CSP-J 初赛模拟卷（第 11 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）

1．1TB代表的字节数是（ ）。

A．2的10次方　　B．2的20次方　　C．2的30次方　　D．2的40次方

2．在 PC 机中，PENTIUM(奔腾)、酷睿、赛扬等是指（ ）。

A．生产厂家名称　　B．硬盘的型号　　C．CPU 的型号　　D．显示器的型号

3．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

4．（ ）不属于操作系统。

A．Windows　　B．DOS　　C．Photoshop　　D．NOI Linux

5．FTP 可以用于（ ）。

A．远程传输文件　　B．发送电子邮件　　C．浏览网页　　D．网上聊天

6．以下不属于无线通信技术的是（ ）。

A．蓝牙　　B．Wifi　　C．GPRS　　D．以太网

7．以下哪个奖项是计算机科学领域的最高奖？（ ）

A．图灵奖　　B．鲁班奖　　C．诺贝尔奖　　D．普利策奖

8．二进制数101.11对应的十进制数是（ ）。

A．6.5　　B．5.5　　C．5.75　　D．5.25

9．10 个三好学生名额分配到 7 个班级，每个班级至少有一个名额，一共有（ ）种不同的分配方案。

A．84　　B．72　　C．56　　D．504

10．下列各无符号十进制整数中，能用八位二进制表示的数中最大的是（ ）。

A．296　　B．133　　C．256　　D．199

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

A．36　　B．48　　C．96　　D．192

12．关于拓扑排序，下面说法正确的是（ ）。

A．所有连通的有向图都可以实现拓扑排序

B．对同一个图而言，拓扑排序的结果是唯一的

C．拓扑排序中入度为0的结点总会排在入度大于0的结点的前面

D．拓扑排序结果序列中的第一个结点一定是入度为0的点

13．仿生学的问世开辟了独特的科学技术发展道路。人们研究生物体的结构、功能和工作原理，并将这些原理移植于新兴的工程技术中。以下关于仿生学的叙述，错误的是（ ）。

A．由研究蝙蝠，发明雷达　B．由研究蜘蛛网，发明因特网

C．由研究海豚，发明声呐　D．由研究电鱼，发明伏特电池

14．甲乙丙丁四人在考虑周末要不要外出郊游。 已知①如果周末下雨，并且乙不去，则甲一定不去；②如果乙去，则丁一定去；③如果丙去，则丁一定不去；④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_____，乙_____，丁_____，周末_____。

（1）甲：

A．去了；　B．没去。

（2）乙：

A．去了；　B．没去。

（3）丁：

A．去了；　B．没去。

（4）周末：

A．下雨；　B．没下雨。

15．（ ）是一种选优搜索法，按选优条件向前搜索，以达到目标。当搜索到某一步时，发现原先选择并不优或达不到目标，就退回一步重新选择。

A．回溯法　　B．枚举法　　C．动态规划　　D．贪心

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 **35.5** 分，各小题分值见题目）

(1)

```
01 #include <stdio.h>
02
03 int n;
04 int a[1000];
05
06 int f(int x)
07 {
08     int ret = 0;
09     for (; x; x &= x - 1) ret++;
10     return ret;
11 }
12
13 int g(int x)
14 {
15     return x & -x;
16 }
17
18 int main()
19 {
20     scanf("%d", &n);
21     for (int i = 0; i < n; i++) scanf("%d", &a[i]);
22     for (int i = 0; i < n; i++)
23         printf("%d ", f(a[i]) + g(a[i]));
24     printf("\n");
25     return 0;
26 }
27
```

判断题

16. (1.5分) 输入的 n 等于1001时，程序不会发生下标越界。（ ）

17. (1.5分) 输入的 $a[i]$ 必须全为正整数，否则程序将陷入死循环。（ ）

18. (1.5分) 当输入为“5 2 11 9 16 10”时，输出为“3 4 3 17 5”。（ ）

19. (1.5分) 当输入为”1 511998”时，输出为”18”。（ ）

20. (1.5分) 将源代码中 g 函数的定义(13-16行)移到main函数的后面，程序可以正常编译运行。（ ）

单选题

21. (3分) 当输入为”2 -65536 2147483647”时，输出为（ ）

A. “65532 33”　　B. “65552 32”　　C. “65535 34”　　D. “65554 33”

(2)

```
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

22. (1.5分) 当 $m>0$ 时，输出的值一定小于 $2n$。（ ）

23. (1.5分) 执行完第27行的“+ + ans”时，ans一定是偶数。（ ）

24. (1.5分) $a[i]$ 和 $b[i]$ 不可能同时大于0。（ ）

25. (1.5分) 右程序执行到第13行时， x 总是小于 y ，那么第15行不会被执行。（ ）

单选题

26. (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. $2n$

27. (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. $2n-2m$

(3)

```
01 #include<cstdio>
02 #include<cstring>
03 using namespace std;
04 int n, m;
05 int f[101][101];
06 int F[101][101];
07 int main(){
08     scanf("%d%d",&n,&m);//n的值在1到100之间
09     memset(f,-1,sizeof(f));
10     for(int i = 1; i <=  m; i++) {
11         int u,v,w;//w的值在0到10000之间
12         scanf("%d%d%d",&u,&v,&w);
13         f[u][v]=f[v][u]=w;
14     }
15     for(int k = 1; k <= n; k++)
16         for(int i=1;i <=n; i++)
17             for(int j=1;j<= n; j++)
18                 if(f[i][k]!=-1&&f[k][j]!=-1)
19                     if(f[i][j]==-1||f[i][j]>f[k][j]+f[i][k])
20                         f[i][j]=f[i][k]+f[k][j];
```

