CSP-J 初赛模拟卷（第 12 套）

入门级 C++ 语言试题

一、单项选择题（共 **14** 题，每题 **2** 分，共计 **28** 分）

1．操作系统的作用是（ ）。

A．把源程序译成目标程序　B．便于进行数据管理

C．控制和管理系统资源　D．实现硬件之间的连接

2．NOI 的中文意思是（ ）。

A．中国信息学联赛　B．全国青少年信息学奥林匹克竞赛

C．中国青少年信息学奥林匹克竞赛　D．中国计算机协会

3．（ ）是一种先进先出的线性表。

A．栈　B．队列　C．哈希表（散列表）　D．二叉树

4．如果256 种颜色用二进制编码来表示，至少需要（ ）位。

A．6　B．7　C．8　D．9

5．设 $x = true, y = true, z = false$，以下逻辑运算表达式值为真的是（ ）。

A．$(y \vee z) \wedge x \wedge z$　B．$x \wedge (z \vee y) \wedge z$　C．$(x \wedge y) \wedge z$　D．$(x \wedge y) \vee (z \vee x)$

6．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．$(1001101011)_{2}$

7．表达式 $a*(b+c)*d$ 的后缀表达式为（ ），其中“*”和“+”是运算符。

A．**a+bcd　B．abc+*d*　C．abc+d**　D．*a*+bcd

8．一副纸牌除掉大小王有52张牌，四种花色，每种花色13张。假设从这52张牌中随机抽取13张纸牌，则至少（ ）张牌的花色一致。

A．4　B．2　C．3　D．5

9．前序遍历序列与中序遍历序列相同的二叉树为（ ）。

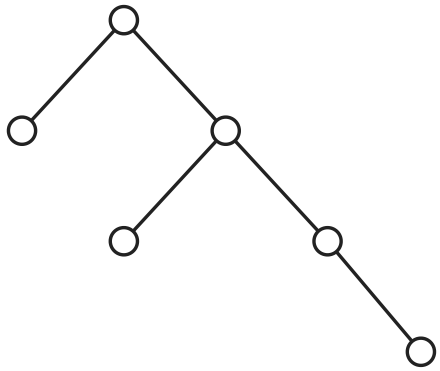
A．根结点无左子树

B．根结点无右子树

C．只有根结点的二叉树或非叶子结点只有左子树的二叉树

D．只有根结点的二叉树或非叶子结点只有右子树的二叉树

10. 一棵二叉树如右图所示，若采用顺序存储结构，即用一维数组元素存储该二叉树中的结点（根结点的下标为1，若某结点的下标为 i ，则其左孩子位于下标2i处、右孩子位于下标 $(2i+1)$ 处），则图中所有结点的最大下标为（ ）。



A. 6　　B. 10　　C. 12　　D. 15

11. 原字符串中任意一段连续的字符所组成的新字符串称为子串。 则字符“AAABBBCCC”共有（ ）个不同的非空子串。

A. 3　　B. 12　　C. 36　　D. 45

12. 现有一段文言文，要通过二进制哈夫曼编码进行压缩。简单起见，假设这段文言文只由4个汉字“之”、“乎”、“者”、“也”组成，它们出现的次数分别为700、600、300、400。若待合并结点权值相同，优先选取由其他结点合并形成的结点。那么，“也”字的编码长度可能是（ ）。

A. 1　　B. 2　　C. 3　　D. 4

13. 设 G 是有 n 个结点、m 条边 ($n \leq m$) 的连通图，必须删去 G 的（ ）条边，才能使得 G 变成一棵树。

A. $m-n+1$　　B. m - n　　C. $m+n+1$　　D. $n-m+1$

14. 双向链表中有两个指针域llink和rlink，分别指向该结点的前驱及后继。设p指向链表中的一个结点，它的左右结点均非空。现要求删除结点p，则下面语句序列中错误的是（ ）。

A. p->rlink->llink = p->rlink; p->llink->rlink = p->llink; delete p;
B. p->llink->rlink = p->rlink; p->rlink->llink = p->llink; delete p;
C. p->rlink->llink = p->llink; p->rlink->llink->rlink = p->rlink; delete p;
D. p->llink->rlink = p->rlink; p->llink->rlink->llink = p->llink; delete p;

