# CSP-J 初赛模拟卷（第 15 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．以下奖项与计算机领域最相关的是（　）。

A．奥斯卡奖　　B．图灵奖　　C．诺贝尔奖　　D．普利策奖

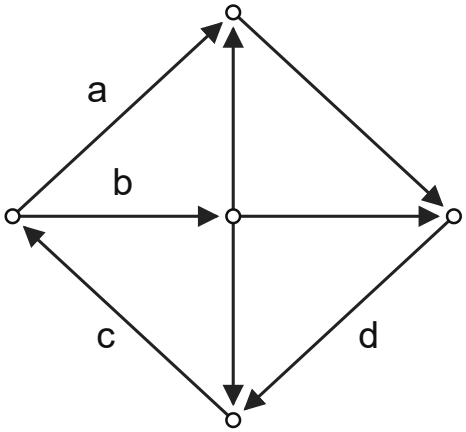
2．计算机应用的最早领域（　）

A．数值计算　　B．人工智能　　C．机器人　　D．过程控制

3．如果树根算第1层，那么一棵 $n$ 层的二叉树最多有（　）个结点。

A．$2^n-1$　　B．$2^n$　　C．$2^n+1$　　D．$2^{n+1}$

4．对一个有向图而言，如果每个节点都存在到达其他任何节点的路径，那么就称它是强连通图。例如，下图就是一个强连通图。事实上，在删除边（　）后，它依然是强连通图。




A．a　　B．b　　C．c　　D．d

5．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（ ）。

A．-1　　B．-3　　C．-5　　D．0

6．对于给定的序列 $\{a_k\}$，我们把 $(i,j)$ 称为逆序对当且仅当 $i<j$ 且 $a_i>a_j$。那么序列 $\{1,7,2,3,5,4\}$ 的逆序对数为（ ）个。

A．4　　B．5　　C．6　　D．7

7．以下关于字符串的判定语句中正确的是（ ）。

A．字符串是一种特殊的线性表　　B．串的长度必须大于零

C．字符串不可以用数组来表示　　D．空格字符组成的串就是空串

8．10000 以内，与 10000 互质的正整数有（ ）个。

A．2000　　B．4000　　C．6000　　D．8000

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

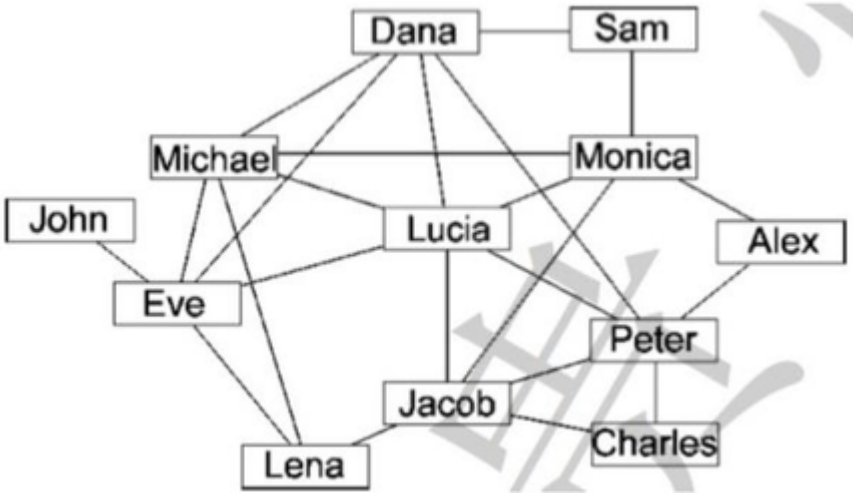
C．只有根结点的二叉树或非叶子结点只有左子树的二叉树

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. 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

11. Lucia 和她的朋友以及朋友的朋友都在某社交网站上注册了账号。下图是他们之间的关系图，两个人之间有边相连接这两个人是朋友，没有边相连代表不是朋友。这个社交网站的规则是:如果某人人向他（她）的朋友 B 分享了某张照片，那么 B 就可以对该照片进行评论；如果 B 评论了该照片，那么他（她）的所有朋友都可以看见这个评论以及被评论的照片，但是不能对该照片进行评论（除非 A 也向他（她）分享了该照片）。现在 Lucia 已经上传了一张照片，但是她不想让 Jacob 看见这张照片，那么她可以向以下朋友（ ）分享该照片。




A. Dana, Michael, Eve　　B. Dana, Eve, Monica

C. Michael, Eve, Jacob　　D. Micheal, Peter, Monica

12. 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 >>= 1　　B. x &= x - 1　　C. x |= x >> 1　　D. x <<= 1