```
21      int ans=2147483647;
22      for(int i=1;i<=n;i++)
23          for(int j=1;j<=n; j++){
24              for(int x=1;x<=n; x++)
25                  for(int y=1;y<=n; y++)
26                      F[x][y]=f[x][y];
27              F[i][j]=F[j][i]=0;
28              for(int x=1;x<= n; x++)
29                  for(int y=1;y<=n; y++)
30                      if(F[x][y]==-1||F[x][y]>F[x][i]+F[i][y])
31                          F[x][y]=F[x][i]+F[i][y];
32              for(int x=1;x<=n; x++)
33                  for(int y=1;y<=n; y++)
34                      if(F[x][y]==-1||F[x][y]>F[x][j]+F[j][y])
35                          F[x][y]=F[x][j]+F[j][y];
36              int res=0;
37              for(int x=1;x<=n; x++)
38                  for(int y=1;y<x; y++)
39                      res +=F[x][y];
40              ans = res < ans ? res : ans;
41          }
42      printf("%d\n",ans);
43      return 0;
44  }
```

判断题

28. (1分) 14到16行，将外层到内层的循环变量依次调整为i、j、k，程序的运行的结果不变。（ ）

29. (2分) 这个程序的时间复杂度和m无关。（ ）

30. (2分) 20行的ans如果初始化为10^7时，可能无法得到正确结果。（ ）

31. (2分) 若将第27到30行的部分和31到34行的两个部分互换，程序的运行的结果不变。（ ）

单选题

32. (3分) 若输入数据为4 5/1 2 3/1 3 6/2 3 4/2 4 7/3 4 2(其中“/”为换行符)，则输出为（ ）。

A. 14　　B. 18　　C. 21　　D. 28

33. (3分) 这个程序使用了（ ）算法。

A. Floyd　　B. Dijkstra　　C. Prim　　D. Kruskal

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 **28** 分，各小题分值见题目）

(1)

完善程序:

(中位数 median) 给定 n(n 为奇数且小于 1000)个整数，整数的范围在 $0-m(0<m<2^{31})$ 之间，请使用二分法求这 n 个整数的中位数。所谓中位数，是指将这 n 个数排序之后，排在正中间的数。

```
01 #include <iostream>
02 using namespace std;
03 const int MAXN = 1000;
04 int n, i, lbound, rbound, mid, m, count;
05 int x[MAXN];
06 int main()
07 {
08     cin >> n >> m;
09     for (i = 0; i < n; i++)
10         cin >> x[i];
11     lbound = 0;
12     rbound = m;
13     while (__(1)__)
14     {
15         mid = lbound + (rbound - lbound) / 2;
16         __(2)__;
17         for (i = 0; i < n; i++)
18             if (__(3)__)
19                 __(4)__;
20         if (count > n / 2)
21             lbound = mid + 1;
22         else
23             __(5)__;
24         cout << mid << " " << lbound << " " << rbound << " " <<
   count << endl;
25     }
26     cout << rbound << endl;
27     return (0);
28 }
```

34. (3分) (1)处应填：_____

35. (3分) (2)处应填：_____

36. (3分) (3)处应填：_____

37. (3分) (4)处应填：_____

38. (2分) (5)处应填：_____

(2)

完善程序:

（二叉查找树）二叉查找树具有如下性质:每个节点的值都大于其左子树上所有节点的值、小于其右子树上所有节点的值。试判断一棵树是否为二叉查找树。输入的第一个行包含一个整数 n，表示这棵树有 n 个顶点，编号分别为 $1,2,...,n$，其中编号为1的为根结点。之后的第1行有三个数value, left_child, right_child，分别表示该节点关键字的值、左子节点的编号、右子节点的编号；如果不存在左子节点或右子节点，则用0代替。输出1表示这棵树是二叉查找树，输出0则表示不是。

```
01 #include <iostream>
02 using namespace std;
03 const int SIZE = 100;
04 const int INFINITE = 1000000;
05 struct node
06 {
07     int left_child, right_child, value;
08 }; node a[SIZE];
09 int is_bst( int root, int lower_bound, int upper_bound )
10 {
11     int cur;
12     if ( root == 0 )
13         return(1);
14     cur = a[root].value;
15     if ( (cur > lower_bound) && ( __(1)__ ) && (is_bst(
   a[root].left_child, lower_bound, cur ) == 1) && (is_bst( __(2)__,
   __(3)__, __(4)__ ) == 1) )
16         return(1);
17     return(0);
18 }
19 int main()
20 {
```

```
21      int i, n; cin >> n;
22      for ( i = 1; i <= n; i++ )
23          cin >> a[i].value >> a[i].left_child >> a[i].right_child;
24      cout << is_bst( __(5)__, -INFINITE, INFINITE ) << endl;
25      return(0);
26  }
```

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. (2分) (5)处应填：_____