

CSP-J 模拟试卷 11 (2018)

一、单项选择题(共 15 题, 每题 2 分, 共计 30 分;每题有且仅有一个正确选项)

1. 中国计算机学会于()年创办全国青少年计算机程序设计竞赛。
A. 1983 B. 1984 C. 1985 D. 1986
2. 下列四个不同进制的数中, 与其它三项数值上不相等的是()。
A. $(269)_{16}$ B. $(617)_{10}$ C. $(1151)_8$ D. $(1001101011)_2$
3. 1MB 等于()。
A. 1000 字节 B. 1024 字节 C. 1000×1000 字节 D. 1024×1024 字节
4. 广域网的英文缩写是()。
A. LAN B. WAN C. MAN D. LNA
5. 设某算法的时间复杂度函数的递推方程是 $T(n) = T(n-1) + n$ (n 为正整数) 及 $T(0) = 1$, 则该算法的时间复杂度为()。
A. $O(\log n)$ B. $O(n \log n)$ C. $O(n)$ D. $O(n^2)$
6. 如果开始时计算机处于小写输入状态, 现在有一只小老鼠反复按照 CapsLock、字母键 A、字母键 S、字母键 D、字母键 F 的顺序循环按键, 即 CapsLock、A、S、D、F、CapsLock、A、S、D、F、.... 屏幕上输出的第 81 个字符是字母()。
A. A B. S C. D D. a
7. 根节点深度为 0, 一棵深度为 h 的满 k ($k > 1$) 叉树, 即除最后一层无任何子节点外, 每一层上的所有结点都有 k 个子结点的树, 共有()个结点。
A. $(k^{h+1}-1)/(k-1)$ B. k^{h+1} C. k^h D. $(k^{h+1})/(k-1)$
8. 以下排序算法中, 不需要进行关键字比较操作的算法是()。
A. 基数排序 B. 冒泡排序 C. 堆排序 D. 直接插入排序
9. 给定一个含 N 个不相同数字的数组, 在最坏情况下, 找出其中最大或最小的数, 至少需要 $N-1$ 次比较操作。则最坏情况下, 在该数组中同时找最大与最小的数至少需要()次比较操作。($\lceil \cdot \rceil$ 表示向上取整, $\lfloor \cdot \rfloor$ 表示向下取整)
A. $\lceil 3N/2 \rceil - 2$ B. $\lfloor 3N/2 \rfloor - 2$ C. $2N-2$ D. $2N-4$
10. 下面的故事与()算法有着异曲同工之妙。
从前有座山, 山里有座庙, 庙里有个老和尚在给小和尚讲故事: “从前有座山, 山里有座庙, 庙里有个老和尚在给小和尚讲故事: ‘从前有座山, 山里有座庙, 庙里有个老和尚给小和尚讲故事....’”
A. 枚举 B. 递归 C. 贪心 D. 分治
11. 由四个没有区别的点构成的简单无向连通图的个数是()。
A. 6 B. 7 C. 8 D. 9
12. 设含有 10 个元素的集合的全部子集数为 s , 其中由 7 个元素组成的子集数为 T , 则 T/s 的值为()。
A. $5/32$ B. $15/128$ C. $1/8$ D. $21/128$.
13. 10000 以内, 与 10000 互质的正整数有()个。
A. 2000 B. 4000 C. 6000 D. 8000
14. 为了统计一个非负整数的二进制形式中 1 的个数, 代码如下:

```
int CountBit(int x){
    int ret = 0;
    while (x)
```

```

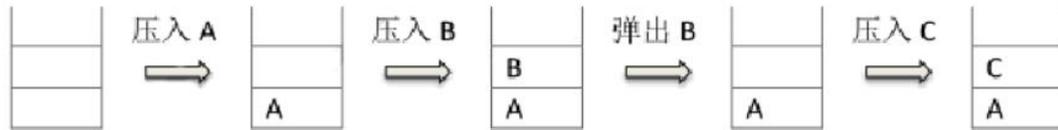
    {
        ret++;
    }
    return ret;
}

