

GESP C++ 五级试卷（样题）

（满分：100 分 考试时间：180 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															

1、以下不属于计算机输出设备的有（ ）。

- A. 麦克风
- B. 音箱
- C. 打印机
- D. 显示器

2、小明想了一个 1 到 100 之间的整数。你可以做多次猜测，每次猜测之后，如果你没有猜中，小明会告诉你，你猜的数比他想的数大还是小。你希望你在运气最坏的情况下花费最少的次数猜中，请问你运气最坏的情况下会猜（ ）次？（包括最后猜中的那次）。

- A. 5
- B. 6
- C. 7
- D. 100

3、关于分治算法，下列说法错误的是（ ）。

- A. 分治算法的核心思想是分而治之，即把问题转化为多个规模更小的子问题求解。
- B. 分治算法可以不使用递归实现。
- C. 分治算法的时间复杂度是 $O(\log N)$ ，其中 N 表示问题的规模。
- D. 分治算法通常较容易在多核处理器上实现加速。

4、有关下面 C++ 代码说法错误的是（ ）。

```
1  #include <iostream>
2  using namespace std;
3
4  int factA(int n) {
5      if (n <= 1)
6          return 1;
7      int ret = 1;
8      for (int i = 2; i <= n; ++i)
9          ret *= i;
10     return ret;
11 }
12
13 int factB(int n) {
14     return n == 1 ? 1 : n * factB(n - 1);
15 }
16
17 int main() {
18     int n;
19     cin >> n;
20     cout << factA(n) << ' ' << factB(n) << endl;
21     return 0;
22 }
```

- A. factA() 采用循环方式求 n 的阶乘，factB() 采用递归方式求 n 的阶乘
- B. 程序执行时如果输入 5，能实现求其阶乘，程序输出结果为 120 120
- C. 任何递归程序都可以使用循环改写
- D. 程序执行时如果输入 100，不能正确求出 100 的阶乘

5、下面 C++ 代码意在实现字符串反序的功能。关于这段代码，以下说法正确的是（ ）

```

1  #include <iostream>
2  #include <cstring>
3  using namespace std;
4
5  void printSReverse(char* sIn, int len) {
6      if (len <= 1) {
7          cout << sIn[0];
8      }
9      else {
10         cout << sIn[0];
11         printSReverse(sIn + 1, len - 1);
12     }
13 }
14
15 int main() {
16     char sIn[100] = "Hello";
17     printSReverse(sIn, strlen(sIn));
18     return 0;
19 }

```

- A. 这段代码可以正确实现字符串反序的功能，其输出为`olleH`
- B. 这段代码不能正确实现字符串反序的功能，其输出为`Hello`
- C. 这段代码不能正确实现字符串反序的功能，其输出为`HHHHH`
- D. 这段代码不能正确实现字符串反序的功能，其输出为`ooooo`

6、阅读下面 C++实现的二分查找代码，下列说法中错误的是（ ）。

```

1  int binarySearch(int * arr, int l, int r, int x) {
2      if (r >= 1) {
3          int mid = 1 + (r - 1) / 2;
4          if (arr[mid] == x)
5              return mid;
6          else if (arr[mid] > x)
7              return binarySearch(arr, l, mid - 1, x);
8          else
9              return binarySearch(arr, mid + 1, r, x);
10     }
11     else
12         return -1;
13 }

```

- A. 上面代码实现的二分查找，最少只需查找一次即可得到结果
- B. 如果调用该函数在列表{2, 3, 4, 10, 12}中查找元素 0，则它实际被调用 3

次

C. 如果调用该函数在列表 {2, 3, 4, 10, 12} 中查找元素 3, 则它实际被调用 3 次

D. 如果调用该函数在列表 {2, 3, 4, 10, 12} 中查找元素 10, 则它实际被调用 3 次

7、使用如下代码片段定义四个字符串（假设头文件已正确定义），以下说法错误的是（ ）。

```
string str1 = "abc";
string str2 = str1;
char str3[] = "abc";
char * str4 = str3;
```

A. 对于四个字符串，都可以使用 `std::cout` 输出其中的内容（例如，`cout << str3;`）

B. `str3` 只占用 4 字节内存，但 `str1` 却要占用更多内存

C. 由于 `str2` 由 `str1` 直接赋值得到，因此二者指向同一块内存，即修改 `str1` 的内容后 `str2` 的内容也会随之改变

D. 由于 `str4` 由 `str3` 直接赋值得到，因此二者指向同一块内存，即修改 `str3` 的内容后 `str4` 的内容也会随之改变

8、有关下面 C++ 代码正确的是（ ）。

```
1  #include <iostream>
2  using namespace std;
3
4  void f1() {
5      cout << "f1()" << endl;
6  }
7
8  void f1(int x) {
9      cout << "f1(" << x << ")" << endl;
10 }
11
12 int main() {
13     f1();
14     f1(0);
15     f1('0');
16     return 0;
17 }
```

