

## 每日一题day11\_5月31日

### 一. 单选

1.

在32位系统环境，编译选项为4字节对齐，那么sizeof(A)和sizeof(B)是：

```
struct A
{
    int a;
    short b;
    int c;
    char d;
};
struct B
{
    int a;
    short b;
    char c;
    int d;
};
```

- ☒ A 16,16
- ☐ B 13,12
- ☒ C 16,12
- ☐ D 11,16

正确答案：C

2. 以下函数中，和其他函数不属于一类的是\_\_\_\_\_。

- ☒ A fwrite
- ☐ B putc
- ☒ C pwrite
- ☐ D putchar
- ☐ E getline
- ☐ F scanf

正确答案：C

3.

当n=5时，下列函数的返回值是：

```
int foo(int n){
    if(n<2){
        return n;
    }
    else
        return 2*foo(n-1)+foo(n-2);
}
```

- ☐ A 5
- ☐ B 11
- ☐ C 29
- ☐ D 10

正确答案：C

4.  
以下程序的输出结果是：

```
#include <iostream.h>
void main() {
    int x=3,y=3;
    switch(x%2) {
        case 1:
            switch (y) {
                case 0:cout<<"first";
                case 1:cout<<"second";break;
                default: cout<<"hello";
            }
        case 2:cout<<"third";
    }
}
```

- ☐ A second third
- ☐ B hello
- ☐ C first second
- ☐ D hellothird

正确答案：D

5.  
下列代码试图打印数字1-9的全排列组合。

```
#include "stdio.h"
```

```
#define N 9
int x[N];
int count = 0;

void dump() {
    int i = 0;
    for (i = 0; i < N; i++) {
        printf("%d", x[i]);
    }
    printf("\n");
}

void swap(int a, int b) {
    int t = x[a];
    x[a] = x[b];
    x[b] = t;
}

void run(int n) {
    int i;
    if (N - 1 == n) {
        dump();
        count ++;
        return;
    }
    for (i = ____; i < N; i++) {
        swap(____, i);
        run(n + 1);
        swap(____, i);
    }
}

int main() {
    int i;
    for (i = 0; i < N; i++) {
        x[i] = i + 1;
    }
    run(0);
    printf("* Total: %d\n", count);
}
```

其中run函数中缺失的部分应该依次为：

- ☒ A n+1, n, n+1
- ☐ B n+1, n, n
- ☐ C n, n, n
- ☐ D n, n+1, n+1
- ☐ E n+1, n+1, n+1
- ☐ F n, n, n+1

正确答案：C

6. 下列哪个用法哪个是错误的()

- A int \*a;
- B extern const int array[256];
- C const int &ra;
- D typedef void (\*FUN)();

正确答案：C

7. 在重载一个运算符为成员函数时，其参数表中没有任何参数，这说明该运算符是（ ）。

- A 无操作数的运算符
- B 二元运算符
- C 前缀一元运算符
- D 后缀一元运算符

正确答案：C

8. 若PAT是一个类，则程序运行时，语句“PAT(\*ad)[3];”调用PAT的构造函数的次数是（ ）。

- A 2
- B 3
- C 0
- D 1

正确答案：C

9.  
以下程序输出结果是\_\_\_\_\_。

```
class A
{
public:
    virtual void func(int val = 1)
    { std::cout<<"A->"<<val <<std::endl;}
    virtual void test()
    { func();}
};
class B : public A
{
public:
    void func(int val=0)
```

```
{std::cout<<"B->"<<val <<std::endl;}
};
int main(int argc ,char* argv[])
{
    B*p = new B;
    p->test();
    return 0;
}
```

- ☐ A A->0
- ☐ B B->1
- ☐ C A->1
- ☐ D B->0
- ☐ E 编译出错
- ☐ F 以上都不对

正确答案：B

10. 下面程序的输出是（ ）

```
class A
{
public:
    void foo()
    {
        printf("1");
    }
    virtual void fun()
    {
        printf("2");
    }
};
class B: public A
{
public:
    void foo()
    {
        printf("3");
    }
    void fun()
    {
        printf("4");
    }
};
int main(void)
{
```

```
A a;
B b;
A *p = &a;
p->foo();
p->fun();
p = &b;
p->foo();
p->fun();
A *ptr = (A *)&b;
ptr->foo();
    ptr->fun();
return 0;
}
```

- ☐ A 121434
- ☐ B 121414
- ☐ C 121232
- ☐ D 123434

正确答案：B

## 二. 编程

1. 标题：最近公共祖先 | 时间限制：3秒 | 内存限制：32768K | 语言限制：[Python, C++, C#, Java]

有一棵无穷大的满二叉树，其结点按根结点一层一层地从左往右依次编号，根结点编号为1。现在有两个结点a, b。请设计一个算法，求出a和b点的最近公共祖先的编号。

给定两个int a,b。为给定结点的编号。请返回a和b的最近公共祖先的编号。注意这里结点本身也可认为是其祖先。

测试样例：

2, 3

返回：1

输入描述：

输出描述：

示例1:

输入

输出

正确答案：

2. 标题：求最大连续bit数 | 时间限制：1秒 | 内存限制：32768K

功能: 求一个byte数字对应的二进制数字中1的最大连续数，例如3的二进制为00000011，最大连续2个1

输入: 一个byte型的数字

输出: 无

返回: 对应的二进制数字中1的最大连续数

输入描述：

输入一个byte数字

输出描述：

输出转成二进制之后连续1的个数

示例1:

输入

3

输出

2

正确答案：