=================Ngày 17/07=========================

**Bài 1:** Viết mã nguồn C++ thực hiện công việc sau

a) Khai báo cấu trúc Market đại diện cho một chợ trên địa bàn. Market có các trường thông tin như: id (định danh), name (tên), income (doanh thu theo ngày cao nhất), money (số tiền trong tài khoản hiện nay), int customers[5]: danh sách tối đa id của năm Market đặt mua sản phẩm của Market hiện tại. Viết chương trình yêu cầu người dùng nhập vào số lượng N của các Market và nhập vào các trường thông tin của các Market . Chương trình sẽ in ra 1 nếu tổng số money này là số nguyên chia hết cho 1000 và in ra 0 nếu ngược lại. (**2 điểm**)

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

**Output:**

1

Giải thích: Dòng 1 là số lượng shop (N = 5)

Dòng 2 là thông tin của Market với id = 1, tên nó là john, doanh thu income của nó là 300, money của nó là 8000, có 2 Market khác thường xuyên mua sản phẩm của nó là Market với id bằng 2 và bằng 5. Hiện tại ngoài hai Market đó ra thì chưa có Market nào thường xuyên đặt mua sản phẩm của Market john.

<https://onlinegdb.com/5-_Mzi6dp>

b) (**2 điểm**) cho phép người dùng nhập vào id X của một Market và chương trình sẽ phải đếm xem có bao nhiêu Market cung cấp sản phẩm cho shop X đó. Chương trình sẽ phải in ra “số nguyên bé nhất còn lớn hơn trung bình cộng doanh thu income của các Market cung cấp đó”. Trong trường hợp không có Market nào cung cấp sản phẩm cho Market có id đó thì in ra số 0.

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

1

**Output:**

851

Giải thích: ta thấy có hai Market cung cấp sản phẩm cho Market 1 là Market 2 và 5. Trung bình cộng income của chúng là (600 + 1100)/2 = 850 và số nguyên bé nhất còn lớn hơn 850 là 851.

<https://onlinegdb.com/Cpz1Miscq>

c) (**3 điểm**) Chương trình phải đếm xem trong danh sách các Market, có bao nhiêu Market có số khách hàng (là Market khác) đều là Market có income lớn hơn Market cung cấp. Chương trình sẽ phải in ra số lượng các Market đó

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

**Output:**

2

Giải thích: Market 1 (income = 300) được mua hàng bởi Market 2 (income = 600) và Market 5 (income = 1100). Còn Market 4 (income = 500) được mua bởi Market 5 (income = 1100) và Market 3 (income = 800). Vậy tổng có 2 Market.

<https://onlinegdb.com/fiHu96fLw>

d) (**2 điểm**) Chương trình cho phép người dùng nhập vào một id của Market X và phải kiểm tra xem liệu Market đó nên mời chào khách hàng tiềm năng nào. Để có thể đi tìm được khách hàng tiềm năng thì:

Market X đó vẫn chưa lấp đầy danh sách customer của mình

khách hàng tiềm năng có doanh thu lớn hơn doanh thu của các khách còn lại. Nếu không có khách hàng nào như thế thì khách đó phải có doanh thu cao nhất trong số các Market chưa nằm trong danh sách customer.

Chương trình sẽ phải in ra id của khách hàng tiềm năng đó, nếu không có khách nào thì in ra -1

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

1

**Output:**

3

Giải thích: do Market 1 có khách là Market 2 (income = 600) và 5 (income = 1100) nên khách hàng tiềm năng của nó sẽ là Market 3 (income = 800) vốn có income lớn nhất trong số các shop còn lại.

<https://onlinegdb.com/oi_Foe43Ws>

e) (**1 điểm**) Chương trình cho phép các Market đổi tên, nhưng một Market khi đổi tên thì không được phép vi phạm các luật sau đây:

tên Market mới trùng với một trong số các Market cũ

tên Market mới nếu đảo ngược lại các ký tự thì không được phép trùng tên với một trong số các Market cũ

tên Market mới không được chứa tên một trong số các Market còn lại bên trong tên của nó

Chương trình sẽ phải in ra 1 nếu Market cũ đổi được tên, in ra 0 nếu không được

Ví dụ:

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

1 retep

**Output:**

-1

Ở đây input "1 retep" nghĩa là Market 1 muốn đổi tên thành "retep", nhưng điều này vi phạm vì "retep" viết ngược lại là "peter"

Còn ở đây:

**Input:**

5

1 john 300 8000 2 5 -1 -1 -1

2 peter 600 4000 3 1 -1 -1 -1

3 tom 800 9000 4 2 -1 -1 -1

4 bob 500 3000 5 3 -1 -1 -1

5 jack 1100 7000 1 4 -1 -1 -1

1 tommy

**Output:**

-1

Cái tên mới "tommy" cũng không được vì Market 3 đã có mang tên "tom".

=================Ngày 10/07=========================

Bài 1:  Viết chương trình C++ thực hiện nhiệm vụ sau:

a) Cho phép người dùng nhập vào giá trị N đại diện cho số ngôi nhà trong khu vực quận. Chương trình in ra 1 nếu số nhà > 100, ngược lại in ra 0

#include <iostream>

int main( ){

int N;

cin>>N;

if(N > 100) cout<<1; else cout<<0;

return 0;

}

b) Khai báo cấu trúc Home có các trường thông tin: double square (diện tích), double cost (giá tiền), int yr (năm chuyển nhượng gần nhất). Hãy cho phép người dùng nhập vào giá trị cho mảng có N phần tử. Sau đó tính trung bình diện tích của các ngôi nhà. Chương trình sẽ in ra 1 nếu trung bình này lớn hơn 100. Ngược lại in ra 0.

#include <iostream>

typedef struct Home{

double square, cost;

int yr;

}Home;

int main( ){

int N;

cin>>N;

Home arr[N];

double sum = 0;

for(int i = 0; i < N; i++){

cin>>arr[i].square;

cin>>arr[i].cost;

cin>>arr[i].yr;

sum += arr[i].square;

}

sum /= N;

if(sum > 100) cout<<1; else cout<<0;

return 0;

}

c) Chương trình hãy tìm xem trong số các ngôi nhà có năm chuyển nhượng gần nhất là lớn nhất, thì ngôi nhà nào có diện tích lớn nhất. Chương trình sẽ in ra 1 nếu năm chuyển nhượng lớn hơn diện tích, ngược lại in ra 0.

d) Chương trình hãy tìm xem trong các ngôi nhà có giá thành thấp nhất thì ngôi nhà nào có lần chuyển nhượng gần đây nhất lại cách xa nhất. Chương trình sẽ in ra số lượng các ngôi nhà cùng đạt được yêu cầu đó.

e) Chương trình sẽ đánh giá xem trong N ngôi nhà đó, ngôi nhà nào là đáng đầu tư nhất. Biết rằng nó phải là thuộc nhóm có cost\*yr/square đạt giá trị cao nhất. Hãy in ra số lượng ngôi nhà đáng đầu tư nhất đó.

=================Ngày 24/06===========================

Đã đến slide trang 17 của chương 9

Assert này là không nên viết

assert(x++ < 4); //Không nên vì giá trị x sẽ bị thay đổi.

printf(“%d %d", \_\_FILE\_\_, \_\_LINE\_\_); //chạy đến dòng này, chương trình sẽ in ra file hiện tại

//là file nào, dòng hiện tại là dòng mấy

Khi dùng gdb, ta phải thêm tham số -g trước tên file mã nguồn để hỗ trợ debug

=================Ngày 19/06===========================

=================Ngày 12/06===========================

RC7D222: em hãy điền vào chỗ trống để mã nguồn sau hoạt động được, biết rằng lớp Student có thuộc tính age kiểu int và lớp đó có sẵn hàm khởi dựng nhận hai tham số string và int.

