

第二十五届全国青少年信息学奥林匹克联赛初赛

CSP-S C++语言模拟试题

竞赛时间:2019 年 10 月

选手注意:

- 试题纸共有 10 页, 答题纸共有 2 页, 满分 100 分。请在答题纸上作答, 写在试题纸上的一律无效。
- 不得使用任何电子设备 (如计算器、手机、电子词典等) 或查阅任何书籍资料。

一、单项选择题 (共 15 题, 无特殊说明每题 2 分, 共计 30 分; 每题有且仅有一个正确选项)

1. 不同类型的存储器组成了多层次结构的存储器体系, 按存取速度从快到慢排列的是 ()。

- A. 快存/主存/辅存
- B. 快存/辅存/主存
- C. 主存/辅存/外存
- D. 外存/主存/辅存

2. RAM 中的信息是 ()

- A. 防止计算机病毒侵入所使用
- B. 生产厂家预先写入
- C. 计算机工作时随机写入
- D. 专门用于计算机开机时自检用

3. 在 24×24 点阵的字库中, 汉字“一”与“编”的字模占用字节数分别是 ()

- A. 32、72
- B. 32、32
- C. 72、32

D.72、72

4.计算机的运算速度取决于给定的时间内，它的处理器所能处理的数据量。处理器一次能处理的数据量叫字长。已知 64 位的奔腾处理器一次能处理 64 个信息，相当于()字节。

A.2

B.8

C.1

D.16

5.操作系统的作用是()。

A.便于进行数据管理

B.实现硬件间的连接

C.控制和管理系统资源

D.把源程序译成目标程序

6.计算机之所以称为“电脑”，是因为()

A.计算机具有逻辑判断功能

B.计算机是人类大脑功能的延伸

C.计算机有强大的记忆能力

D.计算机瞬息万变我控制功能

7.在计算机领域中，通常用英文单词“BYTE”来表示()。

A.字长

B.字节

C.二进制位

D.字

8. 计算机病毒是指 ()。

- A. 已感染病毒的磁盘
- B. 已感染病毒的程序
- C. 具有破坏性的特制程序
- D. 能传染给用户的磁盘病毒

9. GB2312-80 规定了一级汉字 3755 个，二级汉字 3008 个，其中二级汉字字库中的汉字是以 () 为序排列的。

- A. 以 ASCII 码
- B. 以机内码
- C. 以部首
- D. 以笔画的多少

10. 用户在网上最常用的一类信息查询工具叫做 ()。

- A. ISP
- B. 搜索引擎
- C. 网络加速器
- D. 离线浏览器

11. 设栈 S 的初始状态为空，现有 5 个元素组成的序列{1, 2, 3, 4, 5}，对该序列在 S 栈上依次进行如下操作 (从序列中的 1 开始，出栈后不再进栈)：进栈、进栈、进栈、出栈、进栈、出栈、进栈。试问出栈的元素序列是 ()。

- A. 2, 3
- B. 2, 1
- C. 5, 4, 3, 2, 1
- D. 3, 4

12. 设循环队列中数组的下标范围是 n ，其中头尾指针分别是 f 和 r ，尾指针指向实际存储的最后一个元素的下一个位置，则其元素个数是 ()。

- A. $r-f+1$
- B. $(r-f) \bmod n+1$
- C. $r-f$
- D. $(r-f+n) \bmod n$

13. 电线上停着两种鸟 (A, B)，可以看出两只相邻的鸟就将电线分为了一个线段。这些线段可公为两类：一类是两端的小鸟相同；另一类是两端的小鸟不相同。已知：电线上两个顶点上正好停着相同的小鸟，试问两端为不同小鸟的线段数目一定是 ()。

- A. 数目固定
- B. 偶数
- C. 奇数
- D. 可奇可偶

14. 从未排序序列中挑选元素，并将其依次放入已排序序列（初始时空）的一端，这种排序方法称为 ()。

- A. 快速排序
- B. 插入排序
- C. 选择排序
- D. 归并排序

15. 对一个满二叉树， m 个树叶， x 分枝结点， n 个结点，则 ()。

- A. $m=x-1$
- B. $n=2x-1$
- C. $x+m=2n$
- D. $n=x+m$

二、阅读程序(共 18 题，判断对错 2 分，选择题 3 分，共计 40 分；每题有且仅有一个正确选项)

