

每日一题day10_3月14日

一. 单选

1. 32位系统中，定义**a[3][4],则变量占用内存空间为()。

- A 4
- B 48
- C 192
- D 12

正确答案：B

2. 二维数组X按行顺序存储，其中每个元素占1个存储单元。若X[4][4]的存储地址为Oxf8b82140,X[9][9]的存储地址为Oxf8b8221c,则X[7][7]的存储地址为()。

- A Oxf8b821c4
- B Oxf8b821a6
- C Oxf8b82198
- D Oxf8b821c0

正确答案：A

3.

求函数返回值，输入x=9999

```
int func(int x){
    int count=0;
    while (x)
    {
        count++;
        x=x&(x-1);//与运算
    }
    return count;
}
```

- A 8
- B 9
- C 10
- D 12

正确答案：A

4.

根据下面递归函数：调用函数Fun（2），返回值是多少（）

```
int Fun(int n)
{
    if(n==5)
        return 2;
    else
        return 2*Fun(n+1);
}
```

- ☐ A 2
- ☐ B 4
- ☐ C 8
- ☐ D 16

正确答案：D

5.

执行下面语句后的输出为

```
int l=1;
if(l<=0)
    printf("****\n");
else
    printf("%%%%\n");
```

- ☐ A %%
- ☐ B ****
- ☐ C 有语法错，不能正确执行
- ☐ D %%%%

正确答案：A

6. 在C++，下列哪一个可以做为对象继承之间的转换()

- ☐ A static_cast
- ☐ B reinterpret_cast
- ☐ C dynamic_cast
- ☐ D const_cast

正确答案：C

7. 类模板的使用实际上是类模板实例化成一个具体的_____。

- ☒ A 类
- ☐ B 函数
- ☐ C 模板类
- ☐ D 对象

正确答案：A

8.
有如下C++代码：

```
struct A{
    void foo(){printf("foo");}
    virtual void bar(){printf("bar");}
    A(){bar();}
};
struct B:A{
    void foo(){printf("b_foo");}
    void bar(){printf("b_bar");}
};
```

那么

```
A *p=new B;
p->foo();
p->bar();
```

输出为：

- ☒ A barfoob_bar
- ☐ B foobarb_bar
- ☐ C barfoob_foo
- ☐ D foobarb_fpp

正确答案：A

9.
下面的程序输出可能是什么？

```
class Printer{
public:
    Printer(std::string name) {std::cout << name;}
};
class Container{
public:
```

```

    Container() : b("b"), a("a") {}
    Printer a;
    Printer b;
};
int main(){
    Container c;
    return 0;
}

```

- A 可能是 "ab" 或 "ba"。依赖于具体的实现
- B 一直都是 "ba"
- C 一直都是 "ab"

正确答案：C

10.
代码可以通过编译吗？如果不能应该如何修改？

```

template<class T> class Foo{
    T tVar;
public:
    Foo(T t) : tVar(t) {}
};

template<class T> class FooDerived:public Foo<T>
{
};

int main()
{
    FooDerived<int> d(5);
    return 0;
}

```

- A 代码可以正确通过编译。
- B 编译错误，FooDerived是一个继承模板类的非模板类，它的类型不能改变。
- C 编译错误，tVal变量是一个不确定的类型。
- D 编译错误，可以在FooDerived类中添加一个构造函数解决问题。

正确答案：D

二. 编程

1.
对于一个给定的井字棋棋盘，请设计一个高效算法判断当前玩家是否获胜。

给定一个二维数组`board`，代表当前棋盘，其中元素为1的代表是当前玩家的棋子，为0表示没有棋子，为-1代表是对方玩家的棋子。

测试样例：

`[[1,0,1],[1,-1,-1],[1,-1,0]]`

返回：`true`

输入描述：

输出描述：

示例1:

输入

输出

正确答案：

2.

密码按如下规则进行计分，并根据不同的得分为密码进行安全等级划分。

一、密码长度:

5 分: 小于等于4 个字符

10 分: 5 到7 字符

25 分: 大于等于8 个字符

二、字母:

0 分: 没有字母

10 分: 全都是小 (大) 写字母

20 分: 大小写混合字母

三、数字:

0 分: 没有数字

10 分: 1 个数字

20 分: 大于1 个数字

四、符号:

0 分: 没有符号

10 分: 1 个符号

25 分: 大于1 个符号

五、奖励:

2 分: 字母和数字

3 分: 字母、数字和符号

5 分: 大小写字母、数字和符号

最后的评分标准:

>= 90: 非常安全

>= 80: 安全 (Secure)

>= 70: 非常强

>= 60: 强 (Strong)

>= 50: 一般 (Average)

>= 25: 弱 (Weak)

>= 0: 非常弱

对应输出为 :

VERY_WEAK,

WEAK,

AVERAGE,

STRONG,

VERY_STRONG,

SECURE,

VERY_SECURE

请根据输入的密码字符串，进行安全评定。

注：

字母：a-z, A-Z

数字：0-9

符号包含如下：(ASCII码表可以在UltraEdit的菜单view->ASCII Table查看)

!"#\$%&'()*+,-./ (ASCII码：x21~0x2F)

;

```
<=>?@
```

 (ASCII码：x3A~0x40)

[\] ^ _ ` (ASCII码：x5B~0x60)

{ } ~ (ASCII码：x7B~0x7E)

接口描述：

Input Param

String pPasswordStr: 密码，以字符串方式存放。

Return Value

根据规则评定的安全等级。

```
public static Safelevel GetPwdSecurityLevel(String pPasswordStr)
{
    /*在这里实现功能*/
    return null;
}
```

输入描述：
输入一个string的密码

输出描述：
输出密码等级

示例1:
输入
38\$@NoNoNo
输出
VERY_SECURE

正确答案：