二、阅读程序（程序输入不超过数组或字符串定义的范围，判断题正确填 √，错误填 ×；共 18 小题，共计 36 分，各小题分值见题目）

(1)

```
01 #include<iostream>
02 using namespace std;
03 int main()
04 {
05     const int SIZE=10;
06     int height[SIZE],num[SIZE],n,ans;
07     cin>>n;
08     for(int i=0;i<n;i++){
09         cin>>height[i];
10         num[i]=1;
11         for(int j=0;j<i;j++){
12             if((height[j]<height[i])&&(num[j]>=num[i]))
13                 num[i]=num[j]+1;
14         }
15     }
16     ans=0;
17     for(int i=0;i<n;i++){
18         if(num[i]>ans) ans=num[i];
19     }
20     cout<<ans<<endl;
21     return 0;
22 }
```

判断题

15. (1.5分) 如果height数组中的输入有负数，则程序会出错。（ ）

16. (1.5分) 程序输出的ans小于或等于n。（ ）

17. (1.5分) 将12行num[j]>=num[i]“改为”num[j]>num[i]“，程序的输出结果不会改变。（ ）

18. (1.5分) 将18行”num[i]>ans”改为”num[i]>=ans”，程序的输出结果不会改变。（ ）

单选题

19. (3分) 若输入的数据为 10 1 1 1 1 1 1 1 1 1 则程序的输出结果是（ ）。

A. 4　　B. 3　　C. 2　　D. 1

20. (3分) 若输入的数据为 10 3 2 5 11 12 7 4 10 15 6 因程序的输出结果是（ ）。

A. 4　　B. 5　　C. 6　　D. 7

(2)

```
01 #include <iostream>
02 using namespace std;
03
04 int n;
05 int d[1000];
06
07 int main() {
08     cin >> n;
09     for (int i = 0; i < n; ++i)
10         cin >> d[i];
11     int ans = -1;
12     for (int i = 0; i < n; ++i)
13         for (int j = 0; j < n; ++j)
14             if (d[i] < d[j])
15                 ans = max(ans, d[i] + d[j] - (d[i] & d[j]));
16     cout << ans;
17     return 0;
18 }
```

假设输入的n和d[i]都是不超过10000的正整数，完成下面的判断题和单选题：

判断题

21. (1.5分) n必须小于1000，否则程序可能会发生运行错误。（ ）

22. (1.5分) 输出一定大于等于0。（ ）

23. (1.5分) 若将第13行的“j = 0”改为“j = i + 1”，程序输出可能会改变。（ ）

24. (1.5分) 将第14行的“d[i] < d[j]”改为“d[i] != d[j]”，程序输出不会改变。（ ）

单选题

25. (3分) 若输入n为100，且输出为127，则输入的d[i]中不可能有（ ）。

A. 127　　B. 126　　C. 128　　D. 125

26. (3分) 若输出的数大于0，则下面说法正确的是（ ）。

A. 若输出为偶数，则输入的d[i]中最多有两个偶数
B. 若输出为奇数，则输入的d[i]中至少有两个奇数
C. 若输出为偶数，则输入的d[i]中至少有两个偶数
D. 若输出为奇数，则输入的d[i]中最多有两个奇数

(3)

```
01 #include<cstdio>
02 using namespace std;
03 int n, m;
04 int a[100], b[100];
05
06 int main() {
07     scanf("%d%d", &n, &m);
08     for (int i = 1; i <= n; ++i)
09         a[i] = b[i] = 0;
10     for (int i = 1; i <= m; ++i) {
11         int x, y;
12         scanf("%d%d", &x, &y);
13         if (a[x] < y && b[y] < x) {
14             if (a[x] > 0)
15                 b[a[x]] = 0;
16             if (b[y] > 0)
17                 a[b[y]] = 0;
18             a[x] = y;
19             b[y] = x;
20         }
21     }
22     int ans = 0;
```

```
23      for (int i = 1; i <= n; ++i) {
24          if (a[i] == 0)
25              ++ans;
26          if (b[i] == 0)
27              ++ans;
28      }
29      printf("%d", ans);
30      return 0;
31  }
```

假设输入的 n 和 m 都是正整数，x 和 y 都是在 $[1,n]$ 的范围内的整数，完成下面的判断题和单选题:· 判断题

判断题

27. (1.5分) 当m>0时，输出的值一定小于2n。（ ）

28. (1.5分) 执行完第27行的“++ans”时，ans一定是偶数。（ ）

29. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

30. (1.5分) 若每次执行到第13行时都有 x < y，则第15行不会被执行。（ ）

单选题

31. (3分) 若 m 个 x 两两不同，且 m 个 y 两两不同，则输出的值为（ ）

A. $2n-2m$　　B. $2n+2$　　C. $2n-2$　　D. $2n$

32. (3分) 若 m 个 x 两两不同，且 m 个 y 都相等，则输出的值为（ ）

A. $2n-2$　　B. $2n$　　C. $2m$　　D. $2n-2m$

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **15** 小题，共计 **43** 分，各小题分值见题目）