13. 新学期开学了，小胖想减肥，健身教练给小胖制定了两个训练方案。方案一：每次连续跑3公里可以消耗300千卡（耗时半小时）；方案二：每次连续跑5公里可以消耗600千卡（耗时1小时）。小胖每周周一到周四能抽出半小时跑步，周五到周日能抽出一小时跑步。另外，教练建议小胖每周最多跑21公里，否则会损伤膝盖。 请问如果小胖想严格执行教练的训练方案，并且不想损伤膝盖，每周最多通过跑步消耗多少千卡？（ ）

A. 3000　　B. 2500　　C. 2400　　D. 2520

14. 有 10 个顶点的无向图至少应该有（ ）条边才有可能构成连通图。

A. 9　　B. 10　　C. 11　　D. 12

15. Linux下可执行文件的默认扩展名为（ ）。

A. exe　　B. com　　C. dll　　D. 以上都不是

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 **12** 小题，共计 **27** 分，各小题分值见题目）

**(1)**

```
01 #include<iostream>
02 using namespace std;
03 #define MAXN 20
04 int gu[MAXN][MAXN];
05 int luo(int n, int m) {
06     if(n <= 1 || m < 2)
07         return 1;
08     if(gu[n][m] != -1)
09         return gu[n][m];
10     int ans=0;
11     for(int i = 0; i <m; i += 2)
12         ans+=luo(n-1, i);
13     gu[n][m] = ans;
14     return ans;
15 }
16 int main() {
17     int n, m;
18     cin >> n >> m;
19     for(int i = 0; i < MAXN; i++)
20         for(int j = 0; j < MAXN; j++)
```

```
21              gu[i][j]= -1;
22      cout << luo(n, m);
23      return 0;
24  }
```

判断题

16. (1分) luo函数中，m的值不可能是奇数。（ ）

17. (1分) 若将第11行的“<”改为“<=”，程序的输出结果可能会改变。（ ）

18. (2分) 若将第8、9、13行删除，程序的运行的结果不变。（ ）

19. (2分) 在添加合适的头文件后，将第19到21行替换为“memset(gu, 255,sizeof(gu));”可以起到相同的作用。（ ）

单选题

20. (4分) 若输入数据为4 8，则输出为（ ）。

A. 7　　B. 8　　C. 15　　D. 16

21. (3分) 最坏情况下，此程序的时间复杂度是（ ）。

A. $O(m^2n)$　　B. O(nm!)　　C. $O(n^2)$　　D. $O(n^2m)$

**(2)**

```
01  #include <stdio.h>
02  #include <string.h>
03
04  char base[64];
05  char table[256];
06  char str[256];
07  char ans[256];
08
09  void init()
10  {
11      for (int i = 0; i < 26; i++) base[i] = 'A' + i;
12      for (int i = 0; i < 26; i++) base[26 + i] = 'a' + i;
13      for (int i = 0; i < 10; i++) base[52 + i] = '0' + i;
14      base[62] = '+', base[63] = '/';
15
16      for (int i = 0; i < 256; i++) table[i] = 0xff;
17      for (int i = 0; i < 64; i++) table[base[i]] = i;
18      table['='] = 0;
```

```
}

void decode(char *str)
{
    char *ret = ans;
    int i, len = strlen(str);
    for (i = 0; i < len; i += 4) {
        (*ret++) = table[str[i]] << 2 | table[str[i + 1]] >> 4;
        if (str[i + 2] != '=')
            (*ret++) = (table[str[i + 1]] & 0x0f) << 4 | table[str[i + 2]] >> 2;
        if(str[i + 3] != '=')
            (*ret++) = table[str[i + 2]] << 6 | table[str[i + 3]];
    }
}

int main()
{
    init();
    printf("%d\n", (int)table[0]);

    scanf("%s", str);
    decode(str);
    printf("%s\n", ans);
    return 0;
}
```