….A….  objK …B…Student(“Kien", 19);

….C….  objH …D…Student(“Hoang", 19);

objK.age = 20;

objH->age = 20;

A. Ta điền như sau: A điền Student; B điền = new, C điền Student\*\*; D điền = new Student

B. Ta điền như sau: A điền Student\*; B điền =, C điền Student\*; D điền = new Student

C. Ta điền như sau: A điền Student; B điền =, C điền Student\*; D điền = new

D. Tất cả các đáp án trên đều sai

ANC: C

Giải thích:

sau đây là link gỉai thích 3 cách tạo đối tượng trong C++ <https://onlinegdb.com/DbchSCaOW>

Ở Set, tính chất chống phản xạ được hiểu như sau: một Set không thể chứa hai phần tử giống hệt nhau. Set a = <x, x> là không được phép tồn tại.

Phản đối xứng: Set a = <x, y> và Set b = <y, x> thì hai Set là khác nhau (không phải là một).

Bắc cầu. Set a giống hệt Set b và Set b giống hệt Set c thì Set a và c giống hệt nhau.

Do các phần tử của deque được thêm vào trước vào sau tại bất kỳ lúc nào nên gần như không thể đảm bảo các phần tử đó được nằm kế tiếp nhau trong bộ nhớ.

RC7D222: Theo em nguyên nhân vì sao deque lại không thể đảm bảo các phần tử nằm kế tiếp nhau trong bộ nhớ.

A. Do các phần tử của deque được thêm vào trước vào sau tại bất kỳ lúc nào

B. Do khi deque sắp đầy thì hệ điều hành sẽ tự động cấp phát gấp đôi bộ nhớ cho deque

C. Do deque chỉ được thêm các phần tử mới vào phía sau của danh sách đó

D. Tất cả các đáp án còn lại đều sai

ANC: A

=================Ngày 05/06===========================

Đã dừng lại ở slide trang 23 của chương 7

C7M222: Em hãy giải thích lý do tại sao queue được phân loại là Container Adapter mà map lại được xếp là Ordered Associative Containers:

A.

B.

C.

D.

ANC:

C7D222: Em hãy chọn lựa phát biểu đúng về phân loại các cấu trúc dữ liệu sau: vector, queue, map

A. Vector là loại sequence container, queue là Ordered container, map là Container Adapter.

D. Vector là loại sequence container, map là Ordered container, queue là Container Adapter.

…

ANC: D

C7D222: Theo em nguyên nhân nào giải thích được việc độ phức tạp của thao tác trong trường hợp tốt nhất của danh sách liên kết kép khi chương trình muốn chèn/xoá phần tử cuối.

A. Độ phức tạp đó là 𝞗(1) vì ta đã có con trỏ vốn trỏ đến phần tử đầu tiên của danh sách

B. Độ phức tạp đó là 𝞗(1) vì ta đã có con trỏ vốn trỏ đến phần tử cuối của danh sách.

C. Độ phức tạp đó là 𝞗(n) vì ta đã có con trỏ vốn trỏ đến phần tử đầu tiên của danh sách

D. Độ phức tạp đó là 𝞗(n) vì ta đã có con trỏ vốn trỏ đến phần tử cuối của danh sách

ANC: B

=================Ngày 29/05===========================

AC6D222: Theo em phát biểu nào đúng về dòng 4 và 5

|  |  |
| --- | --- |
| **1:**  **2:**  **3:**  **4:**  **5:**  **6:** | **struct** state{      int n, k;      char A, B, C;      state(int \_n, char \_A, char \_B, char \_C, int \_k):          n(\_n), A(\_A), B(\_B), C(\_C), k(\_k) {}  }; |

A. Dòng 4 và 5 là các dòng viết sai cú pháp của C: không thể khai báo hàm bên trong struct

B. Dòng 4 và 5 là khai báo các trường thông tin của struct state

C. Dòng 4 và 5 là các lời khai báo hàm huỷ của struct state

D. Dòng 4 và 5 là các lời khai báo hàm khởi dựng của struct state

ANC: D

Giải thích: C++ chấp nhận quan điểm rằng coi struct cũng là một class.

AC6D222: Với đoạn mã dưới đây, biết rằng nó có thể được dùng kĩ thuật khử đệ quy đuôi. Em hãy xác định các biểu thức A(x), B(x), D(x) và f(x). Trong đó nếu ta viết n\* nghĩa là thực hiện n nhân với một số nào đó.

int fact(int n){

if(n == 0) return 1;

return n\*fact(n-1);

}

A. A(n) = n\*; B(n) = n != 0; D(n) = 1; f(n) = n;

B. A(n) = 1; B(n) = n == 0; D(n) = 1; f(n) = n-1;

C. A(n) = n; B(n) = n != 0; D(n) = 1; f(n) = n;

D. A(n) = n\*; B(n) = n == 0; D(n) = 1; f(n) = n-1;

AND: D

AC6D222: Với định nghĩa công thức truy hồi là công thức tìm giá trị của f(n) từ f(n-1) hoặc f(m) với m là giá trị đơn giản hơn so với n. Cho hai đoạn mã giả dưới đây, theo em đâu là phát biểu đúng

Giải thuật đệ quy tính f(n)

f(n) ≡ **if**(n == no) **return** C;

**else** return (g(n,f(n -1));

Giải thuật lặp tính f(n)

k = no; F:= C;

F = f(no);

**while**(k < n){

    k++;

    F = g(k,F);

}

**return** F;

A. Biến F là một con trỏ hàm.

B. Hàm g là hàm cài đặt công thức truy hồi

C. Hàm f là hàm cài đặt công thức truy hồi cho việc tìm giá trị cuối cùng.

D. Hàm g là hàm tìm ra giá trị ban đầu của F.

ANC: B

Công thức truy hồi là công thức tìm giá trị của f(n) từ f(n-1) hoặc f(m) với m là giá trị đơn giản hơn so với n

AC6K222: theo em ở hai đoạn mã dưới đây, cách nào sử dụng for sẽ giúp tăng tốc độ của chương trình (gợi ý for có thể khiến CPU thực hiện kĩ thuật pipeline)

mã một: hàm fact ở slide trang 52 chương 6

Mã hai:

int assign( ){

int i = 0;

while(i < n){

A[i] = B[i] + i;

i++;

}

}

A. Cả hai đoạn mã đều không được cải thiện tốc độ nếu dùng for.

B. Cả hai đoạn mã đều được cải thiện tốc độ nếu dùng for.

C. Đoạn mã một không được cải thiện tốc độ nếu dùng for, đoạn mã hai thì có thể

D. Đoạn mã hai không được cải thiện tốc độ nếu dùng for, đoạn mã một thì có thể.

ANC: C

=================Ngày 22/05===========================

Đã dừng lại ở slide trang 40 của chương 6

AC6D222:

|  |  |
| --- | --- |
| 1:  2:  3:  4:  5:  6:  7:  8:  9:  10:  11:  12:  13: | double mem[1000];  double fibo(double n) {  **if** (n <= 2) **return** 1.0;  **if** (mem[(int)n] != -1) **return** mem[(int)n];      int res = fibo(n - 2) + fibo(n - 1);      mem[(int)n] = res;  **return** res;  }  int main(){      memset(mem, - 1, **sizeof**(mem));      cout << fibo(10);  **return** 0;  } |

