

VÕ TIẾN

Thảo luận kiến thức CNTT trường BK về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



Kỹ Thuật Lập Trình (Cơ bản và nâng cao C++)

KTILT1 - HK242

TASK 3 FUNCTION

Thảo luận kiến thức CNTT trường BK
về KHMT(CScience), KTMT(CEngineering)
<https://www.facebook.com/groups/khmt.ktmt.cse.bku>



1 Trắc Nghiệm

- Hàm trong C++ có tác dụng gì?
 - Giúp chương trình chạy nhanh hơn
 - Giúp chia chương trình thành các phần nhỏ, dễ quản lý
 - Bắt buộc phải có trong mọi chương trình C++
 - Không có tác dụng gì đặc biệt
- Một hàm trong C++ có thể có bao nhiêu tham số?
 - Chỉ 1 tham số
 - Tối đa 5 tham số
 - Tùy ý, không giới hạn số lượng tham số
 - Không thể có tham số
- Trong C++, kiểu trả về của hàm có thể là gì?
 - Chỉ có kiểu int
 - Chỉ có kiểu void
 - Bất kỳ kiểu dữ liệu hợp lệ nào
 - Không cần kiểu trả về
- Từ khóa nào dùng để kết thúc một hàm và trả về giá trị?
 - break
 - return
 - end
 - stop
- Hàm nào dưới đây là một ví dụ hợp lệ trong C++?
 - `int sum(a, b) { return a + b; }`
 - `void display() { cout << "Hello"; }`
 - `func() { return 0; }`
 - `string name {}`
- Hàm không có kiểu trả về sử dụng từ khóa nào?
 - return
 - void
 - null
 - empty
- Đâu là một khai báo hàm hợp lệ trong C++?
 - `int sum(int, int);`
 - `int sum(int a, int b) { return a + b; }`
 - `sum(int a, int b);`
 - `void(int);`
- Hàm đệ quy là gì?
 - Hàm có nhiều tham số
 - Hàm gọi lại chính nó
 - Hàm chỉ dùng trong các bài toán số học
 - Hàm không có giá trị trả về
- Hàm có thể được nạp chồng (overload) khi nào?
 - Khi có cùng tên nhưng khác danh sách tham số
 - Khi có cùng tên và cùng danh sách tham số
 - Khi có kiểu trả về khác nhau nhưng cùng tên
 - Khi không có tham số
- Hàm `main()` trong C++ có vai trò gì?
 - Là nơi bắt đầu thực thi chương trình
 - Không quan trọng, có thể bỏ qua
 - Được gọi tự động khi chương trình kết thúc
 - Chỉ dùng để khai báo biến
- Truyền tham trị (pass by value) có nghĩa là gì?
 - Truyền địa chỉ của biến vào hàm
 - Truyền bản sao của giá trị vào hàm
 - Truyền con trỏ của biến vào hàm
 - Truyền biến toàn cục vào hàm
- Khi nào nên sử dụng truyền tham chiếu (pass by reference)?
 - Khi muốn thay đổi giá trị của tham số gốc
 - Khi muốn bảo vệ giá trị của biến
 - Khi làm việc với số nguyên nhỏ
 - Khi gọi hàm đệ quy
- Từ khóa nào dùng để truyền tham chiếu nhưng không cho phép thay đổi giá trị?



