

# 几门级 CSP-J 第 9 套初赛模拟试题

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. 关于机器翻译,下列选项中正确的是( )。

- A. 常见的翻译软件只有金山词霸、金山快译两种
- B. 机器翻译的英文全称是 Machine Translation,简称 MT
- C. 百度和谷歌不具有在线翻译功能
- D. 机器翻译是利用计算机把一种自然语言转变成另一种机器语言

2. 以补码存储的 8 位有符号整数 10100011 的十进制表示为( )。

- A. -93
- B. 163
- C. -35
- D. -92

3. 关于网络协议,下面说法中正确的是( )。

- A. Internet 网络协议采用 TCP/IP 协议
- B. 我们所说的 TCP/IP 协议就是指传输控制协议
- C. www 浏览器使用的应用协议是 IPX/SPX
- D. 没有网络协议,网络也能实现可靠地传输数据

4. 以下程序当执行完毕后,输出的值为( )。

```
int f(int n)
{
    if (n==2 || n==1)
        return 1; //注意递归,验证,从最小的地方推
    else
        return f(n-1)+f(n-2);
}
```

cout<<f(9);

- A. 13
- B. 21
- C. 34
- D. 55

5. 下列关键字序列中,哪一项是堆( )。

- A. 16, 72, 31, 23, 94, 53
- B. 94, 23, 31, 72, 16, 53
- C. 16, 53, 23, 94, 31, 72
- D. 16, 23, 53, 31, 94, 72

6. 对  $n$  个不同的排序码进行冒泡排序,在下列哪种情况下比较的次数最多( )。

- A. 从小到大排列好的
- B. 从大到小排列好的
- C. 元素无序
- D. 元素基本有序

7.  $n$  为一个两位数,它的数码之和为  $a$ ,当  $n$  分别各乘以 3、5、7、9 以后得到 4 个乘积,如果每一个积的数码之和都为  $a$ ,那么这样的两位数  $n$  有( )个。

- A. 3
- B. 4
- C. 5
- D. 6

8. 二叉树第 10 层的结点数的最大数目为( )。

- A. 10
- B. 100
- C. 512
- D. 1024

9. 100 以内最大的素数是( )。

- A. 89
- B. 97
- C. 91
- D. 93

10. 15 张卡片,每张卡片上写有 3 个不同的汉字,任意 2 张上的汉字不完全相同;任意 6 张中,一定有 2 张,它们上面有共同的汉字。问:这 15 张卡片上最多有多少个不同的汉字?( )。

- A. 30
- B. 45
- C. 35
- D. 180

11. 仅由数字 1, 2, 3 组成的七位数中,相邻数字均不相同的七位数的个数是( )。

- A. 128
- B. 252
- C. 343
- D. 192

12. 有甲、乙、丙、丁四支球队参加的足球循环赛,每两队都要赛一场,胜得 3 分,负者得 0 分,

如果踢平,两队各得 1 分。现在甲、乙、丙分别得了 7 分、1 分和 6 分,已知甲和乙踢平,那么丁得( )分。

A. 1

B. 3

C. 4

D. 7

13. 若一组记录的排序码为(46, 79, 56, 38, 40, 84)则利用快速排序的方法,以第一个记录为基准得到的一次划分结果为( )。

A. 38, 40, 46, 56, 79, 84

B. 40, 38, 46, 79, 56, 84

C. 40, 38, 46, 56, 79, 84

D. 40, 38, 46, 84, 56, 79

14. 一棵 6 节点二叉树的中序遍历为 ABDGECF,先序遍历为 DBACEGF,后序遍历为( )。

A. DGBEFAC

B. ABGEFCD

C. GBEACFD

D. ABCDEFG

15. 下面哪种图不一定是树( )。

A. 无回路的连通图

B. 有  $n$  个结点,  $n-1$  条边的连通图

C. 每对结点间都有通路的图

D. 连通但删去任意一条边则不连通的图

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×;除特别说明外,判断题 1.5 分,选择题 3 分,共计 40 分)

1.

```
01 #include<bits/stdc++.h>
02 using namespace std;
03 const int N=2e5;
04 int a[N],s[N];
05 int main()
06 {
07     int n,m;
08     cin>>n>>m;
09     memset(s,0,sizeof(s));
10     for(int i=1;i<=n;i++)
11     {
12         cin>>a[i];
13         s[i]=a[i]+s[i-1];
14     }
15     int l,r;
16     while(cin>>l>>r)
17     {
18         cout<<s[r]-s[l-1]<<endl;
19     }
20     return 0;
21 }
```

● 判断题

(1) 输出只能是正整数。( )

(2) 将 03 行的 2e5 改为 2e10 输出结果不变。( )