假设 char 是 8 位有符号类型，输入字符串长度为 4 的倍数，且输入不超过数组范围。

判断题

22. (1.5分) 输出的第二行一定是由小写字母、大写字母、数字和 +、/、= 构成的字符串。（ ）

23. (1.5分) 可能存在输入不同，但输出的第二行相同的情形。（ ）

24. (1.5分) 输出的第一行为 -1。（ ）

单选题

25. (3分) 设输入字符串长度为 $n$，decode 函数的时间复杂度为（ ）。

A. $O(\sqrt{n})$　　B. $O(n)$　　C. $O(n\log n)$　　D. $O(n^2)$

26. (3分) 当输入为 Y3Nx 时，输出的第二行为（ ）。

A. csp　　B. csq　　C. CSP　　D. Csp

27. (3.5分) 当输入为 Y2NmIDIwMjE= 时，输出的第二行为（ ）。

A. ccf2021　　B. ccf2022　　C. ccf 2021　　D. ccf 2022

三、完善程序（程序输入不超过数组或字符串定义的范围；共 **14** 小题，共计 **37** 分，各小题分值见题目）

**(1)**

(马走日)回溯算法实际上一个类似枚举的搜索尝试过程，主要是在搜索尝试过程中寻找问题的解，当发现已不满足求解条件时，就“回溯”返回，尝试其他路径。回溯法是一种选优搜索法按选向前索以达到目标。但当搜索到某一步时，若发现原先选择并优或达不到目标,则退回一步重新选择，这种走不通就退回去再走的技术称为回溯法，而满足回溯条件的某个状态的点称为回溯点。 马在中国象棋中以日字形规则移动。请编写一段程序，给定rxc大小的棋盘以及马的初始位置(m,n)，要求不能重复盘上的同一个点,计算马有多少途径可以遍历盘上的所有点。

```
01 #include<iostream>
02 using namespace std;
03 int r,c;
04 int cnt,tot;
05 int wayr[8]={2,2,1,-1,-2,-2,1,-1};
06 int wayc[8]={1,-1,2,2,1,-1,-2,-2};
07 bool mark[1001][1001];
08 bool check(int x, int y)
09 {
10     if(__(1)__) return true;
11     return false;
12 }
13 void search(int x,int y)
14 {
15     for(int i=0;i<8;i++)
16         if(__(2)__ && __(3)__)
17         {
18             mark[x+wayr[i]][y+wayc[i]]=true;
19             tot++;
20             if(__(4)__) cnt++;
21             search(__(5)__);
22             tot--;
```

```
23              mark[x+wayr[i]][y+wayc[i]]=false;
24          }
25  }
26  int main()
27  {
28      cnt=0;
29      int m,n;
30      cin>>r>>c>>m>>n;
31      if(!check(m,n)) cout<<0<<endl;
32      else if(r==1&&c==1) cout<<1<<endl;
33      else
34      {
35          mark[m][n]=true;
36          search(m,n);
37          cout<<cnt<<endl;
38      }
39  }
```

28. (3分) (1)处应填

A. x>=0 || y>=0 || x<r || y<c.  B. x>=0 || y>=0 && x<r || y<e.
C. x>=0 && y>=0 && x<r && y<c  D. x>=0&&y>=0 || x<r &&y<c

29. (3分) (2)处应填

A. check(wayr[i], wayc[i])  B. check(x+wayr[i], y+wayc[i])
C. !check(x+wayr[i],y+wayc[i])  D. !check(wayr[i],wayc[i])

30. (3分) (3)处应填

A. mark[wayr[i]][wayc[i]]  B. mark[x+wayr[i]][y+wayc[i]]
C. lmark[wayr[i]][wayc[i]]  D. !mark[x+wayr[i]][y+wayc[i]]

31. (3分) (4)处应填

A. tot==r * c-1  B. tot-=r * c  C. cnt==r * c-1  D. cnt==r * c

32. (3分) (5)处应填

A. wayr[i], wayc[i]  B. x-wayr[i], y-wayc[i]
C. x-wayr[i], y+wayc[i]  D. $x+\mathrm{wayr}[i], y+\mathrm{wayc}[i]$

**(2)**

完善程序

（子矩阵）给输入一个 $n1 \times m1$ 的矩阵 $a$，和 $n2 \times m2$ 的矩阵 $b$，问 $a$ 中是否存在子矩阵和 $b$ 相等。若存在，输出所有子矩阵左上角的坐标:若不存在输出“There is no answer”。

```
01 #include<iostream>
02 using namespace std;
03 const int SIZE = 50;
04 int n1,m1,n2,m2,a[SIZE][SIZE],b[SIZE][SIZE];
05 int main()
06 {
07     int i,j,k1,k2;
08     bool good ,haveAns;
09     cin>>n1>>m1;
10     for(i=1;i<=n1;i++)
11         for(j=1;j<=m1;j++)
12             cin>>a[i][j];
13     cin>>n2>>m2;
14     for(i=1;i<=n2;i++)
15         for(j=1;j<=m2;j++)
16             __(1)__;
17     haveAns=false;
18     for(i=1;i<=n1-n2+1;i++)
19         for(j=1;j<= __(2)__;j++){
20             __(3)__;
21             for(k1=1;k1<=n2;k1++)
22                 for(k2=1;k2<=__(4)__ ;k2++){
23                     if(a[i+k1-1][j+k2-1]!=b[k1][k2])
24                         good=false;
25                 }
26             if(good){
27                 cout<<i<<' '<<j<<endl;
28                 __(5)__;
29             }
30         }
31     if(!haveAns)
32         cout<<"There is no answer"<<endl;
33     return 0;
34 }
```

33. (2分) (1)处应填：_____

34. (2分) (2)处应填：_____

35. (2分) (3)处应填：_____

36. (2分) (4)处应填：_____

37. (2分) (5)处应填：_____

**(3)**

完善程序： (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。

```
#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;
}
```

38. (3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：_____