NOIP 2019模拟考试

普及组C++语言试题

竞赛时间：2019年10月13日 9:00~10:00

姓名：

一、单项选择题（共20题，每题1.5分，共计30分； 每题有且仅有一个正确选项）

1. 如果256种颜色用二进制编码来表示，至少需要（C）位。

   A. 6 B. 7 C. 8 D. 9

2. 下列四个不同进制的数中，与其他三项数值上不相等的是（D）。

   A. 16进制：269

   B. 10进制：617

   C. 8进制：1151

   D. 2进制：1001101011

3. 1MB等于（D）。

   A. 1000 字节　　B. 1024字节

   C. 1000 X 1000字节　　D. 1024 X 1024字节

4. 在8位二进制补码中，10101011表示的数是十进制下的（B 负数的补码=反码+1；负数的反码=除了符号位不变，原码的其他位取反）

   A. 43　　B. -85　　C. -43　　D. -84

5. 甲，乙，丙三位同学选修课程，从4门课程中，甲选修2门，乙和丙各选修3门，则不同的选修方案共有（C）

   A. 36　　B. 48　　C. 96　　D. 192

6. 表达式 a*(b+c)*d的后缀形式是（B）

   A. abcd*+*

   B. abc+*d*

   C. a*bc+*d

   D. b+c*a*d

7. 十进制小数13.375对应的二进制数是（A）

   A. 1101.011　　B. 10111.011　　C. 1101.101　　D. 1010.01

8. （C）就是把一个复杂的问题分成两个或者更多的相同或相似的子问题，再把子问题分成更小的子问题……直到最后的子问题可以简单的直接求解。而原问题的解就是子问题解的并。

   A. 动态规划　B. 贪心　C. 分治　D. 搜索

9. 以下排序算法中，不需要进行关键字比较操作的算法是（A）。

   A. 基数排序　　B. 冒泡排序　　C. 堆排序　　D. 插入排序

10. 由四个没有区别的点构成的简单无向图的个数是（A）。

    A. 6　B. 7　C. 8　D. 9

11. 链表不具有的特点是（B）

    A. 不必事先估计存储空间

    B. 可随机访问任一元素

    C. 插入删除不需要移动元素

    D. 所需空间与线性表长度成正比

12. 要求以下程序的功能时计算： s = 1 + 1/2 + 1/3 + ... + 1/10。

```
#include <iostream>
using namespace std;

int main() {
    int n;
    float s;
    s = 1.0;

    for (n = 10; n > 1; n--)
        s = s + 1 / n;
    cout << s << endl;
    return 0;
}
```

    程序运行后输出结果错误，导致错误结果的程序行是（ C）。

    A. s = 1.0;
    B. for (n = 10; n > 1; n--)
    C. s = s + 1 / n;
    D. cout << s << endl;

13. 设变量x为float型且已赋值，则以下语句中能将x中的数值保留到小数点后两位，并将第三位四舍五入的是（ C）。

    A. x = (x * 100) + 0.5 / 100.0;

    B. x = (x * 100 + 0.5) / 100.0;

    C. x = (int) (x * 100 + 0.5) / 100.0;

    D. x = (x / 100 + 0.5) * 100.0;

14. 有以下程序

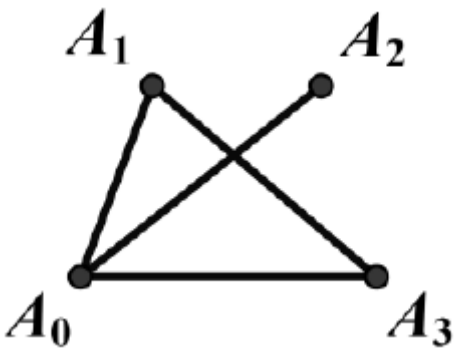
```
#include <iostream>
using namespace std;

int main() {
    int s, a, n;
    s = 0;
    a = 1;
    cin >> n;

    do {
        s += 1;
        a -= 2;
    } while (a != n);
```