- a) pointer b) static c) const d) reference
14. Đây là một cách truyền mảng vào hàm hợp lệ?
- a) `void func(int arr[]);` b) `void func(int arr);`
c) `void func(int &arr);` d) `void func(int ^arr);`
15. Một hàm có thể trả về gì?
- a) Chỉ số nguyên b) Chỉ một kiểu dữ liệu cố định
c) Bất kỳ kiểu dữ liệu nào hợp lệ d) Không thể trả về con trỏ
16. Hàm nào có thể được gọi mà không cần đối số?
- a) Hàm có tham số mặc định b) Hàm nạp chồng
c) Hàm đệ quy d) Hàm `main()`
17. Đây là một cách hợp lệ để gọi hàm trong C++?
- a) `sum(4, 5);` b) `int sum(4, 5);` c) `call sum(4, 5);` d) `func sum(4, 5);`
18. Hàm void có thể sử dụng câu lệnh `return` không?
- a) Có, nhưng không được trả về giá trị nào b) Không, hàm void không thể dùng `return`
c) Có, nhưng chỉ với con trỏ d) Không, `return` chỉ dành cho các kiểu dữ liệu khác void
19. Điều gì xảy ra nếu một hàm không có câu lệnh `return` nhưng có kiểu trả về không phải void?
- a) Chương trình sẽ báo lỗi biên dịch b) Hàm sẽ trả về giá trị ngẫu nhiên
c) Chương trình sẽ tiếp tục chạy bình thường d) Hàm sẽ tự động trả về 0
20. Hàm nào dưới đây là một hàm hợp lệ với tham số mặc định?
- a) `void greet(string name = "Guest");` b) `int sum(int a = 5, int b);`
c) `double square(double);` d) `int multiply(a = 2, b = 3);`
21. Khi truyền tham trị (pass by value), điều gì xảy ra với tham số?
- a) Giá trị gốc bị thay đổi b) Một bản sao của tham số được tạo ra
c) Hàm có thể thay đổi trực tiếp giá trị gốc d) Tham số được truyền bằng địa chỉ
22. Khi truyền tham chiếu (pass by reference), điều gì xảy ra với tham số?
- a) Một bản sao của tham số được tạo ra b) Hàm có thể thay đổi trực tiếp biến gốc
c) Bộ nhớ sẽ được cấp phát động d) Hàm sẽ nhận một bản sao của địa chỉ biến
23. Câu lệnh nào đúng khi truyền tham chiếu vào hàm?
- a) `void func(int x) { x = 10; }` b) `void func(int x) { x = 10; }`
c) `void func(int *x) { *x = 10; }` d) `void func(int x) { x = 10; }`
24. Khi truyền một mảng vào hàm, bản chất của việc truyền này là gì?
- a) Truyền tham trị của mảng b) Truyền tham chiếu của mảng
c) Truyền địa chỉ của phần tử đầu tiên d) Truyền con trỏ hằng đến mảng
25. Hàm nào dưới đây sử dụng tham chiếu hằng (const reference)?
- a) `void show(const int x) { cout << x; }` b) `void show(int x) { cout << x; }`
c) `void show(int *x) { cout << *x; }` d) `void show(int x[]) { cout << x[0]; }`



2 Đọc Code

Câu 1. Kết quả của trường trình

Kết quả và giải thích: ...

- ...
- ...

```
1  #include <iostream>
2  using namespace std;
3
4  void modify(int &x, int y) {
5      x *= 2;
6      y += 5;
7  }
8
9  int main() {
10     int a = 3, b = 4;
11     modify(a, b);
12     cout << a << " " << b << endl;
13     return 0;
14 }
```

Câu 2. Kết quả của chương trình

Kết quả và giải thích: ...

- ...
- ...

```
1  #include <iostream>
2  using namespace std;
3
4  void modify(int arr[]) {
5      arr[0] += 10;
6      int x = arr[1];
7      x *= 2;
8  }
9
10 int main() {
11     int arr[] = {1, 2, 3, 4, 5};
12     modify(arr);
13     cout << arr[0] << " " << arr[1] << endl;
14     return 0;
15 }
```

Câu 3. Kết quả của chương trình



Kết quả và giải thích: ...

- ...
- ...

```
1 void processString(char s[]) {
2     int len = 0;
3     while (s[len] != '\0') len++;
4     for (int i = 0; i < len / 2; i++) {
5         char temp = s[i];
6         s[i] = s[len - i - 1];
7         s[len - i - 1] = temp;
8     }
9     if (s[0] >= 'a' && s[0] <= 'z') {
10        s[0] -= 32;
11    }
12 }
13
14 char input[] = "hello world 123";
15 processString(input);
```

Câu 4. Kết quả của chương trình

Kết quả và giải thích: ...