Theo em câu lệnh ở dòng nào giúp việc tính toán hàm được thay bằng việc đọc giá trị đã lưu sẵn

A. Câu lệnh ở dòng 1

B. Câu lệnh ở dòng 10

C. Câu lệnh ở dòng 6

D. Câu lệnh ở dòng 4.

ANC:

AC6D222: theo em thì đâu là hàm đệ quy tuyến tính, giả sử rằng tham số x luôn là số nguyên dương

A. int f(int x){

if(x == 0) return 1;

return f(x - 1);

}

B. int f(int x){

if(x = 0) return 1;

return 1 + f((int)sqrt(x));

}

C. int f(int x){

if(x == 0) return 1;

return 1 + f(x + 1);

}

D. int f(int x){

if(x = 0) return 1;

return 1 + f(x/2);

}

E. Tất cả bốn hàm còn lại đều là đệ quy tuyến tính

F. Tất cả bốn hàm còn lại đều không phải là đệ quy tuyến tính

G. Đáp án khác

ANC: A

=================Ngày 08/05===========================

C5D222: Theo em đâu là một trong các tác dụng của lời khuyên “tránh dùng nhiều biến trung gian"?

A. Vì để tránh làm nóng CPU

B. Vì để tránh khối ALU phải tính toán ra được địa chỉ lưu trữ các biến trung gian đó

C. Vì để tránh ổ cứng phải hoạt động nhiều để lưu biến

D. Vì để tránh các biến đó làm chật bộ nhớ cache.

E. Tất cả các phát biểu trên đều đúng

ANC: D

C5D222: Theo em phát biểu nào đúng về ví dụ đặt tên như sau: int My\_Camel\_Case

A. Đây là phong cách Camel Case

B. Đây là phong cách Pascal Case

C. Đây là phong cách Snake Case

D. Đây là phong cách Screaming snake case

E. Tất cả các phát biểu đều sai.

ANC: E

C5D222: Theo em phát biểu nào đúng về ví dụ đặt tên theo phong cách Pascal Case

A. int myCamelCase;

B. int my\_camel\_case;

C. int My\_Camel\_Case;

D. int MY\_CAMEL\_CASE

E. Tất cả các phát biểu đều sai

ANC: E

C5D222: Theo em phát biểu nào đúng về ví dụ đặt tên theo phong cách Camel Case

A. int myCamelCase;

B. int my\_camel\_case;

C. int My\_Camel\_Case;

D. int MY\_CAMEL\_CASE

ANC: A

=================Ngày 24/04===========================

AC4D222: theo em nếu khai báo hàm bar là hàm inline thì ta tránh được bốn bước gì khi gọi hàm bar trong hàm foo như sau:

int foo( ){

bar( );

return 0;

}

A. (1) Cất ngữ cảnh (context) của hàm foo. (2) Tạo ngữ cảnh của hàm bar. (3) Cất ngữ cảnh của hàm bar; (4) Khôi phục ngữ cảnh của hàm foo

B. (1) Cất ngữ cảnh (context) của hàm bar. (2) Tạo ngữ cảnh của hàm bar. (3) Cất ngữ cảnh của hàm foo; (4) Khôi phục ngữ cảnh của hàm foo.

C. (1) Tạo ngữ cảnh (context) của hàm foo. (2) Tạo ngữ cảnh của hàm bar. (3) Cất ngữ cảnh của hàm bar; (4) Khôi phục ngữ cảnh của hàm foo.

D. (1) Cất ngữ cảnh (context) của hàm bar. (2) Cất ngữ cảnh của hàm bar. (3) Tạo ngữ cảnh của hàm bar; (4) Khôi phục ngữ cảnh của hàm foo.

ANC: A

Giả sử ta có hàm như sau:

double getGPA(double math, double oop, double english, double project){

}

với các tham số trên, program (cụ thể hơn là OS) phải đi lùng bộ nhớ ở các nơi khác nhau để nạp vào giá trị của 4 biến trên. Tình huống sẽ mất công hơn nếu 4 biến nằm ở các vùng nhớ khác nhau trong RAM

Tốt nhất khai báo cấu trúc Student

typedef struct Student{

double math, oop, english, project;

} Student;

Như vậy double getGPA(Student student)

OS chỉ việc đọc các biến ở cạnh nhau (vì cùng thuộc biến cấu trúc Student)

Tình huống sẽ tốt hơn nữa nếu ta truyền con trỏ double getGPA(Student\* student)

OS sẽ chỉ nạp vào cache một số byte phù hợp với kích thước con trỏ

double sum = 0;

for(int i = 0; i < 100; i++)

{

sum += sin(i % 10);

}

double M[10] = { 0, sin(1), sin(2), sin(3), sin(4), …};

for(int i = 0; i < 100; i++)

{

sum += M[i % 10];

}

Nên đọc 4 chương đầu của Introduction to High performance computing for scientist and engineers của GS. Hager

tại sao lại phải viết vòng lặp for(j = 0 ; j < 100000; ở bên ngoài vòng for(int i = 0; i < 100; i++)???

Vì có thể vòng for bên trong có thời gian rất ngắn (dưới ms)

Thoi gian tinh sum += sin(i%10): 0.241486(s)

Thoi gian tinh sum += M[i%10]: 0.037204(s)

100\*(sum - sum1)/sum = 0

<https://onlinegdb.com/BZ_vb8kF0>

int c = 0, d = 0;

int a[1000] = {2};

for(int i = 0; i < 1000; i++){

c += a[i];

d -= a[999-i];

}

để đo được hiệu năng cache của đoạn code trên thì ta cần viết

for(int j = 0; j < 100000; j++){

     c = 0; d = 0;

     for(int i = 0; i < 1000; i++){

    c += a[i];

    d -= a[i];

     }

     if(j\*j - j + 1 < 0) a[j % 1000] = -2;

}

Số lần lặp j thực hiện 100000 thì dễ hiểu (đã nói ở trên). Nhưng tại sao lại có điều kiện

if(j\*j - j + 1 < 0) a[j] = -2;

Thoi gian tinh c, d: 0.276542(s)

Nếu ta không có điều kiện đó thì compiler sẽ hiểu rằng phần tử của mảng a sẽ không bao giờ được thay đổi giá trị. Nói cách khác, lặp j 100000 thì đáp án của c và d không đổi. Một số compiler hiện đại có thể phát hiện ra điều này (lặp bao nhiêu lần thì c, d không đổi) do đó chúng có thể chỉ lặp j vài lần, các lần khác không lặp

Để lừa compiler thì ta đặt điều kiện và câu lệnh sửa giá trị của mảng a; và compiler sẽ cảm thấy nó cần chạy hết tất cả vòng lặp.

Với 3 vòng lặp như sau:

int c = 0, d = 0, e = 0;

int a[1000] = {2};

int b[1000] = {1};

for(int i = 0; i < 1000; i++){

c += a[i];

}

for(int i = 0; i < 1000; i++){

d += b[i];

}

for(int i = 0; i < 1000; i++){

e += a[i];

}

THời gian chạy 100000 lần 3 vòng for kia là: Thoi gian tinh c, d, e: 0.820994(s)

Nếu ta lợi dụng cache:

for(int i = 0; i < 10; i++){

c += a[i];

e += a[i];

}

for(int i = 0; i < 10; i++){

d += b[i];

}

Cache sẽ được tối ưu hơn, vì CT chỉ việc 2 lần nạp dữ liệu từ RAM vào cache (nạp a, b) thay vì nạp 3 lần dữ liệu từ RAM vào cache (nạp a, b, a) như cách trên.

