1.[单选题]

使用ping命令测试RTT, 请问广州到北京的平均RTT与下面哪个值接近

200 ms

100 ms

50 ms

10 ms

2. [单选题]

64位系统下有 `char \* p[1];` 请问 `sizeof p` 的值是多少

1

8

16

32

3. [单选题]

如果有定义语句 int \*ptr[4]，typeid(ptr).name() 结果为 "A4\_Pi",

请问对 int \*(ptr[4]) ， typeid(ptr).name() 输出是

PPi

A4\_Pi

PA4\_i

A4\_i

4. [单选题]

下面关于 `inline` 说法正确的是

关键字 inline 是强制行为

关键字 inline 只是一种编译器建议

inline 不能避免函数调用时的参数传递带来的栈开销

inline 在运行时生效

5. [单选题]

下面对于友元函数描述正确的是

友元函数的实现必须在类的内部定义

友元函数是类的成员函数

友元函数破坏了类的封装性和隐藏性

友元函数不能访问类的私有成员

6[单选题]

请问对一个排好序的数组用二分法进行查找，时间复杂度为

O(1)

O(n)

O(logn)

O(nlogn)

7. [单选题]

下面关于HTTP状态码的说法哪个是正确的

403 Forbidden, 这是一个服务器错误

401 Unauthorized, 这是一个服务器错误

404 Not Found, 这是一个客户端错误

400 Bad Request, 这是一个服务器错误

8. linux 下使用 write 函数发送网络数据，函数返回 n(n>0),下面哪个说法正确

本地已经发送n个字节到系统缓冲

对端已经收到n个字节

本地已经发送n个字节到网络

本地发送失败

9. [不定项选择题]

下面哪些可以用于数据交换时的序列化和反序列化

protobuf

thrift

avro

rest

10. [不定项选择题]

声明语句为int a[3][4]; 下列表达式中与数组元素a[1][2]等价的是

\*(&a[0][0]+7)

\*(a[1]+2)

\*(a[0]+6)

a[7]

11. [不定项选择题]

下面哪些属于C++的惯用法

SIMD

SWAR

IMPL

RAII

12. [不定项选择题]

下面关于索引的说法哪些是正确的

在mysql上增加索引会提高性能，最坏的情况也能与不加持平

倒排索引常用于文档检索

B-Tree 索引常用于数据库

正确的索引可以极大加速数据查询速度

13. [不定项选择题]

面向对象的设计原则，下面哪些是正确的

SRP（单一职责原则）就一个类而言，应该仅有一个引起它变化的原因

OCP（开闭原则）软件实体应该是可以扩展的，但是不可以修改

DIP（依赖倒置原则）抽象不应该依赖于细节，细节应该依赖于抽象

LSP（Liskov替换原则）基类型必须能够替换掉它们的子类型

14. [不定项选择题]

有定义如下，

```

const char \*p1 = "hello";

char \*const p2 = new char[12]{'h', 'e', 'l', 'l', 'o'};

```

p1++;

p2++;

p1 = '-';

\*(p2+1) = '-';

15. [不定项选择题]

下面关于缓存的说法哪些正确

缓存经常用来减少磁盘读压力

在大数据系统中，数据的写入量越大更新越频繁，缓存越有价值

对应用做性能测试时有时需要避免缓存生效

硬盘也可以用来做缓存

16. [不定项选择题]

下面关于reactor模式的一些说法哪些是正确的

reactor 利用多线程来支持多个连接

reactor 中有阻塞io调用也不会影响并发性

reactor 是异步的

reactor 是非阻塞的

17. [问答题]

