

## 选择题

1、声明一个指向含有10个元素的数组的指针，其中每个元素是一个函数指针，该函数的返回值是int，参数是int\*，正确的是（）

- A、`(int *p[10])(int*)`
- B、`int [10]*p(int *)`
- C、`int ((*p)[10])(int *)`
- D、`int ((int *)[10])*p`

2、下列关于new delete 与malloc free 的联系与区别描述正确的是（）

- A、都是在栈上进行动态的内存操作
- B、用malloc函数需要指定内存分配的字节数并且不能初始化对象，new会自动调用对象的构造函数
- C、delete和free的区别仅在于delete可以调用析构函数
- D、以上说法都不正确

3、有如下程序段

```
#include<stdio.h>
int main(){
    int a, b;
    for(a = 1, b = 1; a <= 100; a++){
        if(b >= 20)
            break;
        if(b % 3 == 1){
            b = b + 3;
            continue;
        }
        b = b - 5;
    }
    printf("%d\n", a);
    return 0;
}
```

它的输出为（）

- A、10
- B、9
- C、8
- D、7

4、有如下程序段

```
class A {
    ...
private:
    int &a;
};
class B : public A {
    ...
private:
```

```

    int a;
public:
    const int b;
    A c;
    static const char* d;
    A* e;
};

```

构造函数中，成员变量一定要通过初始化列表来初始化的是（ ）

- A、 b c
- B、 b c d
- C、 b c e
- D、 b c d e

5、关于“深拷贝”，下列说法正确的是（ ）

- A、会拷贝动态分配的成员对象
- B、会拷贝成员数据的值
- C、会拷贝静态分配的成员对象
- D、以上说法都正确

6、下列运算符，在C++语言中可以重载的是（ ）

- A、 ?:
- B、 ::
- C、 .
- D、 ,

7、有如下程序段

```

#include<iostream>
using namespace std;
class A {
public:
    A(char* s)
    {
        cout << s << endl;
    } ~
    A() {}
};

class B :virtual public A
{
public:
    B(char* s1, char* s2) :A(s1) {
        cout << s2 << endl;
    }
};

class C :virtual public A
{
public:
    C(char* s1, char* s2) :A(s1) {
        cout << s2 << endl;
    }
};

```

```

class D :public B, public C
{
public:
    D(char* s1, char* s2, char* s3, char* s4) :B(s1, s2), C(s1, s3), A(s1)
    {
        cout << s4 << endl;
    }
};
int main() {
    D* p = new D("class A", "class B", "class C", "class D");
    delete p;
    return 0;
}

```

它的输出为 ( )

- A、class A class B class C class D
- B、class D class B class C class A
- C、class D class C class B class A
- D、class A class C class B class D

8、SQL 查询语句中 WHERE、GROUP BY、HAVING 这些关键字区别和用法总结错误的是 ( )

- A、HAVING在查询语句中必须依赖于GROUP BY
- B、WHERE子句用来限制SELECT语句从表中指定选取的行
- C、GROUP BY子句用来分组WHERE子句的输出结果集
- D、HAVING子句用来从分组的结果中筛选列

9、下列关于C++11规范的说法中错误的是 ( )

- A、C++11中重新定义了auto关键字，使其失去了原有含义并赋予了新的含义
- B、C++11中给很多原本就有的关键字赋予了更多的含义，例如delete、default等
- C、C++11中引入了很多新的STL容器，例如multi\_set、forward\_list等
- D、C++11中引入了很多新的语法，例如范围for、Lambda表达式等

10、以下关于C++的描述中哪一个是正确的 ( )

- A、任何指针都必须指向一个实例
- B、子类指针不可以指向父类实例
- C、任何引用都必须指向一个实例
- D、引用所指向的实例不可能无效

## 编程题

### 1、前 K 个高频元素

#### 题目描述

给你一个链表的头 `head`，每个结点包含一个整数值。

在相邻结点之间，请你插入一个新的结点，结点值为这两个相邻结点值的 **最大公约数**。

请你返回插入之后的链表。

两个数的 **最大公约数** 是可以被两个数字整除的最大正整数。

## 样例:

### 示例 1:

```
输入: head = [18,6,10,3]
输出: [18,6,6,2,10,1,3]
```

### 示例 2:

```
输入: head = [7]
输出: [7]
```

## 提示:

- 链表中结点数目在 `[1, 5000]` 之间。
- `1 <= Node.val <= 1000`

**进阶:** 你所设计算法的时间复杂度 **必须** 优于  $O(n \log n)$  , 其中 `n` 是数组大小。

链接: <https://leetcode.cn/problems/insert-greatest-common-divisors-in-linked-list/description/>

## 2、统计一个圆中点的数目

### 题目描述

给你一个数组 `points` , 其中 `points[i] = [xi, yi]` , 表示第 `i` 个点在二维平面上的坐标。多个点可能会有 **相同** 的坐标。

同时给你一个数组 `queries` , 其中 `queries[j] = [xj, yj, rj]` , 表示一个圆心在 `(xj, yj)` 且半径为 `rj` 的圆。

对于每一个查询 `queries[j]` , 计算在第 `j` 个圆 **内** 点的数目。如果一个点在圆的 **边界上** , 我们同样认为它在圆 **内** 。

请你返回一个数组 `answer` , 其中 `answer[j]` 是第 `j` 个查询的答案。

## 样例:

### 示例:

```
输入: points = [[1,3],[3,3],[5,3],[2,2]], queries = [[2,3,1],[4,3,1],[1,1,2]]
输出: [3,2,2]
```

```
输入: points = [[1,1],[2,2],[3,3],[4,4],[5,5]], queries = [[1,2,2],[2,2,2],
[4,3,2],[4,3,3]]
输出: [2,3,2,4]
```

## 提示:

- `1 <= points.length <= 500`
- `points[i].length == 2`
- `0 <= xi, yi <= 500`
- `1 <= queries.length <= 500`

- `queries[j].length == 3`
- `0 <= xj, yj <= 500`
- `1 <= rj <= 500`
- 所有的坐标都是整数。

链接: <https://leetcode.cn/problems/queries-on-number-of-points-inside-a-circle/description/>