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

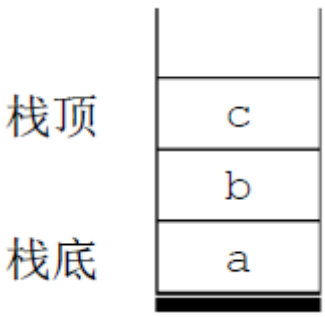
若要使程序的输出值为2，则应该从键盘给n输入的值是（ B ）。

A. -1　　B. -3　　C. -5　　D. 0

15. 二叉树的中序序列是 d g b a e c h f, 后序序列是 g d b e h f c a，则先序是*B*
    A. a b c d f g h e
    B. a b d g c e f h
    C. a c b g d h e f
    D. a c e f h b g d
16. 有向图中每个顶点的度等于该顶点的（ C）。
    A. 入度
    B. 出度
    C. 入度与出度之和
    D. 入度与出度之差
17. 已知一棵二叉树有10个节点，则其中至多有（ A ）个节点有2个子节点。
    A. 4　　B. 5　　C. 6　　D. 7
18. 以A0作为起点，对下面的无向图进行深度优先遍历时，遍历顺序不可能是（ A）。



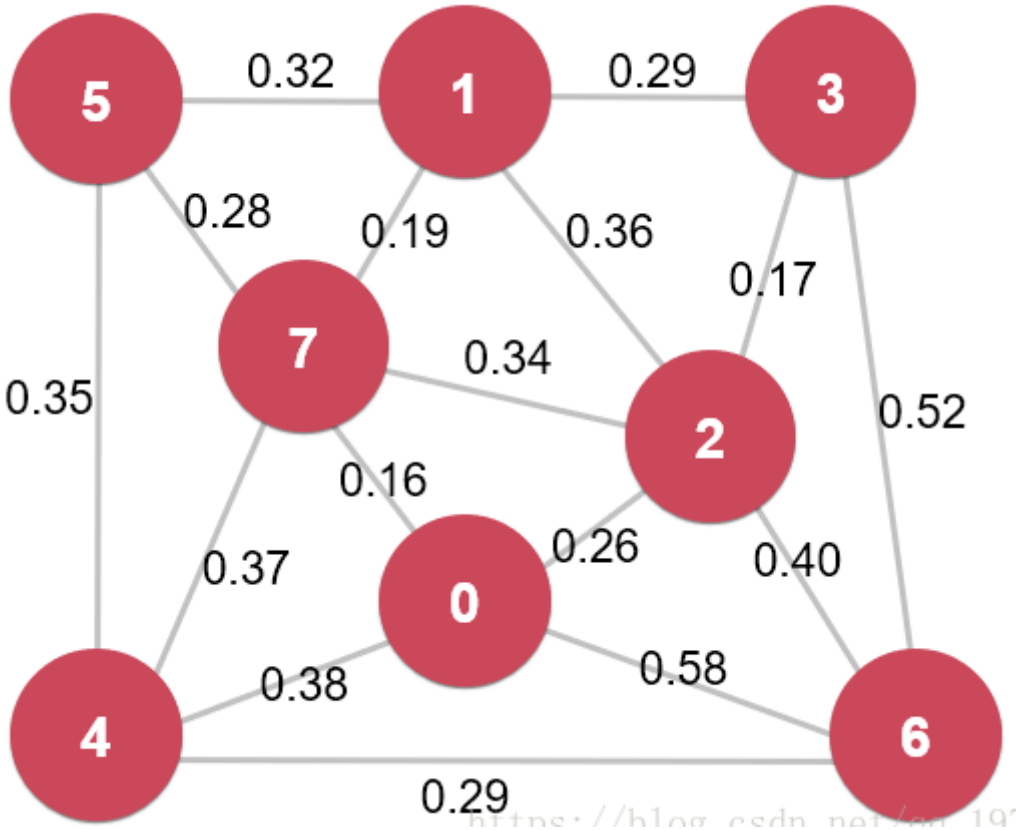
    A.　A0, A1, A2, A3
    B.　A0, A1, A3, A2
    C.　A0, A2, A1, A3
    D.　A0, A3, A1, A2
19. （ A）的平均时间复杂度为O(n log n)，其中n是待排序的元素个数。
    A. 快速排序
    B. 插入排序
    C. 冒泡排序
    D. 基数排序
20. 如果一个栈初始时为空，且当前栈中的元素从栈底到栈顶依次为a, b, c（如右图所示），另有元素d已经出栈，则可能的入栈顺序是（ D ）。



