

一、单项选择题

1-5CDDCC

6-10BACDB

11-15DDABC

1. [答案]C[分析]未参加第一轮认证、不符合本省参加第二轮认证晋级规则或未在系统中报名等情况者，均不能参加第二轮认证
2. [答案]D[分析] $[10110110]_{\text{补}}=11001010=(-74)_{10}$
3. [答案]D[分析]1978-2019 年间的闰年是 1980 年、1984 年、1988 年、1992 年、1996 年、2000 年、2004 年、2008 年、2012 年、2016 年。
4. [答案]C[分析]设该树中所有结点的度为 n ，因为在树的结点中，除了根结点以外，其余结点都有一个分支进入，所以 $n=x+1, n-1$ ，
5. [答案]C[分析]整数在计算机中的二进制表示是补码。
6. [答案]B[分析]逆序对(3,1)(20,4)(20,6)(20,1)(4,1)(6,1)，个数即最少交换次数。
7. [答案]A
8. [答案]C[分析]A 是根结点，所以只有 C 答案正确
9. [答案]D[分析]无向完全图的边数 $m=(n-1)n/2$, n 是顶点个数。所以 $(9-1) \times 9/2=36$
10. [答案]B[分析]栈的特点为先进后出
11. [答案]D[分析]树的边数=点数-1= $n-1$ ，所以要删掉 $m-(n-1)=m-n+1$ 条边。
12. [答案]D[分析]子串的定义是原字符串中连续的一段字符组成的字符串，不同的子串是当且仅当两个子串长度不一样，或者长度一样但有至少任意一位不一样时成立。那么将“abcb”先看成 5 个不同字符，不同字符的子串个数是 $n(n+1)/2+1=15$ ，不包含空串。子串只有一个字符时，多了一个“a”，一个“b”；子串只有 2 个字符时多了一个“ab”。 $15-3=12$ 。
13. [答案]A[分析]13 的二进制数为 1101，0.375 的二进制数为 0.011。
14. [答案]B[分析] $6 \times 2^{10} \times 2^{10}/2^9=12 \times 2^{10} \approx 12000$
15. [答案]C[分析] $C(3,2) \times A(365,2)/(365^3)$

二、阅读程序（程序输入不超过数组或字符串定义的范围；判断正确填√，错误填×；除特殊说明外，判断题 1.5 分，选择题 3 分，共计 40 分）

1.

错	错	错	
A	A	D	

判断题

- 1) 输出 9 3 9。
- 2) 12300400000 超过 int 范围了。
- 3) cin cout 的头文件只能是 iostream。

选择题

- 4) 若输入 3 6 9，输出 6 3 6。
- 5) 若输入 3 6 9，输出 6 3 6。
- 6) 若输入 3 6 9，输出 6 3 3。

2.

错	对	对	错
D	C		

判断题

- 1) 01 背包问题，双重循环里循环变量 j 的枚举顺序必须是 m 到 $w[i]$ 倒序。
- 2) 双重循环中变量 i 的枚举顺序改为从 n 到 1，输出结果一定不变。
- 3) 若输入数据中， $1 \leq n, m, w[i], d[i] \leq 30000$ ，则所求答案一定在 int 范围内，没有

溢出。

4) 若输入数据中, $1 \leq n, m, w[i] \leq 30000$, $1 \leq d[i] \leq 10\ 0000\ 0000$, 则所求答案有可能溢出 `int` 范围。

选择题

5) 1 4 3 12 2 7 组合放进背包, 最大价值为 23。

6) 01 背包算法时间复杂度为 $O(n \cdot m)$

3.

对	对	错	
C	C	C	

判断题

1) 选择排序算法实现了对一个长度为 n 的序列进行排序。

2) 代码没有使用 `scanf` 和 `printf`, 去掉头文件 “`include<stdio>`” 后程序是可以正常编译运行的。

3) 去掉 “`using namespace std;`”, `cin cout` 前面就需要加 `std`, 才能编译成功。

选择题

4) 排序算法的基本思路是为每一个位置选择当前最小的元素。

5) 选择排序算法的时间复杂度为 $O(n)$ 。

6) 32154 \rightarrow 12354 执行两次, 12354 \rightarrow 12345 执行一次, 共 3 次。

三. 完善程序 (单选题, 每小题 3 分, 共计 30 份)

1.

D	C	C	A	B
---	---	---	---	---

1) `st` 表示起点, `int` 类型。

2) 队列的头指针 `hd` \leq 队列的尾指针, 队列中有元素, 继续循环。JubtR1

3) `dis[v] != -1` 代表 `v` 已经被访问过, 跳过后面的执行语句。

4) `G[x]` 存储以 `x` 为起点的边, `G[y]` 存储以 `y` 为起点的边, `G[y].push_back(x)` 就是将边 `(y, x)` 加入 `G[y]` 中。

5) `for(int i=1; i<=n; i++)` 输出起点到各个点的距离。

2.

D	C	A	B	C
---	---	---	---	---

1) 入度为 0 的顶点入队。

2) `i` 从 0 起始, 所以要填入 `i < G[u].size()`。

3) 入度为 0 的顶点入队。迁姚樊赛对验高聂

4) 有向图存储边 `G[x].push_back(y)`, `x` 是起点, `y` 是终点。

5) `y` 的入度+1。