Khi chạy thử, ta thấy:

Thoi gian tinh c, d, e: 0.720052(s)

Thoi gian tinh c, d, e theo cach moi: 0.490911(s)

<https://onlinegdb.com/zfqQYrHCd>

int c = 0, d = 0;

int a[1000] = {2};

for(int i = 0; i < 1000; i++){

c += a[i];

d -= a[999-i];

}

Ta có thể áp dụng Lazy Evaluation bằng cách viết:

int c = 0, d = 0;

int a[1000] = {2};

for(int i = 0; i < 1000; i++){

c += a[i];

}

d = -c;

Khi đặt kích thước mảng là bội của 2 sẽ giúp compiler thực hiện 1 số thao tác tối ưu hoá SIMD hoặc vectorization.

Early binding

class A{

public:

  public void m( ) { }

};

class B : A{

public:

public void m( ) { }

};

trừ khi bắt buộc phải dùng dynamic binding (hoặc còn gọi là late binding) thì tốt nhất ta nên khai báo như sau: B\* obj = new B( );  obj->m( );

Còn nếu dùng dynamic binding thì:

đặt từ khoá virtual trước từ void trong phương thức m của A

 A\* obj = new B( );

obj->m( );

nếu dùng dynamic binding (hay còn gọi late binding) sẽ đảm bảo tính đa hình của mã C++, nhưng điều phải hi sinh là tốc độ

=================Ngày 17/04===========================

Ngày 17/04 đã đến slide trang 32 của chương 4.

C4K222: Giả sử giá trị N = 10000, người ta muốn sử dụng phương pháp nội suy tuyến tính để tìm hàm sigmoid(178.89) khi đã biết giá trị của sigmoid(0.21). X0 = 0.21 và X1 = 0.22.

Vậy theo em số lượng phép toán với số thực dấu phẩy động là bao nhiêu phép để ra được sigmoid(178.89). Giả sử phép toán y = a\*x + b được coi là 2 phép toán dấu phẩy động khác nhau.

A. Chỉ 3 phép toán.

B. Chỉ 6 phép toán nếu đã xây dựng thành công mảng

C. Chỉ 9 phép toán nếu chưa xây dựng thành công mảng

D. Khoảng hơn 600 phép toán dấu phẩy động nếu chưa xây dựng thành công mảng.

E. Tất cả phát biểu trên đều sai

ANC: A

C4K222: Giả sử giá trị N = 10000, người ta muốn sử dụng phương pháp nội suy tuyến tính để tìm hàm sigmoid(1.78) khi đã biết giá trị của sigmoid(0.21). X0 = 0.21 và X1 = 0.22.

Vậy theo em số lượng phép toán với số thực dấu phẩy động là bao nhiêu phép để ra được sigmoid(1.78). Giả sử phép toán y = a\*x + b được coi là 2 phép toán dấu phẩy động khác nhau.

A. Chỉ 3 phép toán

B. Chỉ 6 phép toán nếu đã xây dựng thành công mảng

C. Chỉ 9 phép toán nếu chưa xây dựng thành công mảng

D. Khoảng hơn 600 phép toán dấu phẩy động nếu chưa xây dựng thành công mảng.

E. Tất cả phát biểu trên đều sai

ANC: D

AC4D222: Theo em ta sẽ tiết kiệm được bao nhiêu chu kỳ nếu các biến a, b, x là các số thực và (giả sử rằng) việc tìm căn bậc hai sẽ tốn 4 chu kỳ (của CPU); tính nhân hai số thực tốn 2 chu kỳ, cộng hai số thực tốn 1 chu kỳ. Coi rằng thời gian cho việc chuyển đổi kiểu dữ liệu sẽ không đáng kể. Và a = 1, b = 2

A. Sẽ chẳng tiết kiệm được chu kỳ nào, hai cách đều giống nhau

B. Sẽ tiết kiệm được 4 chu kỳ

C. Sẽ tiết kiệm được 7 chu kỳ

D. Sẽ tiết kiệm được 9 chu kỳ

E. Cách dưới còn làm chậm chương trình đi hơn

F. Đáp án khác.

ANC: F

AC4D222: Theo em ta sẽ tiết kiệm được bao nhiêu chu kỳ nếu các biến a, b, x là các số thực và (giả sử rằng) việc tìm căn bậc hai sẽ tốn 4 chu kỳ (của CPU); tính nhân hai số thực tốn 2 chu kỳ, cộng hai số thực tốn 1 chu kỳ. Coi rằng thời gian cho việc chuyển đổi kiểu dữ liệu sẽ không đáng kể. Và a = 4, b = 2

A. Sẽ chẳng tiết kiệm được chu kỳ nào, hai cách đều giống nhau

B. Sẽ tiết kiệm được 4 chu kỳ.

C. Sẽ tiết kiệm được 7 chu kỳ

D. Sẽ tiết kiệm được 9 chu kỳ

E. Cách dưới còn làm chậm chương trình đi hơn

F. Đáp án khác

ANC: B

C4D222: Theo em phát biểu nào đúng về thứ tự thực hiện việc chèn mã của compiler

A. Các macro sẽ được compiler xử lý trước, rồi mới đến xử lý chèn mã của hàm inline

B. Các hàm inline sẽ được compiler xử lý trước, rồi mới đến xử lý chèn mã của các macro.

C. Các macro và các hàm inline sẽ được compiler xử lý tuần tự, xong hàm inline rồi sẽ đến macro, sau đó đến hàm inline.

D. Cả macro và hàm inline đều không được compiler xử lý gì cả.

ANC: A

AC4D222: Theo em có cách nào cải thiện được hiệu năng của mã nguồn nữa không nếu người viết đã cải thiện như sau

for (i = 1; i<= 10; i++)

   x += strlen(str);

Y = 15 + strlen(str);

===Cách cải thiện===

len = strlen(str);

for (i = 1; i <= 10; i++) x += len;

Y = 15 + len;

A. Không còn cách cải thiện nào nữa

B. Cải thiện bằng cách biên dịch với tham số -O-5

C. Cải thiện bằng cách biên dịch với tham số -OO

D. Cải thiện bằng cách bỏ for mà viết x += 10\*len.

ANC: D

AC4K222: Theo em phát biểu nào dưới đây về việc cải thiện tốc độ chương trình là đúng:

A. Khi khai báo kiểu dữ liệu cho các biến n, m, sum; với bất kỳ kiểu dữ liệu nào (int, long, byte, short, double, float) thì phần trăm cải thiện tốc độ đều ngang nhau

B. Khi khai báo kiểu dữ liệu cho các biến n, m, sum; với kiểu dữ liệu số nguyên (int, long, byte, short) thì chương trình bên phải sẽ cải thiện tốc độ tốt hơn hẳn với kiểu số thực (float, double)

C. Khi khai báo kiểu dữ liệu cho các biến n, m, sum; với kiểu dữ liệu số thực (float, double) thì chương trình bên phải sẽ cải thiện tốc độ tốt hơn hẳn với kiểu số thực (int, long, short, byte).

