CSP-J 初赛模拟卷（第 05 套）

入门级 C++ 语言试题

一、单项选择题（共 **15** 题，每题 **2** 分，共计 **30** 分）

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

2．一个 32 位整型变量占用（　）个字节。

A．4　　B．8　　C．32　　D．128

3．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

4．根节点深度为0，一棵深度为h的满 $k(k>1)$ 又树，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．$k^{h-1}/(k-1)$

5．若有如下程序段，其中 s,a,b,c 均已定义为整型变量，且 a,c 均已赋值 （c大于0）。 `s = a; for (b = 1;b <= c; b++ ) s = s + 1;` 则与上述程序段修改 s 值的功能等价的赋值语句是（　）。

A．`s = a + b;`　　B．`s = a + c;`　　C．`s = s + c;`　　D．`s = b + c;`

6．现有一张分辨率为2048×1024 像素的 32 位真彩色图像。请问要存储这张图像，需要多大的存储空间？（　）。

A．16MB　　B．4MB　　C．8MB　　D．2MB

7．设简单无向图 G 有 16 条边且每个顶点的度数都是 2，则图 G 有（　）个顶点。

A．10　　B．12　　C．8　　D．16

8. 要求以下程序的功能是计算：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（ ）。

A. s = 1.0;　　B. for(n = 10; n > 1; n--)

C. s = s + 1 / n;　　D. cout << s << endl;

9. 在 8 位二进制补码中，10101011 表示的数是十进制下的（ ）。

A. 43　　B. -85　　C. -43　　D. -84

10. 有五副不同颜色的手套（共 10 只手套，每副手套左右手各 1 只），一次性从中取 6 只手套，请问恰好能配成两副手套的不同取法有（ ）种。

A. 120　　B. 180　　C. 150　　D. 30

11. 如果根的高度为 1,具有 61 个结点的完全二叉树的高度为（ ）。

A. 5　　B. 6　　C. 7　　D. 8

12. 下列几个32位IP地址中，书写错误的是（ ）。

A. 162.105.135.27　　B. 192.168.0.1　　C. 256.256.129.1　　D. 10.0.0.1

13. 寄存器是（ ）的重要组成部分.

A. 硬盘　　B. 高速缓存　　C. 内存　　D. 中央处理器（CPU）

14. 下列协议中与电子邮件无关的是（ ）。

A. POP3　　B. SMTP　　C. WTO　　D. IMAP

15. 周末小明和爸爸妈妈三个人一起想动手做三道菜。小明负责洗菜、爸爸负责切菜、妈妈负责炒菜。假设做每道菜的顺序都是：先洗菜10分钟，然后切菜10分钟，最后炒菜10分钟。那么做一道菜需要30分钟。注意：两道不同的菜的相同步骤不可以同时进行。例如第一道菜和第二道的菜不能同时洗，也不能同时切。那么做完三道菜的最短时间需要（ ）分钟。

A. 90　　B. 60　　C. 50　　D. 40

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 **37.5** 分，各小题分值见题目）

(1)

```
01 #include <cstdlib>
02 #include <iostream>
03 using namespace std;
04
05 char encoder[26] = {'C','S','P',0};
06 char decoder[26];
07
08 string st;
09
10 int main()  {
11     int k = 0;
12     for (int i = 0; i < 26; ++i)
13         if (encoder[i] != 0) ++k;
14     for (char x ='A'; x <= 'Z'; ++x) {
15         bool flag = true;
16         for (int i = 0; i < 26; ++i)
17             if (encoder[i] ==x) {
18                 flag = false;
19                 break;
20             }
21         if (flag) {
22             encoder[k]= x;
23             ++k;
24         }
25     }
26     for (int i = 0; i < 26; ++i)
27         decoder[encoder[i]- 'A'] = i + 'A';
28     cin >> st;
29     for (int i = 0; i < st.length( ); ++i)
```

```
30          st[i] = decoder[st[i] -'A'];
31      cout << st;
32      return 0;
33  }
```

