PTA 数组 E

rogeryoungh

2021 年 06 月 20 日

目录

PTA 数组 Easy 部分，PDF。
这次的问题都比较简单。

1 7-1 密码强度

```
ll f[10];
char s[100];

int main() {
    scanf("%s", s + 1);
    ll len = strlen(s + 1);
    for (ll i = 1; i <= len; i++) {
        char c = s[i];
        if (c >= '0' && c <= '9') {
            f[1]++;
        } else if (c >= 'a' && c <= 'z') {
            f[2]++;
        } else if (c >= 'A' && c <= 'Z') {
            f[3]++;
        }
    }
    ll x = (f[1] > 0) + (f[2] > 0) + (f[3] > 0) + (len > 8);
    printf("%lld", x);
    return 0;
}
```

2 7-2 统计不同数字字符出现次数

```
ll f[20];
char s[500];

int main() {
    gets(s);
    ll len = strlen(s);
    for (ll i = 0; i <= len - 1; i++) {
        char c = s[i];
        if (c >= '0' && c <= '9') {
            f[c - '0']++;
        }
    }
    ll flag = 0;
    for (ll i = 0; i <= 9; i++) {
        if (f[i] > 0) {
            flag = 1;
            printf("%lld-%lld\n", i, f[i]);
```

```
        }
    }
    if (!flag) {
        printf("None!\n");
    }
    return 0;
}
```

3 7-3 最多的字母

```
ll f[50];
char s[100];

int main() {
    scanf("%s", s);
    ll len = strlen(s);
    memset(f, 0, sizeof(f));
    for (ll i = 0; i <= len - 1; i++) {
        ll c = s[i];
        if (c >= 'a' && c <= 'z') {
            f[c - 'a']++;
        }
    }
    ll maxi = 0;
    for (ll i = 0; i <= 25; i++) {
        if (f[i] > f[maxi]) {
            maxi = i;
        }
    }
    printf("%c,%lld\n", maxi + 'a', f[maxi]);
    return 0;
}
```

4 7-4 成绩统计分析表 (*)

```
ll ae[10];

void printline(double d) {
    ll ld = d + 0.5;
    printf("%5.1lf ", d);
    _fora (i, 1, ld)
```

```
7           putchar('*');
8       putchar('\n');
9   }
10
11  void prline2(double d) {
12      ll ld = d + 0.5;
13      printf("%5.1lf%% ", d);
14      _fora (i, 1, ld)
15          putchar('*');
16      putchar('\n');
17  }
18
19  int main() {
20      ll n = rr();
21      double max = 0, min = 101, sum = 0;
22      for (ll i = 1; i <= n; i++) {
23          double t;
24          scanf("%lf", &t);
25          ll lt = (ll)t;
26          printf("%03lld: ", i);
27          printline(t);
28
29          max = t > max ? t : max;
30          min = t < min ? t : min;
31          sum += t;
32          lt = lt / 10 - 5;
33          if (lt < 0)
34              lt = 0;
35          ae[lt]++;
36      }
37      putchar('\n');
38
39      printf("Max: ");
40      printline(max);
41      printf("Min: ");
42      printline(min);
43      printf("Avg: ");
44      printline(sum / n);
45      putchar('\n');
46
47      printf("A: ");
48      prline2((ae[4] + ae[5]) * 100.0 / n);
```

```
    printf("B: ");
    prline2(ae[3] * 100.0 / n);
    printf("C: ");
    prline2(ae[2] * 100.0 / n);
    printf("D: ");
    prline2(ae[1] * 100.0 / n);
    printf("E: ");
    prline2(ae[0] * 100.0 / n);
    return 0;
}
```

5 7-5 找鞍点

```
ll max_x[10], max_y[10];
ll mtx[5][5];

int main() {
    for (ll i = 0; i <= 3; i++)
        for (ll j = 0; j <= 3; j++)
            mtx[i][j] = rr();

    for (ll i = 0; i <= 3; i++) {
        ll my = 0, mx = 0;
        _fora (j, 0, 3) {
            if (mtx[i][j] > mtx[i][my])
                my = j;
            if (mtx[j][i] < mtx[mx][i])
                mx = j;
        }
        max_x[i] = mx;
        max_y[i] = my;
    }
    ll flag = 0;
    for (ll i = 0; i <= 3; i++) {
        for (ll j = 0; j <= 3; j++) {
            if (max_x[j] == i && max_y[i] == j) {
                flag++;
                printf("a[%lld][%lld]=%lld\n", i, j, mtx[i][j]);
            }
        }
    }
    if (!flag) {
```

```
30          printf("It is not exist!\n");
31      }
32      return 0;
33  }
```

6 7-6 折半查找

```
1   ll nn[100086];
2
3   ll lower_bound(ll l, ll r, ll val) {
4       while (l < r) {
5           ll mid = (l + r) >> 1;
6           if (nn[mid] >= val)
7               r = mid;
8           else
9               l = mid + 1;
10      }
11      return l;
12  }
13
14  int main() {
15      ll n = rr();
16      for (ll i = 1; i <= n; i++)
17          nn[i] = rr();
18      ll t = rr();
19      ll i = lower_bound(1, n, t);
20      if (nn[i] == t)
21          printf("It's position is %lld!\n", i);
22      else
23          printf("No data!\n");
24      return 0;
25  }
```