- ...
- ...

```
1 bool isPrime(int n) {
2     if (n < 2) return false;
3     for (int i = 2; i * i <= n; i++) {
4         if (n % i == 0) return false;
5     }
6     return true;
7 }
8
9 void processArray(int arr[], int size) {
10    int temp = arr[0];
11    arr[0] = arr[size - 1];
12    arr[size - 1] = temp;
13
14    int primeSum = 0;
15    for (int i = 0; i < size; i++) {
16        if (isPrime(arr[i])) {
17            primeSum += arr[i];
18        }
19    }
20
21    if (primeSum % 2 == 0) {
22        primeSum /= 2;
23    } else {
24        primeSum *= 2;
25    }
26
27    cout << "Total: " << primeSum << endl;
28 }
29
30 int arr[] = {3, 8, 5, 10, 7, 2};
31 processArray(arr, 6);
```

Câu 5. Kết quả của chương trình



Kết quả và giải thích: ...

- ...
- ...

```
1 #include <iostream>
2 using namespace std;
3
4 int recursiveFunction(int n) {
5     if (n <= 0) return 1;
6     if (n % 2 == 0)
7         return 2 * recursiveFunction(n - 2);
8     return recursiveFunction(n - 1) + n;
9 }
10
11 int main() {
12     cout << recursiveFunction(7) << endl;
13     return 0;
14 }
```

Câu 6. Kết quả của chương trình

Kết quả và giải thích: ...

- ...
- ...

```
1 #include <iostream>
2 using namespace std;
3
4 int foo(int arr[], int n) {
5     if (n == 0) return 0;
6     return arr[n - 1] + foo(arr, n - 1);
7 }
8
9 int main() {
10     int arr[] = {3, 1, 4, 1, 5};
11     int n = sizeof(arr) / sizeof(arr[0]);
12     cout << foo(arr, n) << endl;
13     return 0;
14 }
```

Câu 7. Kết quả của chương trình

Kết quả và giải thích: ...

- ...
- ...

```
1 void foo(char str[], int left, int right) {
2     if (left >= right) return;
3     swap(str[left], str[right]);
4     foo(str, left + 1, right - 1);
5 }
6
7 int main() {
8     char str[] = "recursion";
9     int n = sizeof(str) / sizeof(str[0]) - 1;
10    foo(str, 0, n - 1);
11    cout << str << endl;
12    return 0;
13 }
```



3 Bài Tập

Câu 1: Tìm phần tử lớn thứ hai trong mảng

Đề bài: Viết hàm tìm phần tử lớn thứ hai trong mảng số nguyên không sử dụng sort

```
1 int findSecondLargest(int arr[], int n) {  
2     // TODO  
3 }
```

Test case

Input	Output
10 20 4 45 99	45
5 3 1 2 4	4
8 8 8 8 8	-1
100 90 80 70 60	90

Câu 2: Đếm số lần xuất hiện của phần tử trong mảng

Đề bài: Viết hàm đếm số lần xuất hiện của một phần tử trong mảng số nguyên không sử dụng thư viện 'algorithm'.

```
1 int countOccurrences(int arr[], int n, int x) {  
2     // TODO  
3 }
```

Test case

Input (Mảng)	Phần tử cần đếm	Output
1 2 3 2 1 2 4 2	2	4
5 5 5 5 5 5	5	6
10 20 30 40 50	25	0
1 1 1 2 2 2 3 3 3	3	3

Câu 3: Xóa tất cả các phần tử trùng nhau trong mảng

Đề bài: Viết hàm xóa tất cả các phần tử trùng nhau trong mảng và chỉ giữ lại một lần duy nhất.

```
1 int removeDuplicates(int arr[], int n) {  
2     // TODO  
3 }
```

Test case

Input	Output
1 1 2 2 3 4 4 5 5	1 2 3 4 5
1 2 2 3 3 4 5 5 5 6	1 2 3 4 5 6
10 10 20 30 30 30 40	10 20 30 40
5 5 5 5 5 5 5 5	5