```

则空格内要填入的语句是()。

- A. $x \gg= 1$ B. $x \&= x - 1$ C. $x |= x > 1$ D. $x \ll = 1$

15. 下图中所使用的数据结构是()



- A. 哈希表 B. 栈 C. 队列 D. 二叉树

三、阅读程序写结果

1. [6506]

```

#include <iostream>
using namespace std;
03 const int c=2009;
int main()
{
    int n, p, s, i, j, t;
    cin >> n >> p;
08 s=0;t=1;
    for(i=1;i<=n;i++)
    {
11     t=t*p%c;
12     for(j=1;j<=i;j++)
13         s=(s+t)%c;
    }
15 cout << s << endl;
    return 0;
}

```

判断题

判断题

- (1) 将 12 13 行改为 $s=(s+1LL*t*i)\% c;$, 程序输出不会改变。 ()
- (2) 将 15 行改为 $\text{printf}(\text{"\%d\n"}, s);$ 程序输出不会改变。 ()
- (3) 将 08 行的 $s=0;$ 去掉, 程序输出不会改变。 ()
- (4) 将 03 行 const 去掉, 程序输出不会发生变化。 ()

● 选择题

- (5) 输入为 “11 2” 输出为()。
- A. 782 B. 762 C. 802 D. 114

- (6) 该算法的时间复杂度为()。
- A. $O(1)$ B. $O(n)$ C. $O(n^2)$ D. $O(n\log n)$

2.

```
#include <iostream>
using namespace std;
int n, d[100];
bool v[100];
int main() {
    cin>>n;
    for(int i=0; i<n; ++i) {
        scanf("%d", d[i]);
        v[i] = false;
    }
    int cnt = 0;
    for(int i=0; i<n; ++i) {
        if (!v[i]) {
            for(int j=i; !v[j]; j=d[j]) {
                v[j] = true;
            }
            ++cnt;
        }
    }
    cout<<cnt;
    return 0;
}
```

●判断题

- (1) 将第 7 行的 `d+i` 换成 `&d[i]`, 程序运行不受影响。 ()
- (2) 第 12 行的 `!v[i]` 与 `v[i]==false` 语句意思一致。 ()
- (3) 程序的输出结果 `cnt` 至少等于 1。 ()
- (4) 若输入的数组 `d` 中有重复的数字, 则程序会进入死循环。 ()

●选择题

- (5) 若输入数字为 5 1 2 3 4 5, 则输出为()。
- A. 0 B. 1 C. 2 D. 5
- (6) 若输入数字为 10 7 1 4 3 2 5 9 8 0 6, 则输出为()。
- A. 3 B. 6 C. 7 D. 8

3.

```
#include <iostream>
using namespace std;
string s;
long long magic(int l, int r)
{
    long long ans = 0;
```

```

    for (int i = 1; i <= r; ++i)
    {
        ans = ans * 4 + s[i] - 'a' + 1;
    }
    return ans;
}
int main()
{
    cin >> s;
    int len = s.length();
    int ans = 0;
    for (int l1 = 0; l1 < len; ++l1)
    {
        for (int r1 = l1; r1 < len; ++r1)
        {
            bool bo = true;
            for (int l2 = 0; l2 < len; ++l2)
            {
                for (int r2 = l2; r2 < len; ++r2)
                {
                    if (magic(l1, r1) == magic(l2, r2) && (l1 != l2 || r1 != r2))
                    {
                        bo = false;
                    }
                }
            }
            if (bo)
            {
                ans += 1;
            }
        }
    }
    cout << ans << endl;
    return 0;
}

```

判断题

- (1) 输出一定不为 0。 ()
- (2) 将 15, 16 行与 18, 19 行对调, 答案不变。 ()
- (3) 输入字符串最多句含 5 种字符串, 若更多则容易互相冲突。 ()
- (4) 调用第 12 行需要 iostream 库头。 ()

选择题

- (5) 输入 ahacaha. 则输出的结果是()。
- A. 16 B. 17 C. 18 D. 19
- (6) 输入 abcbbcc, 则输出的结果是()。

