GESP C++四级样题卷

（满分：100 分　考试时间：90 分钟）

学校：_____　　姓名：_____

题目	一	二	三	总分
得分				

一、单选题（每题 2 分，共 30 分）

题号	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
答案	**D**	**D**	**D**	**D**	**A**	**C**	**C**	**A**	**B**	**B**	**A**	**B**	**B**	**B**	**C**

1. 在 C++中，指针变量的大小（单位：字节）是（ ）

A. 2

B. 4

C. 8

D. 与编译器有关

2. 以下哪个选项能正确定义一个二维数组（ ）

A. `int a[][];`

B. `char b[][4];`

C. `double c[3][];`

D. `bool d[3][4];`

3. 在 C++中，以下哪种方式**不能**用于向函数传递参数（ ）

A. 值传递

B. 引用传递

C. 指针传递

D. 模板传递

4. 以下关于C++函数的形参和实参的叙述，正确的是（ ）

A. 形参是实参的别名

B. 实参是形参的别名

C. 形参和实参是完全相同的

D. 形参用于函数声明，实参用于函数调用

5. 排序算法的稳定性是指（ ）

A. 相同元素在排序后的相对顺序保持不变

B. 排序算法的性能稳定

C. 排序算法对任意输入都有较好的效果

D. 排序算法容易实现

6. 如果有如下二维数组定义，则 **a[0][3]**的值为（ ）

```
int a[2][2] = {{0, 1}, {2, 3}};
```

A. 编译出错

B. 1

C. 3

D. 0

7. 以下哪个选项能正确访问二维数组 array 的元素（ ）

A. **array[1, 2]**

B. **array(1)(2)**

C. **array[1][2]**

D. **array{1}{2}**

8. 以下哪个选项是C++中正确的指针变量声明（ ）

A. **int *p;**

B. **int p*;**

C. `*int p;`

D. `int* p*;`

9. 在C++中，以下哪个关键字或符号用于声明引用（ ）

A. `pointer`

B. `&`

C. `*`

D. `reference`

10. 以下哪个递推关系式表示斐波那契数列（ ）

A. F(n) = F(n-1) + F(n-2) + F(n-3)

B. F(n) = F(n-1) + F(n-2)

C. F(n) = F(n-1) * F(n-2)

D. F(n) = F(n-1) / F(n-2)

11. 以下哪个函数声明在调用时可以传递二维数组的名字作为参数？

A. `void BubbleSort(int a[3][4]);`

B. `void BubbleSort(int a[][]);`

C. `void BubbleSort(int * a[]);`

D. `void BubbleSort(int ** a);`

12. 在C++中，以下哪个关键字用来捕获异常（ ）

A. `throw`

B. `catch`

C. `try`

D. `finally`

13. 在下列代码的横线处填写（ ），可以使得输出是“20 10”。

```
#include <iostream>
using namespace std;
void xchg(__________________) { // 在此处填入代码
    int t = x;
    x = y;
    y = t;
}
int main() {
    int a = 10, b = 20;
    xchg(a, b);
    cout << a << " " << b << endl;
    return 0;
}
```

A. `int x, int y`

B. `int & x, int & y`

C. `int a, int b`

D. `int & a, int & b`

14. 在下列代码的横线处填写（ ），可以使得输出是“21”。

```
#include <iostream>
using namespace std;
int main() {
    int a[5];
    a[0] = 1;
    for (int i = 1; i < 5; i++)
        a[i] = a[i - 1] * 2;
    int sum = 0;
    for (int i = 0; i < 5; _________) // 在此处填入代码
        sum += a[i];
```

```
    cout << sum << endl;
    return 0;
}
```

A. `i++`

B. `i += 2`

C. `i += 3`

D. `i |= 2`

15. 在下列代码的横线处填写（ ），完成对有 `n` 个 `int` 类型元素的数组 `array` 由小到大排序。

```
void BubbleSort(int array[], int n) {
    for (int i = n; i > 1; i--)
        for (_____________________) // 在此处填入代码
            if (array[j] > array[j + 1]) {
                int t = array[j];
                array[j] = array[j + 1];
                array[j + 1] = t;
            }
}
```

A. `int j = i - 2; j >= 0; j--`

B. `int j = i - 1; j >= 0; j--`

C. `int j = 0; j < i - 1; j++`

D. `int j = 0; j < i; j++`

二、判断题 (每题 2 分，共 20 分)