题目描述

最近开源的实时流处理系统heron中有如下代码（节选），请写出这段代码所用到的设计模式与惯用法及其各自适用的场景或原因

``` c++

class PCQueue {

public:

PCQueue() {}

virtual ~PCQueue() throw() {}

PCQueue(const PCQueue& pcqueue) = delete;

PCQueue& operator=(const PCQueue& pcqueue) = delete;

void enqueue(void\* \_item);

void\* dequeue(void);

protected:

std::queue<void\*> queue\_;

std::condition\_variable cond\_;

std::mutex mutex\_;

};

void PCQueue::enqueue(void\* \_item) {

std::unique\_lock<std::mutex> m(mutex\_);

queue\_.push(\_item);

cond\_.notify\_one();

}

void\* PCQueue::dequeue() {

std::unique\_lock<std::mutex> m(mutex\_);

while (queue\_.empty()) cond\_.wait(m);

void\* item = queue\_.front();

queue\_.pop();

return item;

}

18. [问答题]

题目描述

写一段代码展示RAII在C++中的使用

19. [问答题]

题目描述

c++11 增加了 std::memory\_order，请说明 memory\_order\_acquire， memory\_order\_release 的含义

20. [问答题]

题目描述

对于下面代码片断

void dump(const std::vector<int> &vec)

{

cout << vec.size() << ", ";

for (const auto &i : vec)

std::cout << i;

std::cout << std::endl;

}

std::vector<int> v{1, 2, 1, 4, 5, 6, 7};

dump(v); // a

auto itr = std::remove(v.begin(), v.end(), 1);

dump(v); // b

v.erase(itr, v.end());

dump(v); // c

v.erase(std::begin(v), std::end(v));

dump(v); // d

```

std::remove 算法实现如下面代码片断所示时

\_\_first = \_x0008\_std::find(\_\_first, \_\_last, \_\_value);

if(\_\_first == \_\_last)

return \_\_first;

\_ForwardIterator \_\_result = \_\_first;

++\_\_first;

for(; \_\_first != \_\_last; ++\_\_first)

if(!(\*\_\_first == \_\_value))

{

\*\_\_result = std::move(\*\_\_first);

++\_\_result;

}

return \_\_result;

注释 a 处的输出为 `7, 1214567`

请给出：

注释 b 处的输出为

注释 c 处的输出为

注释 d 处的输出为

21. [问答题]

题目描述

请说明布隆过滤器的实现方式、应用场景及误判的影响

22. 题目描述

请说明下面代码的输出结果

class Foo {

public:

Foo(int i) : a\_(i), b\_(a\_+1) {}

void dump() {

cout << a\_ << "," << b\_ << endl;

}

private:

int b\_;

int a\_;

};

int main() {

Foo foo(INT\_MAX);

foo.dump();

}

23. [问答题]

题目描述

有字符串 “aabbaabb”， 写当正则表达式为 (a.\*b)， (a.\*?b)， (a.??b) 各自匹配到的结果

24. [编程题] 实现my\_strtoint32()函数

时间限制：1秒

空间限制：32768K

实现函数 int32\_t my\_strtoint32(const string& str), 将一个10进制的ASCII码字符串转换成 int32\_t，支持正负号，字符串前后请允许空格, 数字串可以有前置0 `12345 == my\_strtoint32("12345");`

#define UINT32\_MAX 0xffffffff  #define EINVAL 22 #define ERANGE 34 ``` ``` 返回值与错误码参照 <stdlib.h> strtol 说明 Return Value The strtol() function returns the result of the conversion, unless the value would underflow or overflow. If an underflow occurs, strtol() returns LONG\_MIN. If an overflow occurs, strtol() returns LONG\_MAX. In both cases, errno is set to ERANGE. Errors EINVAL, The given base contains an unsupported value. ERANGE, The resulting value was out of range. ``` </stdlib.h>

输入描述:

输入包含一行:

输入一个字符串,包含数字，正负号，空格，前置0，字母。

输出描述:

在一行内输出 处理后的数字|errno值,如果字符串非法,处理后的数字为0

输入例子1:

a 123 123

输出例子1:

0|22 123|0

参考答案

CBBBC

CCA

9.ABC

10.BC

11.CD

12.BCD

13.ABC

14.AD

15.ACD

16.D

17.

producer consumer pattern

non-copyable

spurious wakeup

19.

memory\_order\_acquire: 所有后续读操作都必须在这条操作后完成

A load operation with this memory order performs the acquire operation on the affected memory location:

no reads in the current thread can be reordered before this load.

memory\_order\_release: 所有之前的写操作都必须在这条操作之前完成

A store operation with this memory order performs the release operation:

no writes in the current thread can be reordered after this store.

20.

7, 2456767

5, 24567

0,

23.(a.\*b), aabbaabb, aabbaabb

(a.\*?b), aabbaabb, aab

(a.??b), aabbaabb, aab