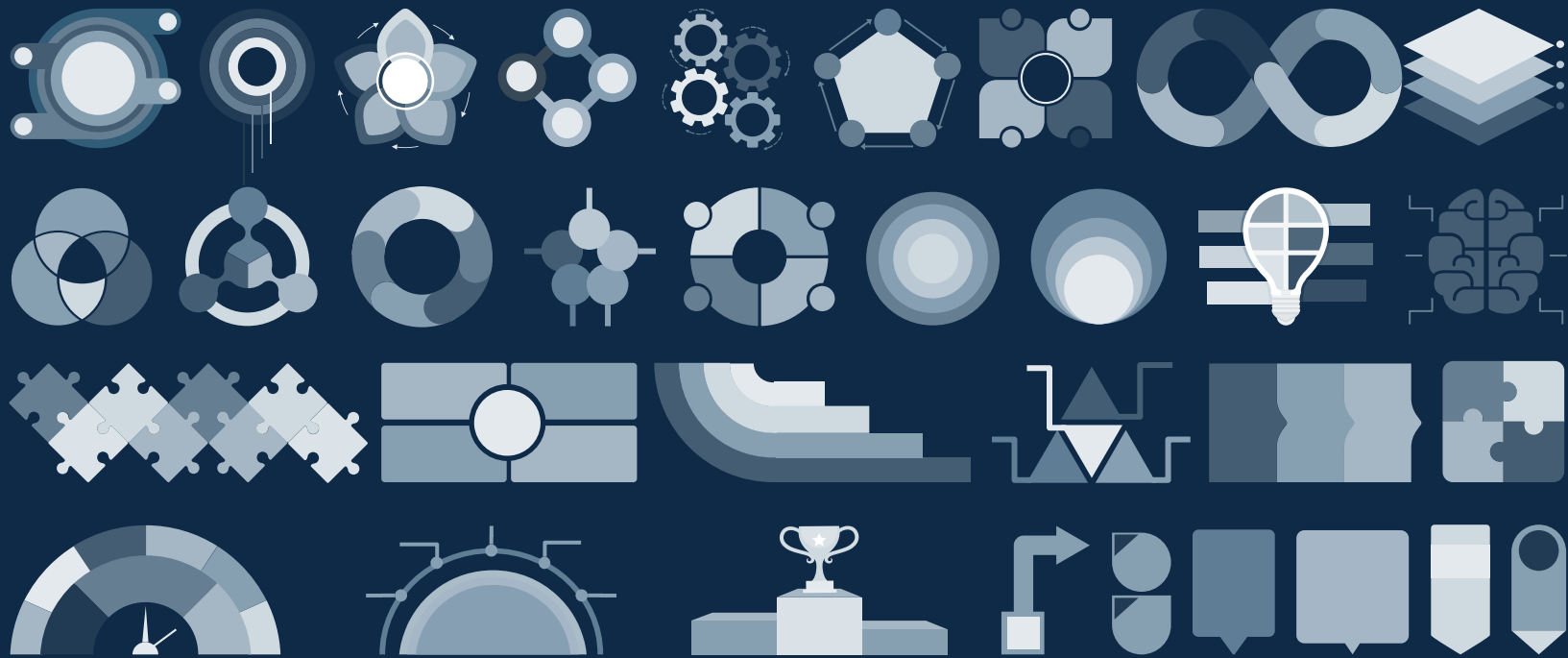
**Please keep this slide for attribution.**

