

每日一题day23_3月29日

一. 单选

1. 在C语言中，设有数组定义：char arrays[]="China"；则数组array所占用的空间为（ ）。

- ☐ A 4个字节
- ☐ B 5个字节
- ☐ C 6个字节
- ☐ D 7个字节

正确答案：C

2.

运行下面这段代码，会出现的情况是:()

```
void GetMemory(char *p)
{
    p = (char *)malloc(100);
}
void Test(void)
{
    char *str = NULL;
    GetMemory(str);
    strcpy(str, "hello world");
    printf(str);
}
```

- ☐ A hello world
- ☐ B 显示为乱码
- ☐ C 程序崩溃
- ☐ D hello

正确答案：C

3.

```
void getmemory(char *p)
{
    p=(char*)malloc(100);
}
void test(void)
```

```
{
    char * str = null;
    getmemory(str);
    strcpy(str,"hello,world");
    printf(str);
}
```

请问运行test函数会有什么样的结果？

- ☐ A 编译错误
- ☐ B 输出"hello world"
- ☐ C 输出""
- ☐ D segmentation fault

正确答案：D

4.
下面程序的功能是什么？

```
main()
{
    long num=0;
    FILE *fp;
    if((fp=fopen("fname.dat","r"))==NULL)
    {
        printf("Can't open the file! ");
        exit(0);
    }
    while(!feof(fp))
    {
        fgetc(fp);
        num++ ;
    }
    printf("num=%d\n",num) ;
    fclose(fp) ;
}
```

- ☐ A 拷贝文件
- ☐ B 统计文件的字符数
- ☐ C 统计文件的单词数
- ☐ D 统计文件的行数

正确答案：B

5.

下列程序执行后输出的结果是

```
#include<iostream>
using namespace std;

int f(int a)
{
    int b = 0;
    static int c = 3;
    a = c++,b++;
    return (a);
}

int main()
{
    int a = 2, i, k;
    for (i = 0; i < 2; i++)
        k = f(a++);
    printf(" %d\n", k);

    return 0;
}
```

- ☒ A 3
- ☐ B 0
- ☐ C 5
- ☐ D 4

正确答案：D

6. 下面对静态数据成员的描述中，正确的是？

- ☒ A 静态数据成员可以在类体内进行初始化
- ☐ B 静态数据成员不可以被类的对象调用
- ☐ C 静态数据成员不受private控制符的作用
- ☐ D 静态数据成员可以直接用类名调用

正确答案：D

7. 下面关于虚函数和函数重载的叙述不正确的是

- ☒ A 虚函数不是类的成员函数
- ☐ B 虚函数实现了C++的多态性
- ☐ C 函数重载允许非成员函数，而虚函数则不行

D 函数重载的调用根据参数的个数、序列来确定，而虚函数依据对象确定

正确答案：A

8. 下面有关c++线程安全，说法错误的是？

- A 线程安全问题都是由全局变量及静态变量引起的
- B 若每个线程中对全局变量、静态变量只有读操作，而无写操作，一般来说，这个全局变量是线程安全的；若有多个线程同时执行写操作，一般都需要考虑线程同步，否则的话就可能影响线程安全
- C c++标准库里面的string保证是线程安全的
- D POSIX线程标准要求C标准库中的大多数函数具备线程安全性

正确答案：C

9. void recursive(int n, int m, int o)

```
{
    if (n <= 0)
    {
        printf("%d,%d\n", m, o);
    }
    else
    {
        recursive(n - 1, m + 1, o);
        recursive(n - 1, m, o + 1);
    }
}
```

以上函数的时间复杂度（ ）

- A $O(n*m*o)$
- B $O(n^2*m^2)$
- C $O(2^n)$
- D $O(n!)$

正确答案：C

10. 下面程序的执行结果：

```
class A{
public:
    long a;
};
class B : public A {
public:
    long b;
};
void seta(A* data, int idx) {
```

```
    data[idx].a = 2;
}
int main(int argc, char *argv[]) {
    B data[4];
    for(int i=0; i<4; ++i){
        data[i].a = 1;
        data[i].b = 1;
        seta(data, i);
    }
    for(int i=0; i<4; ++i){
        std::cout << data[i].a << data[i].b;
    }
    return 0;
}
```

- ☐ A 11111111
- ☐ B 12121212
- ☐ C 11112222
- ☐ D 21212121
- ☐ E 22221111

正确答案：E

二. 编程

1.

春节期间小明使用微信收到很多个红包，非常开心。在查看领取红包记录时发现，某个红包金额出现的次数超过了红包总数的一半。请帮小明找到该红包金额。写出具体算法思路和代码实现，要求算法尽可能高效。

给定一个红包的金额数组gifts及它的大小n，请返回所求红包的金额。

若没有金额超过总数的一半，返回0。

测试样例：

[1,2,3,2,2],5

返回：2

输入描述：

输出描述：

示例1:

输入

输出

正确答案：

2.

Levenshtein 距离，又称编辑距离，指的是两个字符串之间，由一个转换成另一个所需的最少编辑操作次数。许可的编辑操作包括将一个字符替换成另一个字符，插入一个字符，删除一个字符。编辑距离的算法是首先由俄国科学家Levenshtein提出的，故又叫Levenshtein Distance。

Ex：

字符串A:abcdefg

字符串B: abcdef

通过增加或是删掉字符”g”的方式达到目的。这两种方案都需要一次操作。把这个操作所需要的次数定义为两个字符串的距离。

要求：

给定任意两个字符串，写出一个算法计算它们的编辑距离。

请实现如下接口

```
/*
```

```
功能：计算两个字符串的距离
```

```
*
```

```
输入：
```

```
字符串A和字符串B
```

```
*
```

```
输出：无
```

```
*
```

```
返回：如果成功计算出字符串的距离，否则返回-1
```

```
*/
```

```
public
```

```
static
```

```
int
```

```
style="font-family: 'courier new';font-size: 13.0px;"> calStringDistance ( style="background:  
silver;">String charA, String charB)
```

```
{
```

```
return
```

```
0;
```

```
}
```

输入描述：
输入两个字符串

输出描述：
得到计算结果

示例1:
输入
abcdefg
abcdef

输出
1

正确答案：