

选择题

1、数组a的定义语句为 `float a[3][4];`, 下列 () 是对数组元素不正确的引用方法。

- A、`a[i][j]`
- B、`*(a[i]+j)`
- C、`*(*(a+i)+j)`
- D、`*(a+i*4+j)`

2、有如下程序

```
#include <stdio.h>
main()
{
    char x = '\x80';
    printf("%x\n", x+1);
}
```

程序运行后的输出结果是 ()

- A、81
- B、ffffff81
- C、-79
- D、-127

3、设有定义: `double x=2.25;`, 以下不能完整输出变量x值的语句是 ()

- A、`printf("x=%5.0f\n", x);`
- B、`printf("x=%f\n", x);`
- C、`printf("x=%lf\n", x);`
- D、`printf("x=%0.5f\n", x);`

4、设有定义: `int x=11, y=12, z=0;`, 以下表达式值不等于12的是 ()

- A、`(z,x,y)`
- B、`(z=x,y)`
- C、`z=(x,y)`
- D、`z=(x==y)`

5、以下可以将任意一个两位整数n的个位数与十位数对换的表达式为 ()

- A、`(n-n/10*10)*10+n/10`
- B、`n-n/10*10+n/10`
- C、`n/10+(n-n/10)`
- D、`(n-n/10)*10+(n/10)`

6、以下非法的字符常量是 ()

- A、`'\n'`
- B、`'\472'`
- C、`'\x21'`
- D、`'\0'`

7、如果定义了一个类myclass, 则为myclass类对象以成员函数形式重载后++运算符的声明应该是 ()

- A、myclass operator++()
- B、myclass operator++(int)
- C、myclass &operator++()
- D、myclass &operator++(int)

8、以下关于C++语言中虚函数的描述正确的是 ()

- A、构造函数不可以是虚函数
- B、析构造函数不可以是虚函数
- C、纯虚函数不可以拥有函数体
- D、struct中不可以定义虚函数

9、以下C++代码的输出结果是 ()

```
#include <iostream>
using namespace std;
class A {
    A() {}
    ~A() {}
    static int a;
};
int main() {
    cout << sizeof(A) << endl;
    return 0;
}
```

- A、0
- B、1
- C、4
- D、编译不通过

10、有如下定义:

```
const char * const * pp;
```

则如下操作可以通过编译的是 ()

- A、pp++
- B、(*pp)++
- C、**pp = 0
- D、以上选项均不能通过编译

编程题

1、完全数计算

完全数 (Perfect number) , 又称完美数或完备数, 是一些特殊的自然数。

它所有的真因子 (即除了自身以外的约数) 的和 (即因子函数) , 恰好等于它本身。

例如: 28, 它有约数1、2、4、7、14、28, 除去它本身28外, 其余5个数相加, 1+2+4+7+14=28。

输入n, 请输出n以内(含n)完全数的个数。

输入描述:

输入一个数字 n ($1 \leq n \leq 5 \times 10^5$)

输出描述:

输出不超过 n 的完全数的个数

示例:

输入:

1000

输出:

3

2、美国节日

和中国的节日不同，美国的节假日通常是选择某个月的第几个星期几这种形式，因此每一年的放假日期都不相同。具体规则如下：

- * 1月1日：元旦
- * 1月的第三个星期一：马丁·路德·金纪念日
- * 2月的第三个星期一：总统节
- * 5月的最后一个星期一：阵亡将士纪念日
- * 7月4日：美国国庆
- * 9月的第一个星期一：劳动节
- * 11月的第四个星期四：感恩节
- * 12月25日：圣诞节

现在给出一个年份，请你帮忙生成当年节日的日期。

输入描述:

输入 m ，然后输入 m ($m \leq 100$)组数据，每组数据包含一个正整数 $year$ ($2000 \leq year \leq 9999$)。

输出描述:

对应每一组数据，以“YYYY-MM-DD”格式输出当年所有的节日日期，每个日期占一行。

每组数据之后输出一个空行作为分隔。

示例:

输入:

2
2014
2013

输出:

2014-01-01
2014-01-20
2014-02-17
2014-05-26
2014-07-04

2014-09-01
2014-11-27
2014-12-25

2013-01-01
2013-01-21
2013-02-18
2013-05-27
2013-07-04
2013-09-02
2013-11-28
2013-12-25