2024 (CSP-J1) 入门级 C++语言试题

认证时间: 2024 年 9月 21 日 09:30~11:30

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.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.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. $(1001101011)_{2}$

3.1MB 等于(　　)。
A. 1000 字节　　B.1024 字节
B. 1000 × 1000 字节　　D. 1024 × 1024 字节

4.广域网的英文缩写是(　　)。
A. LAN　　B. WAN　　C. MAN　　D. LNA

5. 设某算法的时间复杂度函数的递推方程是 T(n) = T(n - 1) + n（n 为正整数）及 T(0) = 1，则该算法的时间复杂度为（)。
A. O(log n)　　B. O(nlog n)　　C. O(n)　　D. $O(n^2)$

6.如果开始时计算机处于小写输入状态,现在有一只小老鼠反复按照 CapsLock.字母键 A、字母键 S、字母键 D、字母键 F 的顺序循环按键，即 CapsLock、A、S.D. F、CapsLock、A、S、D、F、....屏幕.上输出的第 81 个字符是字母(　　)。
A. A　　B. S　　C. D　　D. a

7.根节点深度为 0，-棵深度为 h 的满 k (k>1) 叉树，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. $(k^{h+1})/(k- 1)$

8.以下排序算法中，不需要进行关键字比较操作的算法是(　　)。
A.基数排序　　B.冒泡排序　　C.堆排序　　D.直接插入排序

9. 给定一个含 N 个不相同数字的数组，在最坏情况下，找出其中最大或最小的数，至少需要 N-1 次比较操作。则最坏情况下，在该数组中同时找最大与最小的数至少需要(　　)次比较操作。(⌈ ⌉ 表示向上取整，⌊ ⌋ 表示向下取整)
A.⌈3N/2⌉ - 2　　B. ⌊ 3N/2⌋ - 2　　C.2N-2　　D.2N - 4

10.下面的故事与(　　)算法有着异曲同工之妙。从前有座山，山里有座庙，

庙里有个老和尚在给小和尚讲故事:“从前有座山，山里有座庙，庙里有个老和尚在给小和尚讲故事:‘从前有座山，山里有座庙，庙里有个老和尚给小和尚讲故事....”
A. 枚举　　B.递归 C.贪心 D.分治

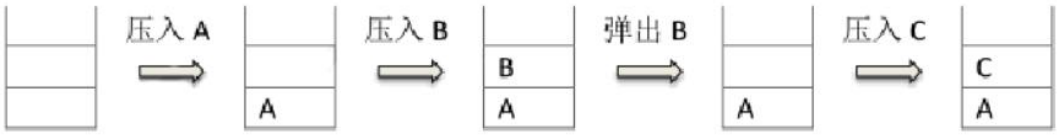
11.由四个没有区别的点构成的简单无向连通图的个数是(　　)。
　　A. 6　　B.7　　C. 8　　D　.9

12.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.

13.10000 以内，与 10000 互质的正整数有(　　)个。
　　A.2000　　B. 4000　　C. 6000　　D. 8000

14.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

15.下图中所使用的数据结构是(　　)



A.哈希表　B.栈　　C.队列　　D.二叉树

三、阅读程序写结果

1.
```
#include <iostream>
using namespace std;
int n, d[100];
bool v[100];
int main() {
    cin>>n;
    for(int i=0; i<n; ++i) {
```

```
        scanf("%d", d[i]);
        v[i] = false;
    }
    int cnt = 0;
    for(int i=0; i<n; ++i) {
        if (!v[i]) {
            for(int j=i; !v[j]; j=d[j]) {
                v[j] = true;
            }
            ++cnt;
        }
    }
    cout<<cnt;
    return 0;
}
```

●判断题

(1)将第 7 行的 d+i 换成&d[i],程序运行不受影响。 ( )

(2)第 12 行的!v[i]与 v[i]==false 语句意思一致。 ( )

(3 程序的输出结果 cnt 至少等于 1。 ( )

(4)若输入的数组 d 中有重复的数字,则程序会进入死循环。 ( )

●选择题

(5)若输入数字为 5 1 2 3 4 5,则输出为( )。

A.0 B.1 C.2 D.5

(6)若输入数字为 10 7 1 4 3 2 5 9 8 0 6,则输出为( )。

A.3 B.6 C.7 D.8

2.

```
#include <iostream>
using namespace std;
string s;
long long magic(int l, int r)
{
    long long ans = 0;
    for (int i = l; i <= r; ++i)
    {
        ans = ans * 4 + s[i] - 'a' + 1;
    }
    return ans;
}
int main()
{
    cin >> s;
    int len = s.length();
    int ans = 0;
```

```
15   for (int l1 = 0; l1 < len; ++l1){
16       for (int r1 = l1; r1 < len; ++r1){
           bool bo = true;
18           for (int l2 = 0; l2 < len; ++l2){
19               for (int r2 = l2; r2 < len; ++r2){
                     if (magic(l1, r1) == magic(l2, r2) && (l1 != l2 || r1 != r2))
                     {
                         bo = false;
                     }
                 }
             }
             if (bo)
             {
                 ans += 1;
             }
         }
    }
    cout << ans << endl;
    return 0;
}
```

判断题

(1)输出一定不为0。 ( )

(2)将15,16行与18,19行对调,答案不变。 ( )

(3)输入字符串最多句含5种字符串,若更多则容易互相冲突。 ( )