题号	1	2	3	4	5	6	7	8	9	10
答案	T	F	T	T	T	F	T	F	F	F

1. C++语言中的指针变量可以指向任何类型的数据。( )

2. 在 C++语言中，函数的参数默认以地址传递方式进行传递。( )

3. C++语言中的全局变量在整个程序的生命周期内都是有效的。( )

4. 递推算法通常有初始值。( )

5. 冒泡排序是一种稳定的排序算法。( )

6. C++语言中，如果异常发生，但没有处理异常的代码，则程序会由于一直等待处理而死机。( )

7. C++语言中的局部变量在函数调用结束后会被销毁。( )

8. `&`和`&&`都是 C++语言的运算符，`*`和`**`也都是。( )

9. 如果希望设计一个函数 `xchg`，实现交换两个 `int` 变量的值，则它的声明可以写为 `void xchg(int a, int b);`。( )

10. 已知数组 `a` 定义为 `int a[100];`，则赋值语句 `a['0'] = 3;`会导致编译错误。( )

三、编程题 (每题 25 分，共 50 分)

题号	1	2
答案		

1. 绝对素数

如果一个两位数是素数，且它的数字位置经过对换后仍为素数，则称为绝对素数，例如 13。给定两个正整数 A、B，请求出大于等于 A、小于等于 B 的所有绝对素数。

【输入格式】

输入 1 行，包含两个正整数 A 和 B。保证 10<A<B<100。

【输出格式】

若干行，每行一个绝对素数，从小到大输出。

【样例输入】

```
11 20
```

【样例输出】

```
11

13

17
```

【参考代码】

```
#include <iostream>
using namespace std;
bool is_prime(int num) {
    if (num < 2)
        return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return false;
    }
    return true;
}
int main() {
    int a = 0, b = 0;
    cin >> a >> b;
    for (int num = a; num <= b; num++) {
        if (is_prime(num)) {
            int reversed_num = (num % 10) * 10 + num / 10;
            if (is_prime(reversed_num))
                cout << num << endl;
        }
    }
    return 0;
}
```

2. 填幻方

在一个N×N的正方形网格中，每个格子分别填上从1到N×N的正整数，使得正方形中任一行、任一列及对角线的几个数之和都相等，则这种正方形图案就称为“幻方”（输出样例中展示了一个3×3的幻方）。我国古代称为“河图”、“洛书”，又叫“纵横图”。

幻方看似神奇，但当N为奇数时有很方便的填法：

1）一开始正方形中没有填任何数字。首先，在第一行的正中央填上1。

2）从上次填数字的位置向上移动一格，如果已经在第一行，则移到同一列的最后一行；再向右移动一格，如果已经在最右一列，则移动至同一行的第一列。如果移动后的位置没有填数字，则把上次填写的数字的下一个数字填到这个位置。

3）如果第2步填写失败，则从上次填数字的位置向下移动一格，如果已经在最下一行，则移到同一列的第一行。这个位置一定是空的（这可太神奇了!），把上次填写的数字的下一个数字填到这个位置。

4）重复2、3步骤，直到所有格子都被填满，幻方就完成了!

快来编写一个程序，按上述规则，制作一个N×N的幻方吧。

【输入格式】

输入为一个正奇数N，保证3≤N≤21。

【输出格式】

输出N行，每行N个空格分隔的正整数，内容为N×N的幻方。

【样例输入】

```
3
```

【样例输出】

```
8 1 6

3 5 7

4 9 2
```

【参考代码】

```
#include<bits/stdc++.h>
using namespace std;
int cube[21][21];
int main() {
    int n = 0;
    cin >> n;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            cube[i][j] = 0; // 清空正方形图表
    int x = 0, y = n / 2;
    cube[x][y] = 1; // 第 1 步，第一行正中填写 1
    for (int d = 2; d <= n * n; d++) {
        int nx = (x + n - 1) % n;
        int ny = (y + 1) % n; // 第 2 步，向右上移动一格
        if (cube[nx][ny] != 0) {
            nx = (x + 1) % n; // 第 3 步，如果第 2 步失败，向下移动一格
            ny = y;
        }
        cube[nx][ny] = d; // 填写下一个数字
        x = nx;
        y = ny;
    }
    for (int i = 0; i < n; i++) { // 输出幻方
        cout << cube[i][0];
        for (int j = 1; j < n; j++)
            cout << " " << cube[i][j];
        cout << endl;
    }
    return 0;
}
```