

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 9 日 15:30~17:30

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, 每题 1.5 分, 共计 22.5 分; 每题有且仅有一个正确选项)

1. 以下哪一个不是 CCF 举办 CSP-J/S 的理由 ()

- A. 为青少年提供权威、公平、公正、公开和客观的评价体系
- B. 通过认证鼓励更多的人学习算法和编程
- C. 让青少年和广大家长为 CCF 提供运营资金
- D. 为非计算机专业人士提供权威、公平、公正、公开和客观的评价体系

2. 以下关于 CSP-J/S 的描述错误的是 ()

- A. 任何人都可以自愿报名参加 CSP-J/S
- B. CSP-J/S 是 CCF 独立主办的认证, 和任何其他机构主办的等级考试无关
- C. CSP-J/S 和 NOIP 有密切关系
- D. CSP-J/S 认证成绩优异者, 可参加 NOI 省级选拔, 省级选拔成绩优异者可参加 NOI

3. -128 的补码表示为 ()

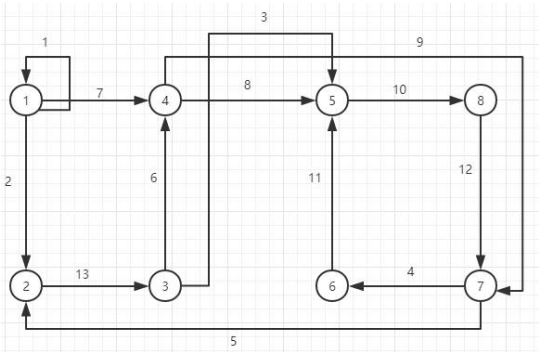
- A. 00000000 B. 00000001 C. 10000000 D. 11111111

4. 以下不属于 TCP 拥塞控制算法的是 ()
A. 慢启动 B. 拥塞避免 C. 快启动 D. 快速重传

5. 以下不是基于 UDP 协议的是 ()
A. DNS B. RIP C. TELNET D. TFTP

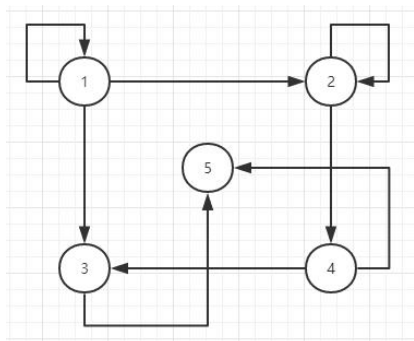
6. 定义如下函数 add_edge 和全局变量:
`int to[MAX],nxt[MAX],h[MAX],top;`
`void add_edge(int u,int v){`
 `to[++top]=v,nxt[top]=h[u],h[u]=top;`
`}`

如下图节点编号从 1 开始，按边的编号顺序，以前向星的方式存储，请问 `nxt[h[3]]` 的值为 ()



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

7. 如下图所示，从节点 1 走 6 步走到节点 5 的方案数有多少种 ()



A. 5 B. 8 C. 7 D. 6

8. 同时查找 $2n$ 个数中的最大值和最小值，最少比较次数为()。

A. $3(n-2)/2$ B. $3n-2$ C. $4n-2$ D. $2n-2$

9. 设 A 和 B 是两个长为 n 的有序数组，现在需要将 A 和 B 合并成一个排好序的数组，请问任何以元素比较作为基本运算的归并算法最坏情况下至少要做 () 次比较。

A. n^2 B. $n \log_2 n$ C. $2n$ D. $2n-1$

10. G 是一个非连通简单无向图，共有 36 条边，则该图至少有 () 个顶点

A. 10 B. 9 C. 8 D. 7

11. 由四个不同的点构成的简单无向连通图的个数是 ()

A. 32 B. 35 C. 38 D. 31

12. 前缀表达式 $- + * 4 + 2 3 1 5$ 的值为 ()

A. 16 B. 17 C. 19 D. 15

13. $2+3*(4-(5+6))/7$ 的逆波兰表达式为 ()

- A. 2 3 4 5 6 - + * 7 / +
- B. 2 3 4 5 6 - + * / 7 +
- C. 2 3 4 5 6 + - * 7 / +
- D. 2 3 4 5 6 + + * / 7 -

14. 若某算法的计算时间表示为递推关系:

$$T(n) = 2.5T(2n/5) + n \log_2^2 n$$

则该算法的复杂度为 ()

- A. $O(n)$
- B. $O(n \log_2 n)$
- C. $O(n \log_2^2 n)$
- D. $O(n \log_2^3 n)$

15. 若某算法的计算时间表示为递推关系:

$$T(n) = 3T(n/4) + n \log_2 n$$

则该算法的复杂度为 ()