(3) 将第 09 行删除,程序运行结果不会改变。( )

(4) 只要输入 int 数据类型的数据,输出结果就是正确的。( )

● 选择题

(5) 如输入的 5 3 1 2 3 4 5 2 4,则输出的结果为( )。

A. 6

B. 9

C. 12

D. 15

(6) 本题涉及下列哪一项的数据范围偏小( )。

A. a[N]

B. s[N]

C. m

D. 以上都不对



2.

```

01  #include<stdio>
02  bool pd(long long n)
03  {
04      if(n==1)
05          return false;
06      for(long long i=2;i<n;i++)
07          if(n%i==0)return false;
08      return true;
09  }
10  int main()
11  {
12      long long n,i,c=0;
13      int INF=1<<30; //1 左移 30 位,把 INF 初始化为较大的数
14      scanf("%d",&n);
15      for(i=2;i<=INF;i++)
16      {
17          if(pd(i))
18          {
19              c++;
20              if(c==n)
21              {
22                  printf("%d",i);
23                  return 0;
24              }
25          }
26      }
27      printf("\n over");
28      return 0;
29  }

```

●判断题:

- (1) 上述代码中,将第 13 行修改为  $INF=1<<40$ ,输出结果一定不变。( )
- (2) 上述代码中,将第 23 行修改为 `break` 或 `continue` 这两种情况后,相同的输入,在这两种情况,输出结果也一定相同。( )
- (3) 上述代码中,将第 23 行修改为 `break` 后,相同的输入,变量 `c` 的值和未修改前一定相同。( )
- (4) 上述代码中,将第 23 行修改为 `break` 后,相同的输入,输出结果也一定相同。( )

●选择题:

- (5) 当输入为:8,输出为( )。
 

A. 17	B. 19\n over	C. 19	D. 23\n over
-------	--------------	-------	--------------
- (6) 上述代码中,将第 06 行的 `i<n` 修改为( )后功能不变,效率更高。
 

A. $i*i<=n$	B. $i<n/2$	C. $i<n/3$	D. $i<n/4$
-------------	------------	------------	------------

3.

```

01  #include<bits/stdc++.h>
02  using namespace std;
03  int s[100001],a[100001],n,ans1,ans2;

```

```

04 int main()
05 {
06     while (scanf("%d", &a[++n]) != EOF);
07     n--;
08     for (int i = n; i >= 1; i--) {
09         s[i] = 1;
10         for (int j = i + 1; j <= n; j++) {
11             if (a[j] <= a[i]) {
12                 s[i] = max(s[i], s[j] + 1);
13             }
14         }
15         ans1 = max(ans1, s[i]);
16     }
17     for (int i = 1; i <= n; i++) {
18         s[i] = 1;
19         for (int j = 1; j <= i; j++) {
20             if (a[j] < a[i]) {
21                 s[i] = max(s[i], s[j] + 1);
22             }
23         }
24         ans2 = max(ans2, s[i]);
25     }
26     printf("%d%d", ans1, ans2);
27     return 0;
28 }

```

### ●判断题:

- (1) 若输入的序列是一个单调递增序列, 则 ans1 的值为 1。 ( )
- (2) 若输入的序列是一个单调递减序列, 则 ans2 的值为 1。 ( )
- (3) 对输入序列数据处理中, ans1 的值越大, ans2 的值将会越小。 ( )
- (4) 输入的数值不能为负数。 ( )

### ●选择题:

- (5) 若输入 389 207 155 300 299 170 158 65, 输出第一个数为 ( )。
  - A. 3
  - B. 4
  - C. 5
  - D. 6
- (6) 若输入 0 -1 0 -1, 则输出 ( )。
  - A. 2 2
  - B. 4 1
  - C. 3 2
  - D. 2 3

### 三、完善程序 (单选题, 每题 3 分, 共计 30 分)

1. (SPFA) 给定一个有  $n$  个顶点 (从 1 到  $n$  编号),  $m$  条边的有向图 (其中某些边权可能为负, 但保证没有负环)。请你计算从 1 号点到其他点的最短路。试补全程序。

```

01 #include <bits/stdc++.h>
02 using namespace std;
03 const int N=200010;
04 const int inf=0x3f3f3f3f;
05 int h[N], e[N], ne[N], ret, w[N];
06 int dis[N];
07 bool st[N];
08 int n, m;