A. 16 B. 20 C. 18 D. 19

四、完善程序

1. (最大公约数之和) 下列程序想要求解整数 n 的所有约数两两之间最大公约数的和对 10007 求余后的值，试补全程序。

举例来说，4 的所有约数是 1, 2, 4。1 和 2 的最大公约数为 1；2 和 4 的最大公约数为 2；1 和 4 的最大公约数为 1。于是答案为 $1+2+1=4$ 。

要求 `getDivisor` 函数的复杂度为 $O(Vn)$ ，`gcd` 函数的复杂度为 $O(\log\max(a, b))$ 。

```
#include <iostream>
using namespace std;
const int N = 110000, P = 10007;
int n;
int a[N], len;
int ans;
void getDivisor() { //找 n 以内的因数
    len = 0;
    for(int i=1; ___(1)___ <= n; ++i)
        if(n%i==0) {
            a[++len] = i;
            if ( ___(2)___ != i)
                a[++len] = n / i;
        }
}
int gcd(int a, int b) {
    if (b== 0) {
        ___(3)___
    }
    return gcd (b, ___(4)___);
}
int main() {
    cin>>n;
    getDivisor() ;
    ans = 0;
    for(int i=1; i<=len; ++i) {
        for(int j=i+1; j<=len; ++j) {
            ans=(___(5)___)%P;
        }
    }
    cout << ans<< endl;
    return 0;
}
```

●选择题

(1)①处应填()。

A. $i*2$ B. $i*i$

C. $i+i$ D. $i*j*i$

(2)②处应填()。

A. n B. n/i

C. $n\%2$ D. $n-i$

(3)③处应填()。

A. `return a` B. `return b`

C. `return a+ b` D. `return a-b`

(4)④处应填()。

A. a/b B. a C. $a\%b$ D. b/a

(5)5 处应填()。

A. `god(a[i+1],a[j])` B. `ans+gcd(a[i+1],a[j-1])`

C. `gcd(a[i],a[j])` D. `ans+gcd(a[i], a[j])`

2. 对于一个 1 到 n 的排列 P (即 1 到 n 中每一个数在 P 中出现了恰好一次), 令 q_i 为第 i 个位置之后第一个比 P_i 值更大的位置, 如果不存在这样的位置, 则 $q_i=n+1$ 。举例来说, 如果 $n=5$ 且 P 为 1 5 4 2 3, 则 P 为 2 6 6 5 6。

下列程序读入了排列 P , 使用双向链表求解了答案。试补全程序。

数据范围 $1 \leq n \leq 10^6$

```
#include <iostream>
using namespace std;
const int N = 100010;
int n;
int L[N],R[N],a[N];
int main() {
    cin>>n;
    for(int i=1; i<=n; ++i) {
        int x;
        cin>>x;
        _____(1)_____
    }
    for(int i=1; i<=n; ++i)
        R[i] = _____(2)_____;
        L[i]=i-1;
    }
    for(int i=1; i<=n; ++i) {

        L[_____(3)_____] = L[a[i]];
        R[L[a[i]]] = R[_____(4)_____];
    }
    for(int i=1; i<=n; ++i) {
        cout<<_____(5)_____<<" ";
    }
    cout<< endl;
```

```
    return 0;
}
```

选择题

- (1) 1 处应填()。
- A. $a[i]=x$ B. $a[\times]=i$ C. $x=a[i]$ D. $a[x]=*i$
- (2) ②处应填()。
- A. $i+1$ B. $i*2$ C. $n-i$ D. $i+1$
- (3) ③处应填()。
- A. $R[a[i]]$ B. $L[a[i]]$ C. $L[R[i]]$ D. $R[L[i]]$
- (4) 4 处应填()。
- A. $a[L[i]]$ B. $a[i]$ C. $R[L[i]]$ D. $L[i]$
- (5) ⑤处应填()。
- A. $L[i]$ B. $a[i]$ C. $R[i]$ D. $R[L[i]]$