【程序清单 1】

```
#include <bits/stdc++.h>
using namespace std;

long long n;

bool isprime(long long x)
{
    for (long long i = 2; i * i <= x; i++)
        if (x % i == 0)
            return false;
    return true;
}

int main()
{
    cin >> n;
    for (long long i = n; i >= 1; i--)
    {
        if (isprime(i))
        {
            cout << i << " ";
        }
    }
}
```

```
        return 0;
    }
}
return 0;
}
```

16.isprime 函数可以判断一个 long long 范围内的正整数是否为质数

- A.正确
- B.错误

17.若输入 2018，则输出 2017

- A.错误
- B.正确

18.该程序中 if 中的 return 0; 可以改成 break;

- A.正确
- B.错误

19.若将该程序中 if 中的 return 0; 删除，则该程序输出的结果为 n 以内所有的质数

- A.错误
- B.正确

20.若将 isprime(i) 改为 !isprime(i) 并删除 if 中的 return 0; 则若输入 4，输出 ()

- A.4
- B.4 1
- C.3 2
- D.2 3

21.若仅将 for (long long i = n; i>= 1; i--) 改成 for (long long i = 1; i<= n; i++), 则若输入 5, 输出 ()

A.2 3 5

B.1 2 3 5

C.2

D.1

【程序清单 2】

```
#include <iostream>
```

```
#include <cstdio>
```

```
using namespace std;
```

```
int n, a[100], b[100];
```

```
int main()
```

```
{
```

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

```
    for (int i = 0; i < n; i++) scanf("%d", &a[i]);
```

```
    b[n - 1] = a[n - 1];
```

```
    for (int i = n - 2; i >= 0; i--) {
```

```
        if (a[i] < b[i + 1])
```

```
            b[i] = a[i];
```

```
        else
```

```
            b[i] = b[i + 1];
```

```
    }
```

```
    int c = 0, d = 0;
```

```
for (int i = 0; i < n; i++) {  
    if (a[i] > c)  
        c = a[i];  
    if (c == b[i])  
        d++;  
}  
cout << d << endl;  
return 0;  
}
```

22.上述代码中,把 `for (int i = n - 2; i >= 0; i--)` 改为 `for(int i = n - 1; i >= 0; i--)` 输出结果一定不变

A.错误

B.正确

23.上述代码中,如果输入的 `a[i]` 为负数不影响该代码所要实现的功能

A.正确

B.错误

24.若输入的 `a[i]` 中有重复数字不影响该代码所要实现的功能

A.错误

B.正确

25.输入的 `n` 最大可以为 100

A.正确

B.错误

26.若输入数据为

8

1 3 2 4 5 7 6 8

输出为 ()

A.6

B.3

C.4

D.5

27.上述代码中 b 数组的含义是

A.a 数组的前缀最小值

B.a 数组的后缀最大值

C.a 数组的后缀最小值

D.a 数组的前缀最大值

【程序清单 3】

```
#include <cstdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int n, r, num[10000];
```

```
bool mark[10000];
```

```
void print()
```

```
{
```

```
    for (int i = 1; i <= r; i++)
```

```
        cout << num[i] << " ";
```

```
    cout << endl;
```

```
}  
void search(int x)  
{  
    for (int i = 1; i <= n; i++)  
        if (!mark[i])  
        {  
            num[x] = i;  
            mark[i] = true;  
            if (x == r) print();  
            search(x + 1);  
            mark[i] = false;  
        }  
}  
int main()  
{  
    cin >> n >> r;  
    search(1);  
    return 0;  
}
```

28. 上述代码中是在求出从 n 个数中选出 r 个数的所有排列

- A. 错误
- B. 正确

29. 上述代码中若将 `mark[i] = false;` 删除, 输出结果一定不变

A. 正确

B. 错误

30. 上述代码中 `x` 的值不会超过 `r`

A. 错误

B. 正确

31. 上述代码中 `x` 的值不会超过 `n`

A. 正确

B. 错误

32. 若输入数据为

3 2

则 `print` 被调用了 () 次

A. 8

B. 3

C. 6

D. 2

33. 当 `r = n` 时上述代码的时间复杂度为 ()

A. $O(n!)$

B. $O(n)$

C. $O(2^n)$

D. $O(n! * n)$

三、阅读程序(共 10 题，共计 30 分；每题有且仅有一个正确选项)

克鲁斯卡尔求最小生成树思想：首先将 n 个点看做 n 个独立的集合，将所有边快排（从小到大）。然后，按排好的顺序枚举每一条边，判断这条边连接的两个点是否属于一个集合。若是，则将这条边加入最小生成树，并将两个点所在的集合合并为一个集合。若否，则跳过。直到找到 $n-1$ 条边为止。