判断题

16. (1.5分) 输入的字符串应当只由大写字母组成，否则在访问数组时可能越界。（ ）

17. (1.5分) 若输入的字符串不是空串，则输入的字符串与输出的字符串一定不一样。（ ）

18. (1.5分) 将第 12 行的 `i < 26` 改为 `i < 16`，程序运行结果不会改变。（ ）

19. (1.5分) 将第 26 行的 `i < 26` 改为 `i < 16`，程序运行结果不会改变。（ ）

单选题

20. (3分) 若输出的字符串为 ABCABCABCA，则下列说法正确的是（ ）。

A. 输入的字符串中既有 S 又有 P　B. 输入的字符串中既有 S 又有 B
C. 输入的字符串中既有 A 又有 P　D. 输入的字符串中既有 A 又有 B

21. (3分) 若输出的字符串为 CSPCSPCSPCSP，则下列说法正确的是（ ）。

A. 输入的字符串中既有 P 又有 A　B. 输入的字符串中既有 S 又有 R
C. 输入的字符串中既有 A 又有 S　D. 输入的字符串中既无 S 又无 A

(2)

```
01  #include <cstdio>
02  using namespace std;
03  int n;
04  int a[100];
05
06  int main() {
07      scanf("%d", &n);
08      for (int i = 1; i <= n; ++i)
09          scanf("%d", &a[i]);
10      int ans = 1;
11      for (int i = 1; i <= n; ++i) {
12          if (i > 1 && a[i] < a[i - 1])
13              ans = i;
14          while (ans < n && a[i] >= a[ans + 1])
15              ++ans;
```

```
16          printf("%d\n", ans);
17      }
18      return 0;
19  }
```

判断题

22. (1分) 第16行输出ans时，ans的值一定大于i。（ ）

23. (1分) 程序输出的ans小于等于n。（ ）

24. (1.5分) 若将第12行的 $<$ 改为 $!=$ ，程序输出的结果不会改变。（ ）

25. (1.5分) 当程序执行到第16行时，若ans-i>2,则 $a[i+1] \leq a[i]$ 。（ ）

单选题

26. (3分) 若输入的a数组是一个严格单调递增的数列，此程序的时间复杂度（ ）

A. O(log n)　　B. O(n^2)　　C. O(n log n)　　D. O(n)

27. (4分) 最坏情况下，此程序的时间复杂度是（ ）

A. O(n^2)　　B. O(log n)　　C. O(n)　　D. O(n log n)

(3)

```
01  #include <iostream>
02  using namespace std;
03  const int V = 101;
04  int n, m, ans, e[V][V];
05  bool visited[V];
06  void dfs(int x, int len)
07  {
08      int i;
09      visited[x] = true;
10      if (len > ans) ans = len;
11      for (i = 1; i <= n; i++)
12          if ((!visited[i]) && (e[x][i] != -1))
13              dfs(i, len + e[x][i]);
14      visited[x] = false;
15  }
16  int main()
17  {
18      int i, j, a, b, c;
19      cin >> n >> m;
```

```
20      for (i = 1; i <= n; i++)
21          for (j = 1; j <= n; j++)
22              e[i][j] = -1;
23      for (i = 1; i <= m; i++) {
24          cin >> a >> b >> c;
25          e[a][b] = c;
26          e[b][a] = c;
27      }
28      for (i = 1; i <= n; i++)
29          visited[i] = false;
30      ans = 0;
31      for (i = 1; i <= n; i++)
32          dfs(i, 0);
33      cout << ans << endl;
34      return 0;
35  }
```

输入表示一个没有重边、自环的无向图，顶点编号为 1 到 n，且 $1 \leq n \leq 100$、边权为正整数。

判断题

28. (1.5分) 第 19 行的输入中，如果满足 $m = n(n-1)/2$，则第 20～22 行的初始化可以省略。（ ）

29. (1.5分) 将第 31 行改为 `for (i = n; i >= 1; i--)`，程序结果不受影响。（ ）

单选题

30. (1.5分) 若输入如下，则程序输出为（ ）。

```
4 2
1 2 1
3 4 1
```

A. 8　　B. 4　　C. 2　　D. 1

31. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 1
2 3 1
3 4 1
4 1 1
1 3 1
2 4 1
```

A. 1　　B. 3　　C. 5　　D. 7

32. (3分) 若输入如下，则程序输出为（　）。

```
4 3
1 2 10
2 3 20
3 1 30
```

A. 30　　B. 50　　C. 90　　D. 120

33. (3分) 若输入如下，则程序输出为（　）。