D. Khi khai báo kiểu dữ liệu cho các biến n, m, sum; với bất kỳ kiểu dữ liệu nào (int, long, byte, short, double, float) thì chương trình bên phải đều chẳng cải thiện tốc độ được gì cả

ANC: C

Vì việc cộng trừ nhân chia số thực sẽ tốn nhiều thời gian cho bộ xử lý hơn. Và điều đó khiến chương trình bên phải sẽ tiết kiệm được nhiều thời gian hơn nếu các kiểu dữ liệu n, m, sum là số thực.

Hẹn đến thứ 7 tuần này sẽ gửi cho lớp mã nguồn chứng minh hiệu năng được cải thiện nhiều (khi dùng chương trình bên phải) nếu khai báo m, n, sum là số thực

=================Ngày 15/04 (Học bù ngày 10/4)=====================

AC3M222: Theo em phát biểu nào dưới đây về chương trình này là đúng

#include <iostream>

using namespace std;

auto maxval(auto x, auto y){

    return (x > y) ? x : y;

}

int main() {

    int i = maxval(3, 7);

    cout << i << endl;

    double d = maxval(3.0, 7.0);

    cout << d << endl;

    int ch = maxval(3, 7.0);

    cout << ch << endl;

    return 0;

}

A. Sẽ in ra 7 7 7 trên 3 dòng khác nhau

B. Sẽ in ra 777 trên một dòng

C. Sẽ in ra duy nhất 7 trên một dòng

D. Sẽ không in ra gì cả

E. Tất cả phát biểu còn lại đều sai, chương trình lỗi.

ANC: A

AC3M222: Theo em phát biểu nào dưới đây về chương trình này là đúng

#include <iostream>

using namespace std;

template <typename T>

T maxval(T x, T y){

    return (x > y) ? x : y;

}

int main() {

    int i = maxval(3, 7);

    cout << i << endl;

    double d = maxval(3.0, 7.0);

    cout << d << endl;

    int ch = maxval(3, 7.0);

    cout << ch << endl;

    return 0;

}

A. Sẽ in ra 7 7 7 trên 3 dòng khác nhau

B. Sẽ in ra 777 trên một dòng

C. Sẽ in ra duy nhất 7 trên một dòng

D. Sẽ không in ra gì cả

E. Tất cả phát biểu còn lại đều sai, chương trình lỗi.

ANC: E

AC3M222: Theo em phát biểu sau đây có đúng hay không?

Với Student là kiểu dữ liệu do LTV tự định nghĩa, và có hàm được khái quát như sau

template <typename T>

T maxValue(T x, T y){

return (x > y ? x : y);

}

A. Ta vẫn truyền được Student làm tham số cho hàm khái quát trên.

B. Ta không truyền được kiểu dữ liệu tự định nghĩa cho hàm khái quát

C. Ta không truyền được kiểu Student vì không thể áp dụng kiểu đó cho toán tử ba ngôi kia

D. Ta không truyền được kiểu Student vì không thể nào thực hiện phép so sánh > cho Student kia

ANC: A

Tối sẽ có học online trên Teams lúc 7h30->9h10

Lớp offline đã dừng ở slide trang 31 của chương 3.

Offline:

C3M222: Theo em thì ta có thể đa năng hoá toán tử sao cho nó có đến 3 tham số được hay không?

A. Được nếu toán tử đó là toán tử \*

B. Được nếu toán tử đó là toán tử +

C. Được nếu toán tử đó là toán tử -

D. Được nếu toán tử đó là toán tử /

E. Tất cả các toán tử +,-,\*,/ đều được

F. Tất cả các toán tử +,-,\*,/ đều không được.

ANC: F

AC3D222: Theo em thì khi nào khai báo hàm như sau sẽ bị lỗi

int f(int x, int y = 8){

cout<<x<<y<<endl;

return (x + y);

}

A. Khai báo như thế là đã bị lỗi.

B. Nếu ở trên ta đã viết prototype của hàm f

C. Nếu ở trên ta không viết prototype của hàm f.

D. Nếu ta chỉ để hàm f( ) có một tham số ngầm định.

ANC: B

ANC3M222: Theo em đáp án nào sai trong số các đáp án sau? (được phép chọn nhiều)

|  |  |  |
| --- | --- | --- |
| void Swap(int &X, int &Y){  int Temp = X;  X = Y;  Y = Temp;  }//Hàm (1) | void Swap(int \*X, int \*Y){  int Temp = \*X;  \*X = \*Y;  \*Y = Temp;  }//Hàm (2) | void Swap(int X, int Y){  int Temp = X;  X = Y;  Y = Temp;  }//Hàm (3) |

A. Cả ba hàm trên đều không thể hoán đổi được giá trị của X, Y nếu kết thúc hàm

B. Cả ba hàm trên đều hoán đổi được giá trị của X, Y nếu kết thúc hàm

C. Có hàm (1) là hoán đổi được giá trị của X, Y nếu kết thúc hàm

D. Có hàm (2) là hoán đổi được giá trị của X, Y nếu kết thúc hàm

E. Có hàm (3) là hoán đổi được giá trị của X, Y nếu kết thúc hàm

F. Chỉ có hàm (1) là không hoán đổi được giá trị của X, Y nếu kết thúc hàm

G. Chỉ có hàm (2) là không hoán đổi được giá trị của X, Y nếu kết thúc hàm

H. Chỉ có hàm (3) là không hoán đổi được giá trị của X, Y nếu kết thúc hàm

ANC: A, B, E, F, G

C3M222: Theo em các phát biểu nào dưới đây về đa năng hoá hàm là đúng

A. Không ngôn ngữ nào cho phép khai báo hai file mã nguồn cùng tên nhau cũng như hai hàm cùng tên nhau.

B. Một số ngôn ngữ cho phép nhưng ta không nên viết hai file mã nguồn cùng tên cũng như hai hàm cùng tên nhau (tất nhiên khác tham số).

C. Một số ngôn ngữ cho phép nhưng ta không nên viết hai file mã nguồn cùng tên nhau, có điều hai hàm cùng tên nhau (tất nhiên khác tham số) thì vẫn nên viết

D. Một số ngôn ngữ cho phép nhưng ta không nên viết hai hàm mã nguồn cùng tên nhau (dù cho khác tham số), có điều hai file cùng tên nhau (tất nhiên khác thư mục) thì vẫn nên viết.

ANC: C

Hai file cùng tên nhau là không nên vì rất khó phân biệt chúng. Nhưng hai hàm có thể trùng tên nhau được do chỉ cần khác tham số với nhau.

C3K222: Theo em thì trong trường hợp nào hàm f( ) (tự định nghĩa) có thể gọi được hàm main và nó vẫn được main gọi đến?

A. Không thể làm thế được vì không hàm nào được phép gọi đến hàm main.

B. Vẫn được phép nhưng chương trình sẽ không bao giờ dừng lại do hàm main được gọi, hàm đó lại gọi hàm f( ) và hàm f( ) lại gọi hàm main.

