模拟练习试题参考答案（C++）

为了帮助大家熟悉 CCF 软件能力认证考试的操作方式与答题环境，了解试题的大致难度，做好考前的准备，故在此提供试题的参考答案。C++程序是灵活的，为了解决同一个问题，即使结果相同，程序的内容也不一定是完全一致的，仅供各位在练习时参考。

1. 出现次数最多的数

```
#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <deque>
#include <list>
#include <map>

using namespace std;

int main()
{
	int n;
	cin >> n;
	map<int, int> f;
	for (int i = 0; i < n; i++)
	{
		int t;
		cin >> t;
		f[t]++;
	}
	int ans, m = 0;
	for (map<int, int>::iterator it = f.begin(); it != f.end(); it++)
	{
		if (it->second > m)
		{
			m = it->second;
			ans = it->first;
		}
```

```
	}
	cout << ans << endl;
	return 0;

}
```

2. ISBN 号码

```
#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <deque>
#include <list>
#include <map>

using namespace std;

int a[10];

int main()
{
	string s;
	cin >> s;
	a[0] = s[0] - '0';
	a[1] = s[2] - '0';
	a[2] = s[3] - '0';
	a[3] = s[4] - '0';
	a[4] = s[6] - '0';
	a[5] = s[7] - '0';
	a[6] = s[8] - '0';
	a[7] = s[9] - '0';
	a[8] = s[10] - '0';
	a[9] = s[12] - '0';
	int sum = 0;
	for (int i = 0, j = 1; i < 9; i++, j++)
	{
		sum += a[i] * j;
	}
	int code = sum % 11;
	char c = code == 10 ? 'X' : '0' + code;
	if (s[12] == c)
	{
```

```
            cout << "Right" << endl;
        }else
        {
            s[12] = c;
            cout << s << endl;
        }
        return 0;
}
```

3. 最大的矩形

```
#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <deque>
#include <list>

using namespace std;

int main()
{
        int n;
        vector<int> a;
        cin >> n;
        for (int i = 0; i < n; i++)
        {
            int x;
            cin >> x;
            a.push_back(x);
        }
        int ans = 0;
        for (int i = 0; i < n; i++)
        {
            int h = a[i];
            for (int j = i; j < n; j++)
            {
                if (a[j] < h)
                    h = a[j];
                int s = (j - i + 1) * h;
                if (ans < s)
                    ans = s;
            }
```

```
    }
    cout << ans << endl;
    return 0;
}
```

4. 有趣的数

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <string>
#include <vector>
#include <deque>
#include <list>

using namespace std;

long long f[2000][3][2]; // f[seq_k to place][0: to place 0 , 1: ethier 0 or 1, 2 : must be 1][3 is placed ? 1 : 0]

int dp(int n, int p1, int p3)
{
    long long &now = f[n][p1][p3];
    if (now != -1)
        return now;
    if (n == 0)
    {
        if (p1 == 2 && p3 == 1)
        {
            now = 1;
        }else
        {
            now = 0;
        }
        return now;
    }
    now = 0;
    if (p1 == 0)
    {
        now +=    dp(n-1, 1, p3); // go 0
    }else if (p1 == 1)
    {
        now += dp(n-1, 1, p3); // go 0
```

```
            now += dp(n-1, 2, p3); // go 1
        }else // p1 == 2
        {
            now += dp(n-1, 2, p3); // go 1
        }

        if (p3 == 0)
        {
            now += dp(n-1, p1, p3); // go 2;
            now += dp(n-1, p1, 1); // go 3;
        }else
        {
            now += dp(n-1, p1, 1); // go 3;
        }
        now %= 1000000007;
}

int main()
{
        int n;
        cin >> n;
        memset(f, -1, sizeof(f));
        int ans = dp(n - 1, 0, 0); // seq[n] is 2
        cout << ans << endl;
        return 0;
}
```

5. I’m stuck!

```
#include <iostream>
#include <cstdio>
#include <string>
#include <vector>
#include <deque>
#include <cstring>
#include <list>

using namespace std;

//

class Move
{
```

```
	public:
		virtual bool CanMove(char from, char to, int dx, int dy) = 0;
};

class ForwardMove : public Move
{
	public:
		virtual bool CanMove(char from, char to, int dx, int dy)
		{
			if (to == '#') return false;
			switch (from)
			{
				case '+' : case 'S' : case 'T' : return true; break;
				case '-' : return dy != 0; break;
				case '|' : return dx != 0; break;
				case '.' : return dx == 1; break;
			}
			return false;
		}
};

class BackwardMove : public Move
{
	public:
		virtual bool CanMove(char from, char to, int dx, int dy)
		{
			if (to == '#') return false;
			switch (to)
			{
				case '+' : case 'S' : case 'T' : return true; break;
				case '-' : return dy != 0; break;
				case '|' : return dx != 0; break;
				case '.' : return dx == -1; break;
			}
			return false;
		}
};

char s[100][100];

typedef bool ARR[100][100];
ARR bs, bt;
int sx, sy, tx, ty;
```

```
int d[4][2] = {{-1, 0},{1, 0},{0, 1},{0, -1}};

void Bfs(ARR b, Move *move, int x, int y)
{
	if (b[x][y])
		return;
	b[x][y] = true;
	for (int o = 0; o < 4; o++)
	{
		int dx = d[o][0];
		int dy = d[o][1];
		int xx = x + dx;
		int yy = y + dy;
		if (move->CanMove(s[x][y], s[xx][yy], dx, dy))
		{
			Bfs(b, move, xx, yy);
		}
	}
}

int n, m;

int main()
{
	cin >> n >> m;

	for (int i = 0; i <= n + 1; i++)
	for (int j = 0; j <= m + 1; j++)
		s[i][j] = '#';

	for (int i = 1; i <= n; i++)
		cin >> s[i]+1;

	for (int i = 0; i <= n + 1; i++)
		s[i][m + 1] = '#';


	for (int i = 0; i <= n + 1; i++)
	{
		for (int j = 0; j <= m + 1; j++)
		{
			if (s[i][j] == 'S')
			{
				sx = i;
```

```
                    sy = j;
                }
                if (s[i][j] == 'T')
                {
                    tx = i;
                    ty = j;
                }

            }
        }

        Bfs(bs, new ForwardMove(), sx, sy);
        Bfs(bt, new BackwardMove(), tx, ty);

        int ans = 0;
        for (int i = 0; i <= n + 1; i++)
        {
            for (int j = 0; j <= m + 1; j++)
            {
                if (bs[i][j] && ! bt[i][j])
                    ans ++;
            }
        }

        if (bs[tx][ty] == false)
            cout << "I'm stuck!" << endl;
        else
            cout << ans << endl;
        return 0;
}
```