- A. 该程序不能正常运行，因为 f1 函数被重复定义。
- B. 该程序可以正常运行，输出结果共 3 行，依次为 f1()、f1()、f1()
- C. 该程序可以正常运行，输出结果共 3 行，依次为 f1()、f1(0)、f1(0)
- D. 该程序可以正常运行，输出结果共 3 行，依次为 f1()、f1(0)、f1(48)

9、关于 C++ 程序的异常处理，以下选项中描述错误的是（ ）。

- A. 编程语言中的异常和错误是不同的概念
- B. 异常一旦发生，程序便一定不能继续执行
- C. 通过 try、catch 等保留字提供异常处理功能
- D. 程序使用 throw 在任何地方抛出各种异常

10、下面代码执行后的输出是（ ）。

```
1  #include <iostream>
2  using namespace std;
3
4  int fibonacci(int N) {
5      cout << N << ",";
6      if (N == 1 || N == 2) {
7          return 1;
8      } else {
9          return fibonacci(N - 1) + fibonacci(N - 2);
10     }
11 }
12
13 int main() {
14     cout << fibonacci(5) << endl;
15     return 0;
16 }
```

- A. 5, 4, 3, 2, 1, 2, 1, 5
- B. 5, 4, 3, 2, 1, 2, 3, 2, 1, 5
- C. 5, 4, 4, 3, 2, 1, 3, 2, 1, 5
- D. 5, 4, 3, 2, 1, 3, 2, 1, 5

11、下列代码中，函数 f 的作用是（ ）。

```
void f(int a, int b) {
    return b == 0 ? a : f(b, a % b);
}
```

- A. 求 a 和 b 的最大公共质因子
- B. 求 a 和 b 的最小公共质因子
- C. 求 a 和 b 的最大公约数
- D. 求 a 和 b 的最小公倍数

12、下面 C++代码用于排序，下列说法中错误的是（ ）。

```
1  void sortA(int * arr, int n) {
2      for (int i = 0; i < n; ++i)
3          for (int j = 0; j < n - i - 1; ++j)
4              if (arr[j] > arr[j + 1]) {
5                  int tmp = arr[j];
6                  arr[j] = arr[j + 1];
7                  arr[j + 1] = tmp;
8              }
9  }
10
11 void sortB(int * arr, int start, int end) {
12     if (start >= end)
13         return;
14
15     int middle = (start + end) / 2;
16     sortB(arr, start, middle);
17     sortB(arr, middle + 1, end);
18
19     int leftSize = middle - start + 1;
20     int rightSize = end - middle;
21
22     int * left = new int[leftSize];
23     int * right = new int[rightSize];
24
25     for (int i = 0; i < leftSize; i++)
26         left[i] = arr[start + i];
27     for (int j = 0; j < rightSize; j++)
28         right[j] = arr[middle + 1 + j];
29
30     int i = 0;
31     int j = 0;
32     int k = start;
33     while (i < leftSize && j < rightSize) {
34         if (left[i] <= right[j]) {
35             arr[k] = left[i];
36             i++;
37         } else {
38             arr[k] = right[j];
39             j++;
40         }
41         k++;
42     }
43     while (i < leftSize) {
44         arr[k] = left[i];
```

```

45         i++;
46         k++;
47     }
48     while (j < rightSize) {
49         arr[k] = right[j];
50         j++;
51         k++;
52     }
53
54     delete[] left;
55     delete[] right;
56 }

```

- A. 两种排序算法的时间复杂度不同。
- B. 两种排序算法的空间复杂度一致。
- C. sortA 的时间复杂度在最好和最坏情况下都是 $O(N^2)$ 。
- D. sortB 的平均时间复杂度、最好情况的时间复杂度都是 $O(\log N)$ ，最坏情况的时间复杂度是 $O(N^2)$ 。

13、上一题中的`sortB`函数，明显体现出的算法思想和编程方法包括（ ）。

- A. 递归
- B. 分治
- C. A、B 都正确
- D. A、B 都不正确

14、下列哪个算法并没有体现分治思想？（ ）。

- A. 二分查找
- B. 埃氏筛法。
- C. 归并排序。
- D. 快速排序。

15、下列关于链表说法，正确的是（ ）。

- A. 不能用数组来实现链表。
- B. 在链表头部插入元素的时间复杂度是 $O(1)$ 。
- C. 循环链表使得任意一个结点都可以很方便地访问其前驱与后继。
- D. 从双向链表的任意一个节点出发，并不一定能够访问到所有其他节点。