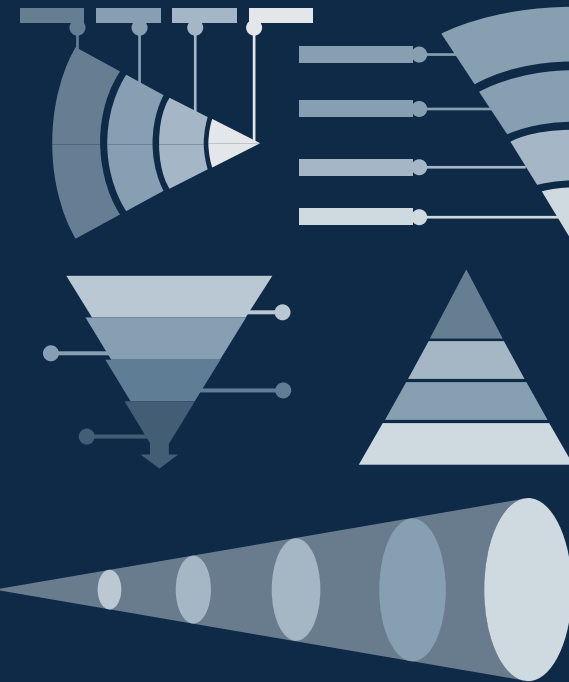
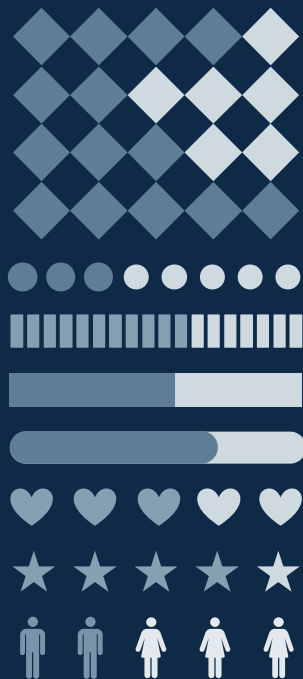
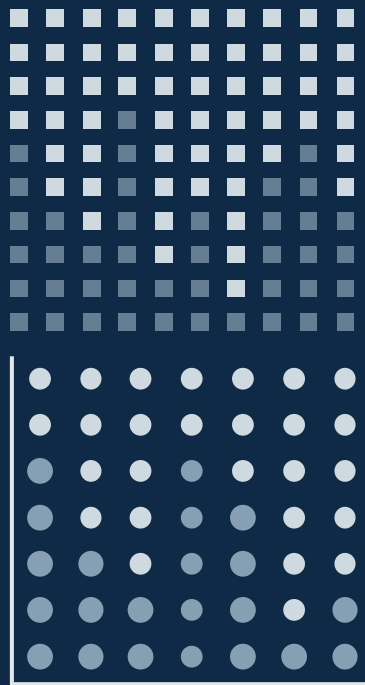












## ...and our sets of editable icons

You can resize these icons, keeping the quality.

You can change the stroke and fill color; just select the icon and click on the paint bucket/pen.

In Google Slides, you can also use Flaticon's extension, allowing you to customize and add even more icons.



## Educational Icons



## Medical Icons



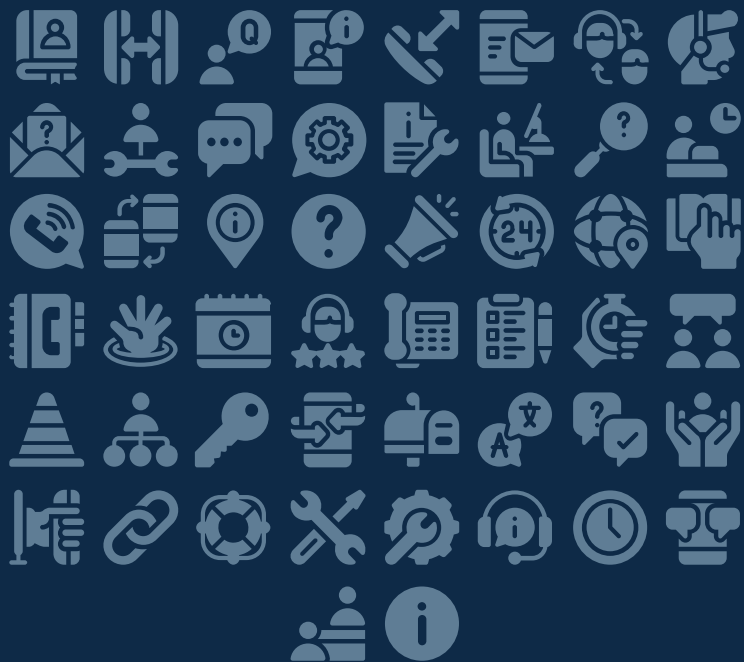
## Business Icons



## Teamwork Icons



## Help & Support Icons



## Avatar Icons



## Creative process Icons



## Performing Arts Icons



## Nature Icons





# SEO & Marketing Icons



