

**C++** 七级

2024 年 09 ⽉

**1** 单选题 (每题 **2** 分,共 **30** 分)

**第 1 题** 已知⼩写字母 b 的ASCII码为98 ，下列C++代码的输出结果是（ ）。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #include <iostream> using namespace std;  int main() {  char a = 'b '; a++;  cout << a;  return 0; } |

口 **A.** b 口 **B.** c  **C.** 98 口 **D.** 99

**第 2 题** 已知 a 为 int 类型变量 ，下列表达式不符合语法的是（ ）。 口 **A.** &a + 3

口 **B.** +a & 3

 **C.** a - - 4

 **D.** a++3

**第 3 题** 下列关于C++语⾔中指针的叙述 ，不正确的是（ ）。 口 **A.** 指针变量中存储的是内存地址。

口 **B.** 指针变量指向的内存地址不一定能够合法访问。

 **C.** 结构类型中的指针成员不能指向该结构类型。

 **D.** 定义指针变量时必须指定其指向的类型。 **第 4 题** 下列关于C++类的说法 ，错误的是( )。

口 **A.** 将C++类对象通过值传递给函数参数时 ，会⾃动调⽤复制构造函数。

口 **B.** 将一个类的对象赋值给该类的另一个对象时 ，不会⾃动调⽤构造函数。

 **C.** 定义C++类对象时 ，一定会调⽤默认构造函数。

 **D.** 构造派⽣类的对象时 ，一定会调⽤基类的构造函数。

**第 5 题** 某⼆叉树T的先序遍历序列为： {AB D C E G H F}， 中序遍历序列为： {D BAH G E C F} ，则下列说法中正 确的是( )。

口 **A.** T的⾼为5

口 **B.** T有4个叶节点  **C.** T是平衡树

口 **D.** 以上说法都不对

**第 6 题** 一棵完全⼆叉树有431个结点 ，则叶结点有多少个？ ( ) 口 **A.** 176

口 **B.** 215

口 **C.** 216

口 **D.** 255

**第 7 题** 下列关于树的说法 ，错误的是（ ）。

口 **A.** ⼆叉树的中序遍历与其深度优先遍历总是相同的。

口 **B.** 所有树都可以构造一颗⼆叉树与之一一对应。

口 **C.** 如果树的一个叶结点有两个不同的祖先结点 ，那么其中一个一定是另一个的祖先结点。

口 **D.** 树的结点不能有两个⽗结点。

**第 8 题** 一个简单⽆向图有10个结点、30条边 。再增加多少条边可以成为完全图 。 ( ) 口 **A.** 10

口 **B.** 15

 **C.** 51

口 **D.** 60

**第 9 题** 以下哪个⽅案可以合理解决或缓解哈希表冲突（ ）。

口 **A.** 丢弃发⽣冲突的新元素。

口 **B.** ⽤新元素覆盖发⽣冲突的元素。

 **C.** ⽤新元素覆盖在冲突位置的下一个位置。

 **D.** 将新元素放置在冲突位置之后的第一个空位。

**第 10 题** 一个迷宫， 已知从起点不经过重复结点到达终点的路径有且仅有一条 ，则下⾯说法错误的是（ ）。

口 **A.** 可以使⽤深度优先搜索找到这条路径。

口 **B.** 可以使⽤⼴度优先搜索找到这条路径。

 **C.** 该迷宫内与起点连通的结点 ，一定也与终点连通。

 **D.** 该迷宫内与起点连通的结点及它们之间的路径可以抽象为⽆向⽆环图。 **第 11 题** 下⾯程序的输出为（ ）。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | #include <iostream> #include <cmath>  using namespace std; int main() {  cout << (int)log(8) << endl; return 0;  } |

口 **A.** 2

口 **B.** 3

口 **C.** 8

 **D.** ⽆法通过编译。

**第 12 题** 下⾯程序的输出为（ ）。

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

#include <iostream> #define N 10

using namespace std;

int path[N][N]; int main() {

for (int i = 1; i < N; i++)

path[i][0] = i;

for (int j = 1; j < N; j++)

path[0][j] = j;

for (int i = 1; i < N; i++)

for (int j = 1; j < N; j++)

path[i][j] = path[i - 1][j] + path[i][j - 1]; cout << path[8][4] << endl;

return 0; }

口 **A.** 84

口 **B.** 495

口 **C.** 1012

 **D.** 结果是随机的。

**第 13 题** 上题中程序的时间复杂度为（ ）。 口 **A.** O(1)

口 **B.** O(N)

口 **C.** O(NlogN)

口 **D.** o(N2)

**第 14 题** 下⾯ fib 函数的时间复杂度为( )。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | int fib\_rcd[MAX\_N];  int fib(int n) {  if (n <= 1)  return 1;  if (fib\_rcd[n] > 0)  return fib\_rcd[n];  return fib(n - 1) + fib(n - 2); } |

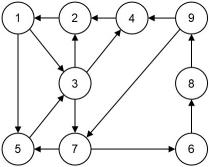
口 **A.** O(n)

**B.** (φn) , 

口 **C.** O(2")

 **D.** ⽆法正常结束。

**第 15 题** 下列选项中 ，哪个可能是下图的⼴度优先遍历序列（ ）。



口 **A.** 1, 3, 5, 7, 4, 2, 6, 8, 9 口 **B.** 9, 4, 2, 1, 3, 7, 5, 6, 8

口 **C.** 1, 3, 5, 7, 6, 8, 9, 4, 2 口 **D.** 9, 4, 7, 2, 1, 3, 5, 6, 8

**2** 判断题 (每题 **2** 分,共 **20** 分)

**第 1 题** 表达式 'a ' << 1 的结果为 'a ' 。

**第 2 题** 在C++语⾔中 ，函数可以定义在另一个函数定义之内。

**第 3 题** 选择排序一般是不稳定的。

**第 4 题** 埃⽒筛法和欧拉筛法都是使⽤筛法思想⽣成素数表的算法 ，欧拉筛法的时间复杂度更低。

**第 5 题** 使⽤ math.h 或 cmath 头⽂件中的正弦函数 ，表达式 sin(30) 的结果类型为 double 、值约为 0.5 。

**第 6 题** 一颗N层的完全⼆叉树 ，一定有2N  1个结点。

**第 7 题** 一个图 ，不管是否连通 ，都可以使⽤深度优先搜索算法进⾏遍历。

**第 8 题** 某个哈希表键值 x 为整数， H(x) = x % p 是常⽤的哈希函数之一 ，要求 p 选择素数是因为这样不会产⽣ 冲突 。( )

**第 9 题** 使⽤单链表实现队列时 ，链表头结点作为队⾸⽐链表头结点作为队尾更便于操作。

**第 10 题** 一个图中 ，每个结点表达一个⼈ ，连接两个结点的边表达两个结点对应的⼈相互认识 ，则这个图可以⽤来 表达社交⽹络。