```
4 6
1 2 10
2 3 20
3 4 30
4 1 40
1 3 50
2 4 60
```

A. 60　　B. 100　　C. 150　　D. 200

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **10** 小题，共计 **31** 分，各小题分值见题目）

(1)

（最小区间覆盖）给出 n 个区间，第 i 个区间的左右端点是 $[a_i, b_i]$。现在要在这些区间中选出若干个，使得区间 $[0, m]$ 被所选区间的并覆盖（即每一个 $0 \leq i \leq m$ 部在某个所选的区间中）。保证答案存在，求所选区间个数的最小值。

输入第一行包含两个整数 n 和 m （$1 \leq n \leq 5000, 1 \leq m \leq 10^3$）接下来 n 行，每行两个整数 $a_i, b_i (0 \leq a_i, b_i \leq m)$。提示:使用贪心法解决这个问题。先用 $O(n^2)$ 的时间复杂度排

序，然后贪心选择这些区间。试补全程序。

```
#include <iostream>

using namespace std;

const int MAXN = 5000;
int n, m;
struct segment { int a, b; } A[MAXN];

void sort() // 排序
{
    for (int i = 0; i < n; i++)
        for (int j = 1; j < n; j++)
            if (__(1)__)
            {
                segment t = A[j];
                __(2)__
            }
}

int main()
{
    cin >> n >> m;
    for (int i = 0; i < n; i++)
        cin >> A[i].a >> A[i].b;
    sort();
    int p = 1;
    for (int i = 1; i < n; i++)
        if (__(3)__)
            A[p++] = A[i];
    n = p;
    int ans =0, r = 0;
    int q = 0;
    while (r < m)
    {
        while (__(4)__)
            q++;
        __(5)__;
        ans++;
```

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

34. (3分) (1)处应填

A. A[j].b>A[j-1].b B. A[j].a<A[j-1].a
C. A[j].a>A[j-1].a D. A[j].b<A[j-1].b

35. (3分) (2)处应填

A. A[j+1]=A[j];A[j]=t; B. A[j-1]=A[j];A[j]=t;
C. A[j]=A[j+1];A[j+1]=t; D. A[j]=A[j-1];A[j-1]=t;

36. (3分) (3)处应填

A. A[i].b>A[p-1].b B. A[i].b<A[i-1].b
C. A[i].b>A[i-1].b D. A[i].b<A[p-1].b

37. (3分) (4)处应填

A. q+1<n&&A[q+1].a<=r B. q+1<n&&A[q+1].b<=r
C. q<n&&A[q].a<=r D. q<n&&A[q].b<=r

38. (3分) (5)处应填

A. r=max(r,A[q+1].b) B. r=max(r,A[q].b)
C. r=max(r,A[q+1].a) D. q++

(2)

完善程序:

(最大子矩阵和)给出 m 行 n 列的整数矩阵，求最大的子矩阵和(子矩阵不能为空)。

输入第一行包含两个整数 m 和 n ，即矩阵的行数和列数。之后 m 行，每行 n 个整数，描述整个矩阵。程序最终输出最大的子矩阵和。比如在如下这个矩阵中:

```
4 4
0 -2 -7 0
9 2 -6 2
-4 1 -4 1
-1 8 0 -2
```

拥有最大和的子矩阵为：

```
9 2
-4 1
-1 8
```

其和为15

```
3 3
-2 10 20
-1 100 -2
0 -2 -3
```

最大子矩阵和为128

```
4 4
0 -2 -9 -9
-9 11 5 7
-4 -3 -7 -6
-1 7 7 5
```

最大子矩阵和为26

```
01 #include <iostream>
02 using namespace std;
03 const int SIZE = 100;
04 int matrix[SIZE + 1][SIZE + 1];
05 int rowsum[SIZE + 1][SIZE + 1]; /* rowsum[i][j]记录第i行前j个数的和
   */
06 int m, n, i, j, first, last, area, ans;
07 int main()
08 {
09     cin >> m >> n;
10     for ( i = 1; i <= m; i++ )
11         for ( j = 1; j <= n; j++ )
12             cin >> matrix[i][j];
13     ans = matrix   __(1)__;
14     for ( i = 1; i <= m; i++ )
15         __(2)__;
16     for ( i = 1; i <= m; i++ )
17         for ( j = 1; j <= n; j++ )
18             rowsum[i][j] = __(3)__;
19     for ( first = 1; first <= n; first++ )
20         for ( last = first; last <= n; last++ )
21         {
22             __(4)__;
```

```
23              for ( i = 1; i <= m; i++ )
24              {
25                  area += __(5)__;
26                  if ( area > ans )
27                      ans = area;
28                  if ( area < 0 )
29                      area = 0;
30              }
31          }
32      cout << ans << endl;
33      return(0);
34  }
```

39. (3分) (1)处应填：_____

40. (3分) (2)处应填：_____

41. (3分) (3)处应填：_____

42. (3分) (4)处应填：_____

43. (4分) (5)处应填：_____