Status	Finished
Started	Friday, 1 November 2024, 9:07 AM
Completed	Monday, 4 November 2024, 9:52 AM
Duration	3 days
Grade	**10.00** out of 10.00 (**100**%)

Question 1

Correct

Mark 10.00 out of 10.00

[FactorialFunction]

Trong quá trình giải quyết các bài toán lập trình, có những phần việc được thực hiện lặp đi lặp lại nhiều lần khiến ta phải viết lại các đoạn code giống nhau nhiều lần. Việc lặp đi lặp lại các đoạn code giống nhau không chỉ gây phiền toái cho lập trình viên, mà còn khiến cho chương trình lủng củng, khó hiểu, khó sửa chữa và nâng cấp.

Hàm là một chuỗi các lệnh dùng để thực hiện một công việc nào đó. Sử dụng hàm sẽ giúp chúng ta tránh được việc viết nhiều lần một đoạn code thực hiện chung một công việc bởi sau khi định nghĩa hàm, mỗi khi cần thực hiện phần việc đó, ta chỉ cần gọi tên hàm để nó thực thi.

Hơn nữa, với những bài toán lớn, ta có thể chia nhỏ thành các bài toán con và viết các hàm để giải quyết từng bài toán con rồi ghép lại thành chương trình hoàn thiện. Phương pháp chia để trị này không chỉ giúp code của chúng ta nhìn gọn gàng và dễ hiểu hơn, mà còn giúp lập trình viên kiểm soát tốt hơn quá trình chạy của chương trình, rất có lợi cho quá trình *debug*.

Hàm có thể nhận vào 0 hoặc nhiều tham số. Một hàm có thể trả về (*return*) một kết quả nào đó, hoặc không trả về gì cả (*void*).

Trong C++, cú pháp để định nghĩa hàm như sau:

```
return_type tên_hàm(arg_type_1 arg_1, arg_type_2 arg_2, ...) {
    // đoạn code thực thi công việc của hàm
    // [nếu <strong>return_type</strong> khác <strong>void</strong>]
    return một giá trị/biến thuộc kiểu <strong>return_type</strong>;
}
```

Trong đó:

- `return_type`: kiểu dữ liệu của kết quả mà hàm trả về;
- `arg_i`: tên tham số (đầu vào) thứ i của hàm;
- `arg_type_i`: kiểu dữ liệu của `arg_i`.

Ví dụ, ta định nghĩa hàm sau đây để tính giá trị lớn nhất trong ba số nguyên:

```
int max_of_three(int a, int b, int c) {
    int max = a;
    if (max < b) max = b;
    if (max < c) max = c;
    return max;
}
```

Bài tập

Viết hàm `long factorial(int n)` trả về kết quả là giai thừa $n!$. Công thức tính giai thừa:

$$n! = 1 \times 2 \times 3 \times \ldots \times (n-1) \times n$$

Đầu vào

Đầu vào từ bàn phím gồm duy nhất một số nguyên n.

Đầu ra

In ra màn hình kết quả của hàm `factorial`🥐 .

Lưu ý

Các phần nhập/xuất dữ liệu trong hàm `main` đã được viết sẵn cho bạn. Bạn chỉ cần viết định nghĩa hàm `factorial` tại vị trí được yêu cầu trong ô trả lời.

For example:

Input	Result
4	24

Answer:

Reset answer

```cpp
#include <bits/stdc++.h>
using namespace std;
#define ll long long
#define ull unsigned long long
#define el "\n"
const int mod=1e9+7;
ll factorial(ll n)
{
    ll ans = 1;
    if( n == 0 || n == 1) return 1;
    for(ll i = 1; i <= n; i++) ans *= i;
    return ans;
}
/*int main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    ll n; cin >> n;
    cout << factorial(n);
    return 0;
}*/
void Print()
{
    ll n; cin >> n;
    cout << factorial(n);
}
```

	Input	Expected	Got	
✓	4	24	24	✓
✓	10	3628800	3628800	✓
✓	14	87178291200	87178291200	✓
✓	0	1	1	✓
✓	20	2432902008176640000	2432902008176640000	✓

Passed all tests! ✓

▸ SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

Correct

Marks for this submission: 10.00/10.00.

Back to Course