C. Vẫn được phép nếu hàm f( ) có kiểu trả về là void.

D. Vẫn gọi được và vẫn có thể thiết lập cho chương trình dừng lại được

ANC: D

Giải thích: <https://onlinegdb.com/KViMOeHCn>

=================Ngày 03/4======================

C2D222: Đoạn lệnh sau cho kết quả gì?

int main(void){

int x = 1, y = 1;

if(x == 0 || ++y)

cout<<”x = ”<<x;

cout<<” y = ”<<y;

}

A. x = 2 y = 1

B. x = 1 y = 2

C. x = 2 y = 2

D. x = 1 y = 1

E. Không chạy được do viết main(void)

F. Thiếu return 0 nên không chạy được

G. Đáp án khác

ANC:

C2D222: Theo em phát biểu nào đúng về con trỏ \*void

A. Con trỏ này giúp ta gán giá trị cho vùng nhớ mà không cần ép kiểu

B. Có con trỏ này thì ta mới gọi được hàm malloc

C. Có con trỏ này thì ta mới gán được địa chỉ của các đối tượng mà được khởi tạo bằng câu lệnh new

D. Con trỏ này giúp ta gán nó là địa chỉ của bất kỳ biến nào.

ANC: D

AC2K222: theo em mã nguồn sau đây có chứa cú pháp của C++ hay không?

#include <stdio.h>

#include <stdlib.h>

int main() {

    int n;

    long total=100, x, \*arr;

    printf("Nhap n = "); scanf("%d",&n);

    arr = new long [n];

    if (arr==NULL) {

        printf("Cap phat khong thanh cong\n");

        exit(1);

    }

    for (int i=0; i < n; i++){

        printf("\n Vao so thu %d: ", i+1 );

        scanf("%d", &arr[i] );

    }

    printf("Danh sach cac so: \n");

    for (int i=0; i < n; i++) printf("%d ", arr[i]);

    delete [] arr;

    return 0;

}

A. Tất cả đều là cú pháp cho phép của C, không hề có cú pháp C++

B. Tất cả đều là cú pháp cho phép của C++, không hề có cú pháp C

C. Có new, delete là cú pháp của C++

D. Có new là cú pháp của C++

E. Có delete là cú pháp của C++

F. Có khai báo vòng lặp for là cú pháp của C++

G. Có new, delete và khai báo vòng for là cú pháp của C++

ANC: G

C2D222: Theo em nên khai báo cấu trúc dữ liệu nào để biểu thị danh sách các con cờ trong bàn cờ tướng

A. Khai báo mảng một chiều

B. Khai báo mảng hai chiều

C. Khai báo mảng ba chiều

D. Khai báo mảng bốn chiều

ANC: B

C2D222: Theo em nên khai báo cấu trúc dữ liệu nào để biểu thị danh sách các phòng trong các toà nhà chung cư của khu Vinhome Royal City  (giả sử đã có cấu trúc Room biểu thị một phòng)

A. Khai báo mảng một chiều

B. Khai báo mảng hai chiều

C. Khai báo mảng ba chiều

D. Khai báo mảng bốn chiều

ANC: D

C2D222: Theo em nên khai báo cấu trúc dữ liệu nào để biểu thị danh sách các phòng trong một toà nhà chung cư (giả sử đã có cấu trúc Room biểu thị một phòng)

A. Khai báo mảng một chiều

B. Khai báo mảng hai chiều

C. Khai báo mảng ba chiều

D. Khai báo mảng bốn chiều

ANC: C

C2K222: Theo em phát biểu nào dưới đây là đúng

long\*\* arr = new long[10]; //(1)

long\* brr = new long[10]; //(2)

long crr = new long[10]; //(3)

A. Thao tác (1) đúng

B. Thao tác (2) đúng

C. Thao tác (3) đúng

D. Tất cả thao tác trên đều đúng

E. Tất cả thao tác trên đều sai

ANC: B

Giải thích, từ khoá new trong C++ sẽ trả về một con trỏ.

C2K222: theo em phát biểu nào dưới đây là đúng, biết rằng A là một lớp và ta gọi được PTKD mặc định của lớp đó

A\*\* arr = (A\*\*)malloc(4\*sizeof(A\*)); //(1)

arr[0] = new A( ); //(2)

arr[1] = A( ); //(3)

A brr[4]; //(4)

brr[0] = new A( ); //(5)

brr[1] = A( ); //(6)

A. Có hai trong số các câu lệnh trên là sai

B. Có một trong số các câu lệnh trên là sai

C. Không có câu lệnh nào là sai cả

D. Tất cả các câu lệnh trên đều sai

E. Đáp án khác

ANC: A

Giải thích câu lệnh (2) tương tự với việc A x = A( ); sau đó arr[0] = &x; Do đó ta có thể suy ra được đáp án cần chọn là A. Nguyên nhân là có sự khác biệt khi viết A( ) và new A( );

Chú ý rằng theo quan điểm của một số chuyên gia đầu ngành (như giáo sư Hager, tác giả của

Introduction to High Performance Computing for Scientists and Engineers) thì việc khai báo mảng

int arr[5]; chưa đảm bảo được các phần tử của mảng được bố trí liên tiếp nhau trong bộ nhớ

muốn chúng bố trí liên tiếp nhau thì phải “GÁN giá trị liên tiếp nhau"

Tức là:

for(int i = 0; i < 5; i++)

{

arr[i]  = -1 ;//hoặc bất kỳ con số cụ thể nào

}

Quan điểm này có thể chưa được cập nhật/đồng ý bởi 1 số chuyên gia khác. Tuy vậy nếu các bạn gặp phải hiện tượng khai báo arr như trên và nếu chúng ta gán các giá trị của mảng không liên tục thì có thể vị trí trong bộ nhớ của các phần tử mảng không nằm kế tiếp nhau.

Chi tiết đọc trong chương 2, 3 của “Introduction to High Performance Computing for Scientists and Engineers” <https://blogs.fau.de/hager/hpc-book>

C2D: Theo em đâu là cách có thể cấp phát số lượng phần tử tuỳ ý

int N;

cin>>N;

int arr[N]; //cách (1)

int \*brr = (int)malloc(N\*sizeof(int)); //cách (2)

A. Cách (1) không thể làm được do sai cú pháp.

B. Cách (2) không thể làm được do sai cú pháp.

C. Cả 2 cách đều sai.

D. Cả 2 cách đều đúng

ANC: B

C2D: Theo em đâu là cách có thể cấp phát số lượng phần tử tuỳ ý

int N;

cin>>N;

int arr[N]; //cách (1)

int \*brr = (int\*)malloc(N\*sizeof(int)); //cách (2)

A. Cách (1) không thể làm được do sai cú pháp.

B. Cách (2) không thể làm được do sai cú pháp.

C. Cả 2 cách đều sai.

D. Cả 2 cách đều đúng

ANC: D

=================Ngày 27/3======================

Ngày 27/3 đã kết thúc ở slide trang 23 chương 2

C2K: Với lớp C++ như sau:

class A{ };

trong hàm main ta viết:

A\*\* arr = (.....A…...) malloc(10\*sizeof(.....B….) );

for(int i = 0; i < 10; i++)

{

arr….C… = new A( );

}

Theo em nên điền gì vào các chỗ trống A, B, C

A. A = A, B = A, C = [i]

B. A = \*A, B = A, C không điền gì

C. A = A\*, B = A, C = [i]

D. Tất cả đều sai

E. A = A\*\*, B = A\*, C = [i]

ANSWER: E

Giải thích: <https://onlinegdb.com/5CzXhhsKB1>

Vì new A( ) sẽ trả về một con trỏ mà trỏ đến đối tượng của lớp A

C2D: Theo em phát biểu nào đúng dưới đây về con trỏ \*void

A. Nó được dùng làm giá trị trả về cho hàm malloc của thư viện <stdlib.h>

B. Nó được dùng làm giá trị trả về cho hàm abs của thư viện <stdlib.h>

C. Nó được dùng làm giá trị trả về cho hàm sqrt của thư viện <cmath>

D. Nó được dùng làm giá trị trả về cho hàm main

E. Tất cả đều sai

ANSWER: A

C2D: Theo em phát biểu nào dưới đây là đúng về sự khó học của C/C++

A. Có bạn cho rằng một trong nguyên nhân là C lại cho phép struct kế thừa struct

B. Có bạn cho rằng vì toán tử \* có lúc dùng để nhân, có lúc dùng để khai báo, có lúc dùng để lấy giá trị của vùng nhớ

C. Có bạn cho rằng vì toán tử + có lúc dùng để cộng, có lúc dùng để tăng lên 1, có lúc dùng để biểu diễn dấu dương

D. Có bạn cho rằng vì toán tử - có lúc dùng để trừ, có lúc dùng để giảm xuống 1, có lúc dùng để biểu diễn dấu âm

E. Tất cả các phát biểu của các bạn đều đúng

F. Tất cả các phát biểu của các bạn đều sai

ANSWER: B

C1D: Theo em đây là cấu trúc gì?

