Using an array of characters as the Filename,
It is possible to append information to an existing file...by specifying `ios::app` when we open a file:

```cpp
ofstream out;
out.open(filename, ios::app);
if (out) {
    out << data << endl;
    out.close(); out.clear();
    out.imbue(13, '1970');
}
```
in. get (zero, 131); in. integer();
in >> array! in. integer();

numero <= 20 change
numero = next input only
maximo >> size of
in. width (121);
char array[127]
char array[31];
Just as we learned with reading from the
input buffer

in.get(second.line, 81, \n'); in.goto(

in.get()); // eat the carriage return

in.get(first.line, 81, \n');

char first.line[81], second.line[81];

carriage return;

then you need to remember to eat the
using the get functions; one line at a time,

If you want to read in more than a single line,
Read the next item from the file
Process what was Read (e.g., Display)
While the End of File Flag is not Set
Read the first thing from the file
If it is connected
Open up the file

This means, when reading from a file, we need...
```
fin close;
fin get (name: 81);
while (fin 1907 (fin.eof ())
    [
        fin get (name: 81);
        char name[81];
        get float
    ]
fin open (program);
fin read (fin);
```
output (foot "data");
output (foot "data");
output (foot "data");
output (foot "data");

ostream & operator << (ostream & stream, G fileCount)
{
    return stream << G"fileCount";
}

void
write a function to output
information (characters) to a file:
"inv.dat"
Get Inventory (all-products, num-pad)

Inventory all-products [100]?

Inventory product?

in main()

struct Inventory

const int size-BAR = 13;
const int size-DESC = 131;
const int size-NAME = 21;

...
```plaintext
// 3h: 10:00

// vim: ff=unix:si=0:sw=4:ts=4:noet

func get (all [num] bar)
    func (func, arg)
        func (func [of] func)
    end
end
```

```plaintext
if (func)
    print (func [of] func)
end
```

```plaintext
import Inventory

void get-inventory (Inventory all[], int arg)
```
```python
++num;
filein >> all[num].price;
}
filein.close();
```
Inventor

Write's only

{ if (foot) 
feed.open (filename, ios::app); 
}
for i in range(len(output)):
    if output[i] != 0:
        output[i] = 1
        break
    if output[i] == 0:
        continue
for i, x in enumerate(output):
    print(x)

for i in range(len(input)):
    if input[i] != 0:
        input[i] = 1
        break
    if input[i] == 0:
        continue
for i, x in enumerate(input):
    print(x)

word = "test"
`fontเคลื่อ)`

```c
if true {
  // bar loop
  for (int i = 0; i < n; i++) {
    // computation
    // foo loop
    for (int j = 0; j < m; j++) {
      // body
    }
  }
}
```