【程序清单 1】

```
#include <iostream>
#include <algorithm>
using namespace std;
struct point {
    int x;
    int y;
    int v;
};
point a[10000];

int cmp(const point &a, const point &b) {
    if ( ① ) return 1;
    else return 0;
}

int fat[101];
int father(int x) {
    if (fat[x] != x) return fat[x]= ② ;
    else return fat[x];
}
```

```

void unionn(int x,int y) {
    int fa = father(x);
    int fb = father(y);
    if (fa != fb) fat[fa] = fb;
}

int main() {
    int i, j, n, m, k = 0, ans = 0, cnt = 0;
    cin >> n;
    for (i = 1; i <= n; i++)
        for (j = 1; j <= n; j++)
        {
            cin >> m;
            if (m != 0) {
                k++;
                a[k].x = i;
                a[k].y = j;
                a[k].v = m;
            }
        }
    sort(a + 1, a + 1 + k, ③ );

    for (i = 1; i <= n; i++) {
        fat[i] = i;
    }
}

```

```
for (i = 1; i <= k; i++) {  
    if (father(a[i].x) != ④ ) {  
        ans += a[i].v;  
        unionn(a[i].x, a[i].y);  
        cnt++;  
    }  
    if ( ⑤ ) break;  
}  
  
cout << ans;  
return 0;  
  
}
```

34. 请选择 ① 应该填写的代码。

- A. a.v > b.v
- B. a.v < b.v
- C. a.v <= b.v
- D. a.v >= b.v

35. 请选择 ② 应该填写的代码。

- A. father(fat[x])
- B. x
- C. fat[fat[x]]
- D. father(x)

36. 请选择 ③ 应该填写的代码。

- A. cmp
- B. greater<point>
- C. point
- D. less<point>

37. 请选择 ④ 应该填写的代码。

- A. father(a[i].y)
- B. a[i].y
- C. fat[a[i].y]
- D. a[i].x

38. 请选择 ⑤ 应该填写的代码。

- A. cnt == n-1
- B. cnt
- C. cnt>n
- D. cnt==n

欧拉回路问题由七桥问题而来，其基本问题是是否能一次性不重复地走遍这七座桥，转换为数学问题中的图论就是指的是从图中的一个顶点出发，是否能够一次性不回头地走遍所有的边，假设图为无向图，并且确保有欧拉回路，现在给定起点要求出欧拉回路的一种具体路径，算法代码如下：

【程序清单 2】

```
#include <iostream>
#include <ctime>
using namespace std;
```

```

int G[5][5];
int visited[5][5];
int n, m, s;
void euler(int u) {
    for (int v = 0; v < n; v++) {
        if (G[u][v] && ⑥ ) {
            cout << u << "->" << v << endl;
            visited[u][v] = visited[v][u] = ⑦;
            euler( ⑧ );
        }
    }
}

int main() {
    cin >> n >> m;
    for (int i = 0; i < m; i++) {
        cin >> u >> v;
        G[u][v] = ⑨ = 1;
    }
    cin >> s;
    euler( ⑩ );
    return 0;
}

```

39. 请选择 ⑥ 应该填写的代码

- A.visited[u][v]
- B.visited[u][v]==1
- C.G[v][u]==1
- D.!visited[u][v]

40. 请选择 ⑦ 应该填写的代码。

- A.v
- B.1
- C.u
- D.0

41. 请选择 ⑧ 应该填写的代码。

- A.u
- B.visited[u][v]
- C.G[u][v]
- D.v

42. 请选择 ⑨ 应该填写的代码。

- A.G[u][v]
- B.visited[u][v]
- C.visited[v][u]
- D.G[v][u]

43. 请选择 ⑩ 应该填写的代码。

- A.n

- B.s
- C.1
- D.0

第二十五届全国青少年信息学奥林匹克联赛初赛

CSP-S模拟卷参考答案

一、单项选择题（共15题，每题2分，共计30 分）

1	2	3	4	5	6	7	8
A	C	D	B	C	B	B	C
9	10	11	12	13	14	15	
C	B	D	D	B	C	D	

二、阅读程序写结果（共3题，每题10分，共计40分）

第一题	16	17	18	19	21	21
	B	B	A	A	A	D
第二题	22	23	24	25	26	27
	A	B	A	B	C	C
第三题	28	29	30	31	32	33
	B	B	B	B	C	D

三、完善程序 (每空3 分, 共计30 分)

第一题	34	35	36	37	38
	B	A	A	A	A
第二题	39	40	41	42	43
	D	B	D	D	B