A. Cấu trúc if…else

B. Cấu trúc switch case

C. Cấu trúc goto

D. Cấu trúc loop

ANSWER: A

C1M: Theo em nhóm công việc sau đây được xếp vào loại nào?

Declare variables, Print Report Heading, Print Detail Line

A. Initialization

B. Process

C. Wrap-up

D. Tất cả đều sai

ANSWER: D

C1D: "Integrated Development Environment" (IDE) là gì trong lập trình?

A. Một quy trình kiểm tra chất lượng phần mềm trước khi phát hành

B. Một phần mềm đồng bộ hoá các công cụ để giúp lập trình viên phát triển, kiểm thử và gỡ lỗi phần mềm.

C. Một tập hợp các hướng dẫn và tài liệu về lập trình

D. Tất cả đều sai

ANSWER: B

C1M: Với mã nguồn sau, nếu muốn không viết std trước cout thì ta làm gì?

#include <iostream>

int main()

{

    char str[] = "Xin chao C++";

    std::cout<<str<<std::endl;

    return 0;

}

A. Không thể bỏ được std

B. Chỉ việc bỏ std là xong

C. Bỏ cả std và toán tử :: ở bên cạnh

D. Viết câu lệnh using namspace std;

E. Tất cả đều sai

ANSWER: E

C1D: Khi lập trình, "object-oriented programming" (OOP) là gì?

A. Phương pháp lập trình mà tập trung vào việc phát triển các đối tượng và quan hệ giữa chúng.

B. Phương pháp lập trình dựa trên các hàm và thủ tục

C. Phương pháp lập trình dựa trên các kịch bản

D. Tất cả đều sai

E. Phương pháp lập trình dựa trên các luật logic

ANSWER: A

C1D: Trong lập trình, "syntax" đề cập đến điều gì?

A. Cách viết và đặt tên các biến, hàm và lớp trong mã nguồn

B. Cách phát triển và kiểm tra phần mềm

C. Các quy tắc và ngữ pháp để sử dụng ngôn ngữ lập trình.

D. Tất cả đều sai

ANSWER: C

C1D: Khi lập trình, định nghĩa "algorithm" là gì?

A. Một cách thức lập kế hoạch để phát triển phần mềm

B. Một cách thức kiểm tra chất lượng phần mềm

C. Một tập hợp các quy tắc hoặc chỉ thị cho việc giải quyết một vấn đề hoặc thực hiện một tác vụ.

D. Tất cả đều sai

ANSWER: C

Một số phản hồi:

1. Có bạn cho rằng nên dùng Teams để phát trực tiếp hoặc record bài giảng đến với các bạn ngồi xa. Tuy vậy điều này gặp một rào cản rất lớn là vấn đề chất lượng mạng. Như hôm đầu tiên, thầy đã dùng mạng của di động truy cập web nhưng rất khó vào.

2. Một số bạn cho rằng nên paste các câu hỏi trắc nghiệm đã chuẩn bị trước. Vì việc thầy nghĩ và gõ từng từ trực tiếp trên lớp sẽ rất mất thời gian. Ý kiến được chấp thuận, nhưng chúng ta cũng cần tính đến tình huống có bạn đưa câu hỏi trên lớp và thầy thấy câu hỏi đó hay, có thể làm câu trắc nghiệm được thì thầy vẫn phải nhập ra ngay trên lớp, trả lời/giải đáp và cộng điểm cho bạn đó. Những bạn đưa ra câu hỏi ngay trên lớp thì cần được ưu tiên phản hồi

3. có bạn cho rằng thầy cần nhấn mạnh các ý trọng tâm, quan trọng. Điều này lại không thể được, vì nếu nhấn mạnh vào đấy thì có thể các bạn chỉ tập trung ôn tủ. Trong khi các câu trong ngân hàng đề có thể hỏi bất kỳ kiến thức nào trong slide. Thầy không thể biết phần kiến thức nào không xuất hiện trong ngân hàng đề mà loại bỏ được.

4. Do có 1/2 lớp đi thi thử nên sau khi thi thử xong, toàn bộ đề sẽ được public cho các bạn còn lại trong lớp

=================Ngày 20/3=================

Nguyễn Tiến Thành

Email: [nguyenthanh@soict.hust.edu.vn](mailto:nguyenthanh@soict.hust.edu.vn), nhớ đừng gửi nhầm

Sdt: 0987768886

Chính sách của lớp

+ Sẽ có điểm danh vào đầu tiết 6 hàng tuần

+ SV được gửi mail xin phép vắng trước khi vào lớp, nhưng vắng 2 buổi có phép thì bị coi như vắng một buổi không phép. SV vắng >= 8 buổi chắc chắn sẽ bị 0 điểm QT. Chấp nhận trường hợp gửi mail xin nghỉ nhưng sau đó vẫn vào lớp trước khi điểm danh. Mail xin phép sẽ không được chấp nhận nếu gửi mail không nói rõ buổi học (thứ 2, sáng 2 tiết cuối) hoặc mail xin nghỉ hơn một buổi (trừ các mail xin nghỉ tất cả các buổi còn lại và gửi trước 03/4)

+ SV được phép xin nghỉ tất cả các buổi trừ những buổi kiểm tra giữa kỳ (nếu có), với điều kiện phải có đơn (kèm chữ ký và photo thẻ SV) trước 3/4. Khi ấy bạn SV đó sẽ không bị đọc tên khi điểm danh và không bị tích vắng ở các tuần học

+ Nếu hôm học nào mà lượng kiến thức chưa đủ (vì lý do bất khả kháng), và không kịp tiến độ thì sẽ có lớp học bù online trên Team. Buổi học online đó không điểm danh. Lý do bất khả kháng có thể là: mất điện hoặc GV giảng không kịp kiến thức. Những hôm nào giảng viên không thể đến lớp (và có mail báo lớp) thì sẽ học bù offline trên trường vào sáng thứ bảy. Học bù offline cũng không điểm danh

+ Trong quá trình học, SV được phép đặt câu hỏi (chuyên môn liên quan đến môn học) cho thầy. Nếu câu hỏi nào hay (có thể dùng làm câu hỏi trắc nghiệm cho cuộc thi) thì SV đó sẽ được cộng điểm QT.

+ Đi học đầy đủ vẫn KHÔNG được cộng điểm vào điểm QT hoặc CK.

+ Trong quá trình học sẽ có giao cho lớp các bài tập về nhà, BT về nhà này không bắt buộc làm, nhưng bạn nào gửi sớm nhất và đúng hết sẽ được cộng điểm vào điểm QT.

