

7-1 找鞍点

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;
5      int a[10][10] = {0};
6      int max_line[10] = {0}, min_row[10] = {0};
7      scanf("%d", &n);
8      for (int i = 0; i < n; i++)
9          for (int j = 0; j < n; j++)
10             scanf("%d", &a[i][j]);
11     // 行最大值和列最小值可以同时查找
12     for (int i = 0; i < n; i++)
13     {
14         max_line[i] = a[i][0];
15         min_row[i] = a[0][i];
16         for (int j = 0; j < n; j++)
17         {
18             if (max_line[i] < a[i][j])
19                 max_line[i] = a[i][j];
20             if (min_row[i] > a[j][i])
21                 min_row[i] = a[j][i];
22         }
23     }
24     for (int i = 0; i < n; i++)
25         for (int j = 0; j < n; j++)
26             if (min_row[j] == a[i][j] && max_line[i] == a[i][j])
27             {
28                 printf("%d %d\n", i, j);
29                 return 0;
30             }
31     printf("NONE");
32     return 0;
33 }
```

7-2 螺旋方阵

每次大循环填充一层数字共 $4(n - 1 - 2i)$ 个，总共循环 $\frac{n}{2}$ 次

$$\sum_{i=0}^{\frac{n}{2}-1} 4(n - 1 - 2i) = n^2$$

```
1  #include <stdio.h>
2  int main()
3  {
4      int i, j, k, n, a[15][15];
5      scanf("%d", &n);
6      k = 1;
7      for (i = 0; i < n / 2; i++) //按螺旋方阵的层数循环， n 为单数时 最后一层单独输出
8      {
9          for (j = i; j < n - 1 - i; j++)
10             a[i][j] = k++;
```

```

11     for (j = i; j < n - 1 - i; j++)
12         a[j][n - 1 - i] = k++;
13     for (j = n - i - 1; j > i; j--)
14         a[n - i - 1][j] = k++;
15     for (j = n - i - 1; j > i; j--)
16         a[j][i] = k++;
17 }
18 if (n % 2 == 1)
19     a[n / 2][n / 2] = k;
20 for (i = 0; i < n; i++)
21 {
22     for (j = 0; j < n; j++)
23         printf("%3d", a[i][j]);
24     cout << endl;
25 }
26 return 0;
27 }

```

7-3 IP地址转换

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int main()
4  {
5      string s;
6      cin >> s;
7      for (int i = 0; i < 4; i++)
8      {
9          int sum = 0;
10         for (int j = 0; j < 8; j++)
11             sum += (s[i * 8 + j] - 48) << (7 - j);
12         printf("%d%c", sum, ".\n"[i==3]);
13     }
14     return 0;
15 }

```

7-4 二分法求多项式单根

```

1  #include <stdio.h>
2  #include <math.h>
3  float a0, a1, a2, a3;
4  float f(float x) { return a3 * x * x * x + a2 * x * x + a1 * x + a0; }
5  int main()
6  {
7      float a, b;
8      scanf("%f %f %f %f", &a3, &a2, &a1, &a0);
9      scanf("%f %f", &a, &b);
10     float mid;
11     while (b - a > 0.0001)
12     {
13         if (f(a) == 0)
14         {
15             printf("%.2f", a);
16             return 0;
17         }
18         if (f(b) == 0)

```

```

19     {
20         printf("%.2f", b);
21         return 0;
22     }
23     mid = (a + b) / 2;
24     if (f(mid) * f(a) > 0)
25     {
26         a = mid;
27     }
28     else
29     {
30         b = mid;
31     }
32 }
33 printf("%.2f", mid);
34 return 0;
35 }

```

7-5 猴子选大王

```

1  #include <stdio.h>
2  int main()
3  {
4      int N, i, count = 0, k = 0, flag = 0;
5      scanf("%d", &N);
6      int n[N];
7      for (i = 0; i < N; i++)
8          n[i] = 1;
9      while (k != N - 1)
10         for (i = 0; i < N; i++)
11             if (n[i] == 1)
12             {
13                 flag = i;
14                 count++;
15                 if (count == 3)
16                 {
17                     n[i] = 0;
18                     k++;
19                     count = 0;
20                 }
21             }
22         printf("%d", flag + 1);
23         return 0;
24     }

```

7-6 N个数求和

```

1  #include <stdio.h>
2  #include <math.h>
3  using namespace std;
4  int gcd(int a, int b) { return b ? gcd(b, a % b) : a; }
5  int main()
6  {
7      int n;
8      scanf("%d", &n);
9      int a, b;