选择题

(5)输入ahacaha.则输出的结果是( )。

A.16 B.17 C. 18 D.19

(6)输入abcbbcc,则输出的结果是( )。

A.16 B.20 C. 18 D.19

3.

```
1 #include<iostream>
2 using  namespace  std;
3 long  long  a[1000001]={0};
4 long  long  maxx;
5 int  main()
6 {
7        unsigned  long  long  ans=0;
8        int  n;
9        cin>>n;
10        for(int  i=1;i<=n;i++)
11        {
12                long  long  t;
13                cin>>t;
```

```
14                  a[t]++;
15                  maxx=max(maxx,t);
16          }
17          for(int  i=1;i<=maxx/i;i++)
18          {
19                  if(!a[i])continue;
20                  for(int  j=i;j<=maxx;j++)
21                  {
22                          if(i*j>maxx)break;
23                          ans+=a[i]*a[j]*a[i*j];
24                          if(i!=j)ans+=a[i]*a[j]*a[i*j];
25                  }
26          }
27          cout<<ans;
28          return  0;
29 }
```

判断题

(1)unsigned int 是计算机编程语言中一种表示大于等于 0 的整数类型（　　）

(2) 删除 19 行，对程序结果没有影响。　　（　　）

(3) 当 n 大于 3 时，输出一定大于 3!　　（　　）

选择题

(4)输入 5 1 2 4 8 16 后结果为（　　）

A.1　　B.4　　C.15　　D.0

(5)当输入为 n 1……1（n 个 1）时，结果为(　　)

A. 0　　B.n　　C.n^2　　D.n^3

(6)该程序的平均时间复杂度接近于(　　)

A. O(n)　　B. O(nlogn)　　C.O(n^2)　　D.O(n*sqrt(n))

三、完善程序(单选题，每小题 3 分，共计 30 分)

1. 给定一个数列，共有 n 个正数，现在有 m 次询问，每次询问给出一个 t，求满足最小的 k 使得从第一个数到第 k 个数之和大于等于 t。

```
1 #include<cstdio>
2 #include<iostream>
3 #include<cstring>
4 using namespace std;
5 const int maxn =250010;
6 int n,m;
7 int a[maxn];
8 int main()
```

```
9 {
10  cin >>n>>m;
11  memset(a,0x3f,sizeof a);
12  a[0]=0;
13  for(int i=1;i<=n;i++)
14  {
15      cin >>a[i];
16      ____①____
17  }
18  while(m--)
19  {
20          int t,k,p;
21          cin >>t;
22          ②_________
23          while( ___③____)
24          {
25              if(a[k+p] <= t)  k=k+p,___④_____;
26              else  ____⑤_____;
27              if(!p)break;
28          }
29      cout <<k<<endl;
30  }
31   return 0;
32 }
```

① 1 处应填（　　）

A. a[i] = a[i-1] + a[i-2];　　B. a[i] -= a[i-1];

C. a[i] += a[i-1];　　D. a[i] = a[i-1]-a[i-2];

② 2 处应填（　　）

A. k=0,p=0;　　B.k=1,p=1;

C. k=1,p=0;　　D. k=0,p=1;

③ 3 处应填（　　）

A. !p　　B. p　　C. !k　　D. k

④ 4 处应填（　　）

A. p/=2　　B. p*=2　　C. p+=2　　D. p-=2

⑤ 5 处应填（　　）

A. p/=2　　B. p*=2　　C. p+=2　　D. p-=2

2 在笔直的无限长的海岸线一侧有许多小岛。解放军准备在海岸线上安装雷达，假设每个雷达的监测范围为 d，当小岛与某雷达的距离不超过 d 时，该小岛可以被雷达覆盖。我们使用笛卡尔坐标系，定义海岸线为 x 轴，海的一侧在 x 轴上方。 现在给出每个小岛的具体坐标以及雷达的检测范围，请你求出能够使所有小岛都被雷达覆盖所需的最小雷达数目。

第一行输入两个整数 n 和 d，分别代表小岛数目和雷达检测范围。接下来 n 行，每行输入两个整数，分别代表小岛的 x，y 轴坐标。输出一个整数，代表所需的最小雷达数目，若没有解决方案则所需数目输出 -1。

```
1 #include<iostream>
2 #include<cmath>
3 #include<algorithm>
4 using namespace std;
5 const int N=1e3+10;
6 struct Node
7{
8    double l;
9    double r;
10 }a[N];
11 bool cmp(Node a,Node b)
12 {
13       ____①____
14 }
15 int main()
16{
17   int n,d;
18   scanf("%d%d",&n,&d);
19   bool is_cover=false;
20   for(int i=0;i<n;i++)
21   {
22       int x,y;
23       scanf("%d%d",&x,&y);
24       if(____②_________)
25       {
26           is_cover=true;
27           break;
28       }
29       double len=sqrt(d*d-y*y);
30     ____③_______
31  }
```

```
32  if(is_cover) {
33      cout<<"-1";
34  }
35  sort(a,a+n,cmp);
36  int ans=0;
37  double point=-2e9;
38  for(int i=0;i<n;i++)
39  {
40      if(a[i].l>point)
41      {
42          ans++;
43          ___④___
44      }
45   }
46  if(____⑤____) printf("%d\n",ans);
47  return 0;
48 }
```

1) 1 处应填（　　）

A. return a.r<b.r;　　B. return a.r>b.r;

C. return a.l<b.l;　　D. return a.l>b.l;

2) 2 处应填（　　）

A. x>d　　B. x<d

C. y>d　　D. y<d;

3) 3 处应填（　　）

A. a[i].l=x-len,a[i].r=x-len

B. a[i].l=x-len,a[i].r=x+len

C. a[i].l=x+len,a[i].r=x+len

D. a[i].l=x+len,a[i].r=x-len

4) 4 处应填（　　）

A. point=a[i].l　　B.point=a[i].r

C. point=a[i].l-len　　D.point=a[i].r-len

5) 5 处应填（　　）

A. !is_cover　　B. is_cover　　C. !ans　　D. ans