```

```

09 void add(int a,int b,int c){
10     e[ret]=b;
11     w[ret]=c;
12     ne[ret]=h[a];
13     h[a]=ret++;
14 }
15 void spfa()
16 {
17     memset(dis,inf,sizeof(dis));
18     dis[1]=0;
19     queue<int>q;
20     q.push(1);
21     st[1]=true;
22     while(q.size()){
23         int t=q.front();
24         q.pop();
25         st[t]=false;
26         for(int i=①; ②;i=ne[i]){
27             int j=e[i];
28             if(dis[j]>dis[t]+w[i])
29             {
30                 ③;
31                 if(!st[j]){
32                     ④;
33                     st[j]=true;
34                 }
35             }
36         }
37     }
38 }
39 int main(){
40     scanf("%d%d",&n,&m);
41     memset(h,-1,sizeof h);
42     while(m--){
43         int a,b,c;
44         scanf("%d%d%d",&a,&b,&c);
45         ⑤;
46     }
47     spfa();
48     for(int i=2;i<=n;++i)
49         printf("%d\n",dis[i]);
50     return 0;
51 }

```

(1)①处应填( )。

A. 1

B. -1

C. h[t]

D. t



(2) ②处应填( )。

A.  $i=0$ B.  $i>0$ C.  $i!=-1$ D.  $i>-1$ 

(3) ③处应填( )。

A.  $dis[j]=dis[t]+w[t]$ B.  $dis[j]=abs(dis[t]+w[i])$ C.  $dis[j]=dis[t]+w[i]$ D.  $dis[j]=dis[i]+w[i]$ 

(4) ④处应填( )。

A.  $q.push(j)$ B.  $q.push(i)$ C.  $q.push(st[j])$ D.  $q.push(1)$ 

(5) ⑤处应填( )。

A.  $spfa()$ B.  $add()$ C.  $add(a,b,c)$ D.  $spfa$ 

2. (01 背包) 在网友的国度中共有  $n$  种不同面额的货币, 第  $i$  种货币的面额为  $a[i]$ , 你可以假设每一种货币都有无穷多张。为了方便, 我们把货币种数为  $n$ 、面额数组为  $a[1..n]$  的货币系统记作  $(n, a)$ 。

在一个完善的货币系统中, 每一个非负整数的金额  $x$  都应该可以被表示出, 即对每一个非负整数  $x$ , 都存在  $n$  个非负整数  $t[i]$  满足  $a[i] * t[i]$  的和为  $x$ 。然而, 在网友的国度中, 货币系统可能是不完善的, 即可能存在金额  $x$  不能被该货币系统表示出。例如在货币系统  $n=3, a=[2, 5, 9]$  中, 金额 1, 3 就无法被表示出来。

两个货币系统  $(n, a)$  和  $(m, b)$  是等价的, 当且仅当对于任意非负整数  $x$ , 它要么均可以被两个货币系统表示出, 要么不能被其中任何一个表示出。

现在网友们打算简化一下货币系统。他们希望找到一个货币系统  $(m, b)$ , 满足  $(m, b)$  与原来的货币系统  $(n, a)$  等价, 且  $m$  尽可能的小。他们希望你来协助完成这个艰巨的任务: 找到最小的  $m$ 。

```
01 #include<bits/stdc++.h>
```

```
02 using namespace std;
```

```
03 int a[105], f[25005];
```

```
04 int main()
```

```
05 {
```

```
06     int T, n;
```

```
07     scanf("%d", &T);
```

```
08     while( ① ) {
```

```
09         scanf("%d", &n);
```

```
10         for(int i=0; i<n; i++)
```

```
11             scanf("%d", &a[i]);
```

```
12         sort(a, a+n);
```

```
14         int x= ② ; //x 代表 a[i] 的最大值
```

```
15         memset(f, 0xcf, sizeof(f));
```

```
16         f[0]= ③ ;
```

```
17         for(int i=0; i<n; i++) {
```

```
18             for(int j=a[i]; j<=x; j++) {
```

```
19                 f[j]=max(f[j], ④ );
```

```
20             }
```

```
21         }
```

```
22         int res=0;
```

```
23         for(int i=0; i<n; i++) {
```

```
24             if( ⑤ )
```

```
25                 res++;
```

```
26         }
```

```

27     printf("%d\n",res);
28 }
29     return 0;
30 }

```

(1)①处应填( )。

- A. T                      B. T--                      C. 1                      D. 0

(2)②处应填( )。

- A. a[0]                      B. a[n-1]                      C. a[1]                      D. a[n]

(3)③处应填( )。

- A. -1                      B. 0                      C. 1                      D. n

(4)④处应填( )。

- A. f[j+a[i]]+1                      B. f[a[i]]+1                      C. f[j-a[i]]                      D. f[j-a[i]]+1

(5)⑤处应填( )。

- A. f[a[i]]==1                      B. f[a[i]]==0                      C. f[a[i]]>1                      D. f[a[i]]>2