- A. $O(n)$
- B. $O(n \log_2 n)$
- C. $O(n \log_2^2 n)$
- D. $O(n \log_2^3 n)$

二、不定项选择题 (共 5 题, 每题 1.5 分, 共计 7.5 分; 每题有一个或多个正确选项, 多选或少选均不得分)

1. 下列关于二叉树前序、中序和后序遍历的描述正确的是 ()

- A. 已知前序遍历序列和中序遍历序列, 可以唯一确定一棵唯一的二叉树
- B. 已知后序遍历序列和中序遍历序列, 可以唯一确定一棵唯一的二叉树
- C. 已知前序遍历序列和后序遍历序列, 是不能确定一棵唯一的二叉树
- D. 已知前序遍历序列、中序遍历序列和后序遍历序列才能确定一棵唯一的二叉树

2. 下列网络技术描述错误的是 ()

- A.OSI 四层模型是参考模型是国际标准化组织（ISO）制定的一个用于计算机或通信系统间互联的标准体系
- B.IP 协议可以不经 MAC 寻址直接进行高效的交流
- C.建立 TCP 通信必须经过三次握手
- D.ICMP 协议要经过 IP 协议进行转发

3.对句子“zyd ak ioi and noi”进行哈夫曼编码，下列哪些是可能产生的结果（）

- A. 字母 i 编码为 11
- B. 字母 o 编码为 000
- C. 字母 k 编码为 011
- D. 字母 n 编码为 110

4.以下日期中为星期三的是（）

- A.1992 年 10 月 28 日
- B.1949 年 10 月 1 日
- C.1986 年 3 月 20 日
- D.2010 年 6 月 23 日

5.当你 CSP 初赛考砸了，以下哪个是恰当的行为（）

- A.乖乖回家等明年
- B.回家对 NOI 系列网站进行 DDOS
- C.回家好好复习，订正错题
- D.在 11 月 16 日前，在自己博客上发出 CSP 复赛的题目解析

三、问题求解(共 2 题，每题 5 分，共计 10 分；每题全部答对得 5 分，没有部分分)

1.现有变量 a, b, c, d ，取值范围均为 $[0, 15]$ ，假设每个值出现的概率相同，则表达式 $a|b|c|d$ 的值能被 3 整除的概率_____（|为计算机中的或运算符，结果用分数形式表达）

2. 假设 h_n 代表一串由 a、b、c 组成的文字的长度位 n 的字符串方案数，并且保证没有连续的 a，请问 $h_{10} =$ _____ (2 分)， $h_n =$ _____ (3 分)
(填通项公式)

四、阅读程序写结果(共 4 题，每题 8 分，共计 32 分)

1.

```
#include<iostream>
using namespace std;
int a,b,c;

int* cal(int *p,int &q,int r){
    q+=r;
    *p+=q;
    return p;
}

int main(){
    cin>>a>>b>>c;
    cout<<*cal(&a,b,c)<<" "<<a<<" "<<b<<" "<<c;
}
```

输入: 1 2 3

输出: _____

2.

```
#include<iostream>
#include<cstring>
#define LL long long
using namespace std;
LL l,r;
```

```

LL f[12][10][10][2][2][2],a[20];
LL Dfs(LL now,LL p,LL pp,LL _4,LL _8,LL top,LL hw){
    if(_4&&_8) return 0;
    if(!now) return hw;
    if(!top && f[now][p][pp][_4][_8][hw]!=-1) return f[now][p][pp][_4][_8][hw];
    LL Up=top?a[now]:9;
    LL ret(0);
    for(LL i=0;i<=Up;++i)
        ret+=Dfs(now-1,i,p, _4|(i==4),_8|(i==8), top&&(i==Up) ,hw|(i==pp&&i==p));
    if(!top) f[now][p][pp][_4][_8][hw]=ret;
    return ret;
}
inline LL Solve(LL x){
    LL tot(0);
    while(x){
        a[++tot]=x%10;
        x/=10;
    }
    if(tot!=11) return 0;
    LL ret(0);
    for(LL i=1;i<=a[tot];++i)
        ret+=Dfs(tot-1,i,0,(i==4),(i==8),i==a[tot],0);
    return ret;
}
int main(){
    cin>>l>>r;
    memset(f,-1,sizeof(f));
    cout<<Solve(r)-Solve(l-1);
}