7 7-7 字符串转换为整数

```
1   ll read() {
2       ll s = 0;
3       int c;
4       while ((c = getchar()) != EOF) {
5           if (c >= '0' && c <= '9')
6               s = s * 10 + c - '0';
```

```
    }
    return s;
}

int main() {
    printf("%lld", read());
    return 0;
}
```

8 7-8 去掉多余空格

```
char s[1000086];

int main() {
    gets(s);
    ll len = strlen(s);
    if (len > 30) {
        s[30] = 0;
        len = 30;
    }
    printf("[%s]\n", s);
    ll l, r;
    for (ll i = 0; i <= len - 1; i++) {
        if (s[i] != ' ') {
            l = i;
            break;
        }
    }
    for (ll i = len - 1; i >= 0; i--) {
        if (s[i] != ' ') {
            r = i;
            break;
        }
    }
    s[r + 1] = 0;
    printf("[%s]\n", s + l);
    return 0;
}
```

9 7-9 矩阵对角线求和

为什么要把矩阵存下来才计算呢?

```
int main() {
    ll n = rr();
    ll sum = rr();
    for (ll i = 1; i <= n - 1; i++) {
        for (ll j = 1; j <= n; j++)
            rr();
        sum += rr();
    }
    printf("%lld\n", sum);
    return 0;
}
```

10 7-10 倒置字符串并输出

```
char s[1000086];

int main() {
    gets(s);
    ll len = strlen(s) - 1;
    for (ll i = len; i >= 0; i--)
        putchar(s[i]);
    putchar('\n');
    putchar(s[len]);
    return 0;
}
```

11 7-11 去掉最大值和最小值

```
int main() {
    ll sum = 0;
    ll min = 101, max = 0;
    for (int i = 1; i <= 10; i++) {
        ll t = rr();
        max = max < t ? t : max;
        min = min > t ? t : min;
        sum += t;
    }
    ll score = sum - min - max;
```

```
    printf("%lld", score);
    return 0;
}
```

12 7-12 有重复的数据

```
ll nn[100086];

void quick_sort(ll *nn, ll l, ll r) {
    if (l >= r)
        return;
    int i = l, j = r;
    int x = nn[(l + r) / 2];
    while (i <= j) {
        while (nn[j] > x)
            j--;
        while (nn[i] < x)
            i++;
        if (i <= j) {
            ll t = nn[i];
            nn[i] = nn[j];
            nn[j] = t;
            i++;
            j--;
        }
    }
    quick_sort(nn, l, j);
    quick_sort(nn, i, r);
}

int main() {
    ll n = rr();
    memset(nn, 0, sizeof(nn));
    _fora (i, 1, n) { nn[i] = rr(); }
    quick_sort(nn, 1, n);
    int flag = 0;
    for (ll i = 2; i <= n; i++)
        flag += nn[i] == nn[i - 1];
    if (flag)
        printf("YES");
    else
        printf("NO");
```

```
    return 0;
}
```

13 7-13 判断题

```
ll f[100], ans[100];

int main() {
    ll n = rr(), m = rr();
    for (ll i = 1; i <= m; i++) {
        f[i] = rr();
    }
    for (ll i = 1; i <= m; i++) {
        ans[i] = rr();
    }
    for (ll i = 1; i <= n; i++) {
        ll sum = 0;
        for (ll j = 1; j <= m; j++) {
            if (rr() == ans[j])
                sum += f[j];
        }
        printf("%lld\n", sum);
    }
    return 0;
}
```

14 7-14 统计单词数

```
int isAlpha(int x) {
    if ('A' <= x && x <= 'Z')
        return 1;
    else if ('a' <= x && x <= 'z')
        return 1;
    return 0;
}

char s[1000086];

int main() {
    gets(s);
    int len = strlen(s);
```

```
    int p = isAlpha(s[0]);
    int sum = 0;
    for (int i = 1; i <= len; i++) {
        int t = isAlpha(s[i]);
        if (p && !t)
            sum++;
        p = t;
    }
    printf("%d", sum);
    return 0;
}
```

15 7-15 删除某字符

```
char s[1000086];

int main() {
    gets(s);
    char t;
    scanf("%c", &t);
    ll len = strlen(s) - 1;
    for (ll i = 0; i <= len; i++) {
        if (s[i] != t)
            putchar(s[i]);
    }
    return 0;
}
```

16 7-16 自守数

为什么要老老实实算呢，直接打表输出。

```
ll nn[] = {
    0,       1,       5,        6,        25,        76,
    376,     625,     9376,     90625,    109376,    890625,
    2890625, 7109376, 12890625, 87109376, 212890625, 787109376,
};

ll lower_bound(ll l, ll r, ll val) {
    while (l < r) {
        ll mid = (l + r) >> 1;
        if (nn[mid] > val)
```

```
            r = mid;
        else
            l = mid + 1;
    }
    return l;
}

int main() {
    ll n = rr();
    for (ll i = 1; i <= n; i++) {
        if (i != 1)
            putchar(' ');
        printf("%lld", lower_bound(0, 17, rr()));
    }
    return 0;
}
```