```

```

10     int p = 0, q = 1; // 总和为 p/q
11     for (int i = 0; i < n; i++)
12     {
13         scanf("%d/%d", &a, &b);
14         p = p * b + q * a;
15         q = q * b;
16         int r = gcd(abs(p), abs(q));
17         p /= r;
18         q /= r;
19     }
20     if (q == 1)
21         printf("%d", p);
22     else if (p > q)
23         printf("%d %d/%d", p / q, p % q, q);
24     else
25         printf("%d/%d", p, q);
26     return 0;
27 }

```

7-7 整数分解为若干项之和

```

1  #include <iostream>
2  #include <vector>
3  using namespace std;
4  int k = 0, n;
5  void dfs(int x, vector<int> ans, int st)
6  {
7      if (x == 0)
8      {
9          cout << n << "=";
10         for (int i = 0, l = ans.size(); i < l; i++)
11             cout << ans[i] << (string[2]){ "+", ""}[i == l - 1];
12         cout << ";\n"[++k % 4 == 0 || ans[0] == n];
13         return;
14     }
15     for (int i = st; i <= x; i++)
16     {
17         ans.push_back(i);
18         dfs(x - i, ans, i);
19         ans.pop_back();
20     }
21 }
22 int main()
23 {
24     cin >> n;
25     dfs(n, {}, 1);
26     return 0;
27 }

```

7-8 输出全排列

```

1  #include<iostream>
2  #include<vector>
3  using namespace std;
4  int n;
5  vector<int> num(10);

```

```

6  vector<int> arr(10);
7  void dfs(int step) {
8      int i;
9      if (step == n + 1) {
10         for (int i = 1; i <= n; i++) {
11             cout<<num[i];
12         }
13         cout<<endl;
14         return;
15     }
16     for (i = 1; i <= n; i++) {
17         if (arr[i] == 0) {
18             num[step] = i;
19             arr[i] = 1;
20             dfs(step+1);
21             arr[i] = 0;
22         }
23     }
24 }
25 int main() {
26     cin>>n;
27     dfs(1);
28 }

```

使用c++ STL函数 `std::next_permutation`

```

1  #include <iostream>
2  #include <algorithm>
3  using namespace std;
4  int main()
5  {
6      int a[10], n;
7      cin >> n;
8      for (int i = 0; i < n; i++)
9          a[i] = i + 1;
10     do
11     {
12         for (int i = 0; i < n; i++)
13             cout << a[i];
14         cout << endl;
15     } while (next_permutation(a, a + n));
16     return 0;
17 }

```

7-9 出栈序列的合法性

```

1  #include <bits/stdc++.h>
2  using namespace std;
3  int main()
4  {
5      int m, n, k;
6      cin >> m >> n >> k;
7      while (k--)
8      {
9          int a[n];
10         stack<int> s;

```

```

11     int x = 1;
12     bool is_right = true;
13     for (int j = 0; j < n; j++)
14         cin >> a[j];
15     for (int j = 0; j < n;)
16     {
17         while (x <= a[j])
18         {
19             s.push(x++);
20             if (s.size() > m)
21                 is_right = false;
22         }
23         if (s.top() != a[j])
24         {
25             is_right = false;
26             break;
27         }
28         else
29         {
30             s.pop();
31             j++;
32         }
33     }
34     cout << (is_right ? "YES" : "NO") << endl;
35 }
36 return 0;
37 }

```

7-10 包装机

```

1  #include <iostream>
2  #include <stack>
3  #include <queue>
4  using namespace std;
5  const int MAX = 100 + 5;
6  int main()
7  {
8      stack<char> s;
9      queue<char> q[MAX];
10     string t;
11     int n, m, s_max, op;
12     cin >> n >> m >> s_max;
13     cin.get();
14     for (int i = 1; i <= n; i++)
15     {
16         getline(cin, t);
17         for (char c : t)
18             q[i].push(c);
19     }
20     while (cin >> op && op != -1)
21     {
22         if (op == 0)
23         {
24             if (!s.empty())
25             {
26                 cout << s.top();
27                 s.pop();

```

```
28         }
29     }
30     else if (!q[op].empty())
31     {
32         if (s.size() >= s_max)
33         {
34             cout << s.top();
35             s.pop();
36         }
37         s.push(q[op].front());
38         q[op].pop();
39     }
40 }
41 return 0;
42 }
```