```

```
    return 0;
}
```

输入: 12121284000 12121285550

输出: _____

3.

```
#include <cstdio>
#include <cstring>
```

```
const                                     char
alphe[27][6]={{"00000"}, {"00001"}, {"00010"}, {"00011"}, {"00100"}, {"00101"}, {"00110"}, {"00111"}, {"01000"}, {"01001"}, {"01010"}, {"01011"}, {"01100"}, {"01101"}, {"01110"}, {"01111"}, {"10000"}, {"10001"}, {"10010"}, {"10011"}, {"10100"}, {"10101"}, {"10110"}, {"10111"}, {"11000"}, {"11001"}, {"11010"}};
int maps[30][30];
int num[1000];
char str[100];
```

```
int main()
{
    int n;
    scanf("%d", &n);
    for (int icase = 1; icase <= n; ++icase)
    {
        memset(num, 0, sizeof(num));
        for (int i = 0; i < 30; ++i)
            for (int j = 0; j < 30; ++j)
                maps[i][j] = -1;
```



```

int row, col;
scanf("%d%d", &row, &col);
getchar();

gets(str);
int cur = 0;
for (int i = 0; str[i] != '\0'; ++i)
{
    int sym = 0;
    if (str[i] <= 'Z' && str[i] >= 'A')
        sym = str[i] - 'A' + 1;
    for (int j = 0; j < 5; ++j)
        num[++cur] = alphe[sym][j] - '0';
}
int tot = 0;
int x = 0;
int y = 0;
maps[0][0] = num[++tot];
while (tot < cur)
{
    while (y+1 < col && maps[x][y+1] == -1)
        maps[x][++y] = num[++tot];

    while (x+1 < row && maps[x+1][y] == -1)
        maps[++x][y] = num[++tot];

    while (y-1 >= 0 && maps[x][y-1] == -1)
        maps[x][--y] = num[++tot];
}

```

```

        while (x-1 >= 0 && maps[x-1][y] == -1)
            maps[--x][y] = num[++tot];
    }
    printf("%d ", icafe);
    for (int i = 0; i < row; ++i)
    {
        for (int j = 0; j < col; ++j)
        {
            if (maps[i][j] == -1)
                printf("0");
            else
                printf("%d", maps[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

输入:

2

4 4 ACM

5 2 HI

输出: _____

```

4.
#include <iostream>
#include <string>

using namespace std;

size_t equalizeLength(string &s1, string &s2)
{
    size_t len1 = s1.size(), len2 = s2.size();
    if (len1 < len2)
    {
        for (int i = 0; i < len2 - len1; ++i)
            s1 = '0' + s1;
        return len2;
    }
    else if (len1 > len2)
    {
        for (int i = 0; i < len1 - len2; ++i)
            s2 = '0' + s2;
    }
    return len1;
}

string strAddition(string s1, string s2)
{
    string ret;
    int carry = 0;
    size_t len = equalizeLength(s1, s2);

```

```

for (int i = len - 1; i >= 0; --i)
{
    int firstBit = s1.at(i) - '0';
    int secondBit = s2.at(i) - '0';

    int sum = (firstBit ^ secondBit ^ carry) + '0';
    ret = static_cast<char>(sum) + ret;

    carry = (firstBit & secondBit) | (firstBit & carry) | (secondBit & carry);
}
if (carry)
    ret = '1' + ret;
return ret;
}

long int Karatsuba(string s1, string s2)
{
    size_t len = equalizeLength(s1, s2);

    // base case
    if (len == 0) return 0;
    if (len == 1) return (s1[0] - '0') * (s2[0] - '0');

    size_t floor = len / 2;
    size_t ceil = len - floor;
    string a = s1.substr(0, floor);
    string b = s1.substr(floor, ceil);

```

```

string c = s2.substr(0, floor);
string d = s2.substr(floor, ceil);

long int p1 = Karatsuba(a, c);
long int p2 = Karatsuba(b, d);
long int p3 = Karatsuba(strAddition(a, b), strAddition(c, d));
return (1<<(2 * ceil)) * p1 + (1<<(ceil)) * (p3 - p1 - p2) + p2;
}

int main() {
    string s1,s2;
    cin>>s1>>s2;
    cout <<Karatsuba(s1, s2) << endl;
    return 0;
}

```

输入: 10101010 111101
 输出: _____

五、完善程序(每题 14 分, 共计 28 分)

1. (链表反转) 单向链表反转是一道经典算法问题, 比如有一个链表是这样的, 1->2->3->4->5, 反转后成为 5->4->3->2->1。现给定如下链表节点的定义:

```

struct LinkNode{
    int value;
    LinkNode* next;};

```

非递归实现:

```

LinkNode* Reverse(LinkNode* header){

```

```

if (header == NULL || header->next == NULL){
    return header;
}

LinkNode* pre = header, *cur = header->next;
pre->next = NULL;
while(cur != NULL)
{
    LinkNode* next = _____;
    _____ = pre;
    pre = cur;
    cur = next;
}
return pre;}

```

递归实现:

```

LinkNode * Reverse(LinkNode * head){
    if (head == NULL || head->next == NULL){
        return head;
    }
    LinkNode * newhead = _____;
    _____ = head;
    head->next = _____;
    return newhead;
}

```

2. (一般图最大匹配—带花树) 在二分图中, 我们使用匈牙利算法, 通过寻找增广路找到二分图最大匹配。而在一般图中的二分图最大匹配问题, 我们称作一般图最大匹配。带花树算法则是用于求解一般图最大匹配问题, 由于一般图中存在奇环, 增广路到奇环上会出现如下的问题:


```

inline int read()
{
    RG int x=0,t=1;RG char ch=getchar();
    while((ch<'0' || ch>'9') && ch!='-') ch=getchar();
    if(ch=='-') t=-1, ch=getchar();
    while(ch<='9' && ch>='0') x=x*10+ch-48, ch=getchar();
    return x*t;
}

struct Line{int v,next;}e[MAXL];
int h[MAX],cnt=1;
inline void Add(int u,int v){e[cnt]=(Line){v,h[u]};h[u]=cnt++;}
int match[MAX],pre[MAX],f[MAX],vis[MAX],tim,dfn[MAX];
int n,m,ans;
int getf(int x){return x==f[x]?x:f[x]=getf(f[x]);}
int lca(int u,int v)
{
    ++tim;u=getf(u);v=getf(v);
    while(dfn[u]!=tim)
    {
        dfn[u]=tim;
        u=getf(pre[match[u]]);
        if(v) swap(u,v);
    }
    return u;
}

queue<int> Q;
void Blossom(int x,int y,int w)
{

```



```

while (getf(x) != w)
{
    pre[x]=y, y=_____ ;
    if (vis[y]==2) vis[y]=1, Q.push(y);
    if (getf(x)==x) f[x]=w;
    if (getf(y)==y) f[y]=w;
    x=pre[y];
}
}
bool Aug(int S)
{
    for (int i=1; i<=n; ++i) f[i]=i, vis[i]=pre[i]=0;
    while (!Q.empty()) Q.pop(); Q.push(S); vis[S]=1;
    while (!Q.empty())
    {
        int u=Q.front(); Q.pop();
        for (int i=h[u]; i; _____)
        {
            int v=e[i].v;
            if (getf(u)==getf(v) || vis[v]==2) continue;
            if (!vis[v])
            {
                vis[v]=2; pre[v]=u;
                if (!match[v])
                {
                    for (int x=v, lst; x; x=lst)
                        lst=match[pre[x]], _____, match[pre[x]]=x;
                    return true;
                }
            }
        }
    }
}

```

```

        }
        vis[match[v]]=1,Q.push(match[v]);
    }
    else
    {
        int w=_____ ;
        Blossom(u,v,w);
        Blossom(v,u,w);
    }
}
}
return false;
}
int main()
{
    n=read();m=read();
    for(int i=1;i<=m;++i)
    {
        int u=read(),v=read();
        Add(u,v);Add(v,u);
    }
    for(int i=1;i<=n;++i)if(!match[i])ans+=Aug(i);
    printf("%d\n",ans);
    for(int i=1;i<=n;++i)printf("%d ",_____);puts("");
    return 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1.5 分，共计22.5 分）

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

二、不定项选择题（共5 题，每题1.5 分，共计7.5 分，多选或少选均不得分）

1	2	3	4	5
ABC	ABD	ABCD	AD	AC

三、问题求解（共2 题，每题5 分，共计10 分）

1.3/8

2.24960, $h_n = \frac{2+\sqrt{3}}{2\sqrt{3}}(1+\sqrt{3})^n + \frac{-2+\sqrt{3}}{2\sqrt{3}}(1-\sqrt{3})^n$

四、阅读程序写结果（共4题，每题8分，共计32分）

1.6 1 2 3

2.5

3.

1 0000110100101100

2 0110000010

4.10370

五、完善程序（第2题第2 空3 分，其余每空2.5 分，共计28 分）以下各程序填空可能还有一些等价的写法，各省赛区可请本省专家审定和上机验证，可以不上报CCF NOI科学委员会检查。

		Pascal 语言	C++语言	C 语言
1	①	cur-> next		
	②	cur->next		

	③		ReverseList(head->next)
	④		head->next->next
	⑤	NULL或0	
2	①		match[x]
	②		i=e[i].next
	③		match[x]=pre[x]
	④		lca(u,v)
	⑤		match[i]