A. a, d, c, b
B. b, a, c, d
C. a, c, b, d
D. d, a, b, c

二、问题求解（共2题，每题5分，共计10分）

1. 如图所示，图中每条边上的数字表示该边的长度，则从5到6的最短距离是__0.64___。



2. 甲乙丙丁私人在考虑周末要不要外出郊游。已知：

a. 如果周末下雨，并且乙不去，则甲一定不去。

b. 如果乙去，则丁一定去。

c. 如果丙去，则丁一定不去。

d. 如果丁不去，而且甲不去，则丙一定不去。

如果周末丙去了，则甲____（**去了**/没去），乙____（去了/**没去**），丁____（去了/**没去**），周末 ____（下雨/**没下雨**）

三、阅读程序写结果（共4题，每题8分，共计32分）

1.

```
#include <iostream>
using namespace std;
int main()
    int a, b, c, d, ans;
    cin >> a >> b >> c
    d = a - b;
    a = d + c;
    ans = a * b;
    cout << "Ans = " << ans << endl;
    return 0;
}
输入：2 3 4
输出：__9_______
```

2.

```
#include <iostream>
using namespace std;
int fun(int n)
    if (n == 1)
        return 1;
    if (n == 2)
        return 2;
    return fun(n - 2) - fun(n - 1);
}

int main()
    int n;
    cin >> n;
    cout << fun(n) << endl;
    return 0;
}
输入：7
输出：___-11_____

解析：用一个数组来保存结果，数组的第一个元素的值就是fun(1)，第二个元素的值就是fun(2)，以此类推，得到这个数组如下：
| 1 | 2 | -1 | 3 | -4 | 7 | -11 |
```

3.

```
#include iostream
#include <string>
using namespace std;
int main()
{
    string st;
    int i, len;
    getline(cin, st)
    len = st.size();
    for ( i = 0; i < len; i++) {
        if (st[i] >= 'a' && st[i] <= 'z')
            st[i] = st[i] - 'a' + 'A';
    }
```

```
    cout << st << endl
    return 0;
}
输入：Hello, my name is Lostmonkey.
输出：_HELLO, MY NAME IS LOSTMONKEY._________
```

4.

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, u, i, num;
    cin>>a>>b>>u;
    num = 0;
    for (i = a; i <= b; i++)
        if ((i % u) == 0)
            num++;
    cout<<num<<endl;
    return 0;
}
输入：1 100 15
输出：_____6____
```

四、完善程序（共2题，每题14分，共计28分）

1. （数字删除）下面程序的功能是将字符串中的数字字符删除后输出。请填空。（每空3分，共12分）

```
#include <iostream>
using namespace std;
int delnum(char *s) {
    int i, j;
    j = 0;
    for (i = 0; s[i] != '\0'; i++)
        if (s[i] < '0' (1) s[i] > '9') {
            s[j] = s[i];
            (2) ;
        }
    return (3) ;
}

const int SIZE = 30;

int main() {
    char s[SIZE];
    int len, i;
    cin.getline(s, sizeof(s));
    len = delnum(s);
    for (i = 0; i < len; i++)
        cout << (4) ;
    cout << endl;
    return 0;
}

(1) &&
```

```
(2) j++
(3) j
(4) s[i]
```

2. （序列重排）全局数组变量a定义如下：

```
const int SIZE = 100;
int a[SIZE], n;
```

它记录着一个长度为n的序列a[1], a[2], ..., a[n]。
现在需要一个函数，以整数p (1 ≤ p ≤ n)为参数，实现如下功能：将序列a的前p个数与后n－p个数对调，且不改变这p个数（或n－p个数）之间的相对位置。例如，长度为5的序列1, 2, 3, 4, 5，当p = 2时重排结果为3, 4, 5, 1, 2。
有一种朴素的算法可以实现这一需求，其时间复杂度为O(n)、空间复杂度为O(n)：

```
void swap1(int p)
{
    int i, j, b[SIZE];
    for (i = 1; i <= p; i++)
        b[ (1) ] = a[i]; //（3分）

    for (i = p + 1; i <= n; i++)
        b[i - p] = (2) ; //（3分）

    for (i = 1; i <= (3) ; i++) //（2分）
        a[i] = b[i];
}

(1) n - p + i
(2) a[i]
(3) n
```

我们也可以用时间换空间，使用时间复杂度为O(n2)、空间复杂度为O(1)的算法：

```
void swap2(int p)
{
    int i, j, temp;
    for (i = p + 1; i <= n; i++) {
        temp = a[i];
        for (j = i; j >= (4) ; j--) //（3分）
            a[j] = a[j - 1];
     (5) = temp; //（3分）
    }
}

(4) 2
(5) a[j]
```