(1)

（质因数分解）给出正整数 n ，请输入将 n 质因数分解的结果，结果从小到大输出。例如:输入 $n=120$ ，程序应该输出22235，表示: $120=2\times2\ \ 2\times3\times5$ ，输入保证

$2 \leq n \leq 10^0$ 。提示:先从小到大枚举变量 i ，然后用 i 不停试除 n 来寻找所有的质因子。试补全程序。

```
#include <cstdio>
using namespace std;
int n, i;

int main() {
    scanf("%d", &n);
    for(i = __(1)__; __(2)__ <=n; i ++){
        __(3)__{
            printf("%d ", i);
            n = n / i;
        }
    }
    if(__(4)__)
        printf("%d ", __(5)__);
    return 0;
}
```

33. (3分) (1)处应填

A. 1　　B. n-1　　C. 2　　D. 0

34. (3分) (2)处应填

A. n/i　　B. n/(i*i)　　C. i*i　　D. i*i*i

35. (3分) (3)处应填

A. if(n%i==0)　B. if(i*i<=n)　C. while(n%i==0)　D. while(i*i<=n)

36. (3分) (4)处应填

A. n>1　　B. n<=1　　C. i<n/i　　D. i+i<=n

37. (3分) (5)处应填

A. 2　　B. n/i　　C. n　　D. i

(2)

完善程序

（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 $O(\sqrt{n})$，gcd函数的复杂度为 $O(\log \max(a,b))$ 。

```
01 #include <iostream>
02 using namespace std;
03 const int N = 110000, P = 10007;
04 int n;
05 int a[N], len;
06 int ans;
07 void getDivisor() {
08     len = 0;
09     for (int i = 1; __(1)__ <= n; ++i)
10         if (n % i == 0) {
11             a[++len] = i;
12             if ( __(2)__ != i) a[++len] = n / i;
13         }
14 }
15 int gcd(int a, int b) {
16     if (b == 0) {
17         __(3)__ ;
18     }
19     return gcd(b, __(4)__ );
20 }
21 int main() {
22     cin >> n;
23     getDivisor();
24     ans = 0;
25     for (int i = 1; i <= len; ++i) {
26         for (int j = i + 1; j <= len; ++j) {
27             ans = ( __(5)__ ) % P;
28         }
29     }
30     cout << ans << endl;
31     return 0;
32 }
```

38. (2分) (1)处应填：_____

39. (3分) (2)处应填：_____

40. (3分) (3)处应填：_____

41. (3分) (4)处应填：_____

42. (3分) (5)处应填：_____

(3)

完善程序:

（切割绳子）有 n 条绳子，每条绳子的长度已知且均为正整数。绳子可以以任意正整数长度切割，但不可以连接。现在要从这些绳子中切割出 m 条长度相同的绳段，求绳段的最大长度是多少。

输入:第一行是一个不超过100的正整数 n，第二行是 n 个不超过 10^6 的正整数，表示每条绳子的长度，第三行是一个不超过 10^8 的正整数 m。输出:绳段的最大长度，若无法切割，输出Failed。

```
01 #include<iostream>
02 using namespace std;
03 int n, m, i, lbound, ubound, mid, count;
04 int len[100]; // 绳子长度
05 int main()
06 {
07     cin >> n;
08     count = 0;
09     for (i = 0; i < n; i++)
10     {
11         cin >> len[i];
12         __(1)__;
13     }
14     cin >> m;
15     if (__(2)__)
16     {
17         cout << "Failed" << endl;
18         return 0;
19     }
20     lbound = 1;
21     ubound = 1000000;
22     while (__(3)__)
23     {
24         mid = __(4)__;
25         count = 0;
26         for (i = 0; i < n; i++)
27             __(5)__;
28         if (count < m)
29             ubound = mid - 1;
```

```
30          else
31              lbound = mid;
32      }
33      cout << lbound << endl;
34      return 0;
35  }
```

43. (2.5分) (1)处应填：_____

44. (2.5分) (2)处应填：_____

45. (3分) (3)处应填：_____

46. (3分) (4)处应填：_____

47. (3分) (5)处应填：_____