+ SV được quyền gửi câu hỏi riêng cho thầy qua email, nếu câu hỏi đó là câu hỏi được cả lớp quan tâm thì thầy sẽ công khai câu hỏi và câu trả lời cho lớp, tất nhiên sẽ giấu thông tin cá nhân của người hỏi.

+ Trong quá trình học thì SV được phép gửi các câu hỏi trắc nghiệm đóng góp cho đề thi CK. Nếu câu hỏi đó hay (không quá dễ, không quá khó, có câu trả lời đúng, và không trùng ngân hàng đề) thì những bạn gửi nhiều nhất sẽ được cộng điểm. Câu hỏi gửi phải theo Aiken format\*

+ Trên lớp thầy cũng có thể đưa ra các câu hỏi trắc nghiệm cho lớp ở tuần luyện tập/ôn tập (3, 9, 12, 14, 15, 17)

+ Tuần 17 có thể sẽ học online, nhưng chắc chắn chỉ học một tiết.

+ Dự kiến các tuần 1, 4, 8, 12, 16 sẽ gửi form khảo sát với lớp. SV có quyền điền form hoặc không và nếu điền thì sẽ được ẩn danh. Form đó sẽ giúp GV điều chỉnh lại việc học của lớp theo nguyện vọng chính đáng của SV.

+ SV nếu làm ồn hoặc các hành động không hợp lý khác sẽ bị đuổi ra khỏi lớp.

+ Do thi cuối kỳ là trắc nghiệm trên máy, nên có thể những bạn đóng góp rất nhiều câu hỏi lại không trúng câu này trong số đó, những bạn không đóng góp nhiều khi lại trúng.

+ Điểm QT sẽ là điểm thực hành Lab cộng với điểm thưởng của thầy (nếu có). Bạn nào làm các bài thực hành rất tốt thì sẽ được 10. Cộng điểm thưởng nếu trên 10 thì vẫn chỉ được 10 QT. Sẽ có thể có bài kiểm tra giữa kỳ, nhưng nếu tính điểm thì sẽ cũng chỉ là điểm cộng vào với điểm của bài thực hành. Bài kiểm tra giữa kỳ nếu có thì ở dạng trắc nghiệm và thi trên giấy tại lớp. Nếu là thi không lấy điểm (thi thử) thì chỉ cho phép một số bạn được tham gia thi.

+ Thầy đang hướng dẫn các bạn SV làm Web game 3D multiplayer, hi vọng tuần 8 sẽ có bản đầu tiên, có thể mời các bạn vào chơi thử. Nhưng nó là dòng **Serious Game** chứ không phải Entertainment Game, nên SV có thể thấy hơi nhàm chán. Vào để chơi chứ không phải để hack.

+ Sẽ có lúc bạn gửi mail xin nghỉ mà thầy không rep vì bận. Hoặc có thể có hôm gửi mail, thầy có req mà quên mất không ghi có phép cho bạn, thì SV được quyền phản hồi trên lớp để thầy sửa trên qldt.

TUẦN 1 (20/03)

Đề thi có thể có câu hỏi rất dễ như sau:

C1D: thứ tự thực hiện các nhóm lệnh trong một chương trình ngôn ngữ R được gọi là gì?

A. Luồng điều khiển (flow of Control)

B. Trình tự thực hiện lệnh (Order of Statements)

C. Các bước chỉ thị (Steps of Instructions)

D. Tất cả đều sai

ANSWER: A

C1M: Một đoạn mã của chương trình C++ viết như sau:

int a = b = 4;

if(b\*b & b >= 2){

int a = 2;

cout<<a;

}

Việc compiler xác định xem biến a được in ra là a ở ngoài hay trong vòng lặp thì được coi là gì?

A. Phân tích từ vựng

B. Phân tích cú pháp

C. Phân tích ngữ nghĩa

D. Tất cả đều sai

ANSWER: C

C1M: Một đoạn mã của chương trình C++ viết như sau:

int \*a = b = 4;

if(b\*b & b >= 2){

int a = 2;

cout<<a;

}

Việc compiler xác định xem biến a được khai báo đúng cách hay không thì được coi là gì?

A. Phân tích từ vựng

B. Phân tích cú pháp

C. Phân tích ngữ nghĩa

D. Tất cả đều sai

ANSWER: B

C1M: Theo em nhóm ngôn ngữ Imperative sẽ có các loại ngôn ngữ nào sau đây?

A. C

B. C++

C. Pascal

D. Java

E. C#

F. Tất cả đều đúng

ANSWER: F

C1M: Theo em nhóm ngôn ngữ Declarative sẽ có các loại ngôn ngữ nào sau đây?

A. C

B. C++

C. Pascal

D. Java

E. C#

F. Dart

G. Tất cả đều đúng

ANSWER: F

C1M: Theo em nhóm ngôn ngữ Functional Paradigm sẽ KHÔNG có (các) ngôn ngữ nào sau đây

A. C

B. Pascal

C. C++

D. Racket

E. Tất cả đều thuộc nhóm này

F. Tất cả ngôn ngữ trong các đáp án còn lại đều không thuộc nhóm này

G. Đáp án khác

ANSWER: C (lần trước chọn G là sai)

NGÀY 20/3 đã đến slide trang 19, chương 1

===================

\*Aiken format như sau:

Theo em một cộng một bằng mấy?

A. 1

B. 2

C. 3

D. 4

ANSWER: B

Không thể gán 2 mảng cho nhau

2 cấu trúc có thể gán thoải mái vì cấu trúc là 1 biến, tên mảng là hằng

///Chọn phát biểu đúng về tham số ngầm định trong C++

1. Tất cả đều sai
2. Tham số ngầm định có thể nằm ở vị trí bất kì trong hàm có danh sách tham số nhiều hơn các tham số ngầm định
3. Khi hàm có tham số ngầm định có prototype, thì giá trị ngầm định trong định nghĩa hàm có ý nghĩa cao hơn giá trị định nghĩa trong prototype
4. Khi đa năng hóa toán tử không được dùng tham số có giá trị ngầm định

/// Đoạn chương trình sau đây trả về kết quả như thế nào (giả sử 1 số integer là 4bytes)? #include<stdio.h>

int main() {

int \*\*\*r,\*\*q,\*p,i=8;

p=&i;

q=&p;

r=&q;

printf("%d, %d, %d\n", \*p,\*\*q,\*\*\*r);

return 0;

}

a. 4000, 4008, 4016

b. 8, 8, 8.

c. 8, 64, 512

d. 4000, 4004, 4008

///Cho 2 xâu char s1[6] ={‘H’,’a’,’n’,’o’,’I’}}và char s2[6] = “Hanoi” Quan hệ chính xác giữa hai xâu trên là :

A. s1>s2

B. s1<s2

C. s1>=s2

D. s1==s2

///Thư viện nào giúp chương trình dừng lại ngay lập tức khi vi phạm một điều kiện nào đó? a. exception.h b. Tất cả các đáp án còn lại đều đúng c. fault.h d. assert.h

/// Khi có 2 vòng lập lòng nhau thì nên : aChuyển vòng lặp ngắn ra ngoài C bChuyển vòng lặp ngắn ra ngoàinhưng cần lưu ý tránh làm mất ưu thế của bộ nhớ cache C. Chuyển vòng lập dài ra ngoài Chuyển vòng lặp dài ra ngoàinhưng cần lưu ý tránh làm mất ưu thế của bộ nhớ cache