

2020C程序设计

期中考试

Problem A: 今年是2020年的第几天

题目描述

输入一个日期，用三个整数分别表示年、月、日。这三个整数都是合法的日期表示，且在公历2020年之内。计算出输入的日期是2020年的第几天。

输入

输入三个整数，分别表示年月日。

输出

输出一个整数，表示是2020年的第几天。

样例输入

```
2020 1 1
2020 7 11
2020 12 31
```

样例输出

```
1
193
366
```

```
1  #include <stdio.h>
2  int main()
3  {
4      int year, month, day;
5      int days[] = { 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
6      int sum, i;
7      while (scanf("%d %d %d", &year, &month, &day) != EOF)
8      {
9          sum = 0;
10         if (month == 1)
11         {
12             sum = day;
13         }
14         else
15         {
16             for (i = 0; i < month - 1; i++)
17                 sum += days[i];
18             sum += day;
19         }
20         printf("%d\n", sum);
21     }
22     return 0;
23 }
```

Problem B: 山谷数

题目描述

在前几次作业中，同学们已经学会如何判断平方数和质数。今天我们来认识一种全新的数：山谷数。

当一个数字，从左到右依次看过去数字没有出现先递增接着递减的“山峰”现象，就称为“山谷数”。它可以递增，也可以递减，还可以先递减再递增。在递增或递减的过程中可以出现相等的情况。

比如，1，10，12，212，111，112，32122都是山谷数。121，12331，21212则不是。

输入

输入整数 n ， $0 \leq n \leq 2^{31} - 1$

输出

如果整数 n 是山谷数，输出Yes，否则输出No

样例输入

```
1
112
121
```

样例输出

```
Yes
Yes
No
```

```

1 #include <stdio.h>
2 #include <string.h>
3
4 int main()
5 {
6     char str[20];
7     int len, i;
8     char pre, cur;
9     int flag, find;
10
11     while (scanf("%s", str) != EOF)
12     {
13         flag = 0;
14         find = 0;
15         len = strlen(str);
16
17         pre = str[0];
18         for (i = 1; i < len; i++)
19         {
20             cur = str[i];
21             if (cur > pre)
22                 flag = 1;

```

```

23         else if (cur < pre)
24         {
25             if (flag == 1)
26             {
27                 find = 1;
28                 break;
29             }
30         }
31         pre = cur;
32     }
33
34     if (find == 1)
35         printf("No\n");
36     else
37         printf("Yes\n");
38 }
39 return 0;
40 }

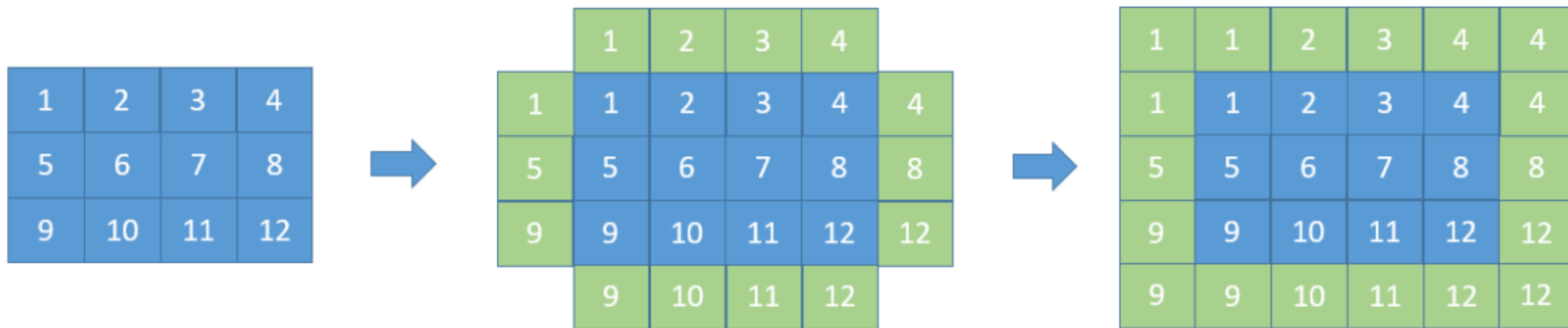
```

Problem C: 矩阵扩展

题目描述

通过本课程的学习，我们对矩阵越来越熟悉。

给定一个 $m \times n$ 的矩阵，拷贝矩阵最外围一层元素，将原矩阵扩展为一个新的矩阵，如下图所示。扩展后的矩阵大小为 $(m+2) \times (n+2)$ 。



```

1 #include<stdio.h>
2
3 int main()
4 {
5     int a[30][30], b[30][30];
6     int m, n, i, j;
7
8     while (scanf("%d %d", &m, &n) != EOF)
9     {
10         for (i = 0; i < m; i++)
11         {
12             for (j = 0; j < n; j++)
13             {
14                 scanf("%d", &a[i][j]);
15                 b[i + 1][j + 1] = a[i][j];
16             }
17         }
18
19         for (i = 1; i <= m; i++)
20         {
21             b[i][0] = b[i][1];
22             b[i][n + 1] = b[i][n];
23         }
24

```

展

```

25         for (j = 0; j <= n + 1; j++)
26         {
27             b[0][j] = b[1][j];
28             b[m + 1][j] = b[m][j];
29         }
30
31         for (i = 0; i < m + 2; i++)
32         {
33             for (j = 0; j < n + 2; j++)
34             {
35                 printf("%d ", b[i][j]);
36             }
37             printf("\n");
38         }
39     }
40
41     return 0;
42 }

```

Problem D: 统计字符的出现次数

题目描述

输入字符串，字符串由英文字母、数字、标点符号和运算符组成（不含空格）。

统计每个字符在该字符串中出现次数，按出现次数从大到小的顺序，将字符和其出现次数输出。

如果存在出现次数相等的字符，ASCII码值较大的字符排在前面。

输入

输入一个字符串，字符串长度最多为1000。

输出

按字符出现次数的排序，输出字符和其出现次数，两者之间用冒号隔开。

样例输入

```
HelloWorld!  
XMU:(1+2)*9=27
```

样例输出

```
l:3 o:2 r:1 e:1 d:1 W:1 H:1 !:1  
2:2 X:1 U:1 M:1 =:1 ::1 9:1 7:1 1:1 +:1 *:1 ):1 (:1
```



```
1  #include <stdio.h>
2  #include <string.h>
3
4  int main()
5  {
6      char s[1000];
7      int count[256], i, j;
8      int ascii[256];
9      int t1, t2;
10
11     while (scanf("%s", s) != EOF)
12     {
13         for (i = 0; i < 256; i++)
14         {
15             count[i] = 0;
16             ascii[i] = i;
17         }
18         for (i = 0; i < strlen(s); i++)
19         {
20             count[s[i]] ++;
21         }
22     }
```

Problem D: 统计字符的出现次数

```
23 // sort
24 for (j = 0; j < 255; j++)
25 {
26     for (i = 0; i < 255 - j; i++)
27     {
28         if (count[i] < count[i + 1])
29         {
30             t1 = count[i]; count[i] = count[i + 1]; count[i + 1] = t1;
31             t2 = ascii[i]; ascii[i] = ascii[i + 1]; ascii[i + 1] = t2;
32         }
33         else if (count[i] == count[i + 1])
34         {
35             if (ascii[i] < ascii[i + 1])
36             {
37                 t2 = ascii[i]; ascii[i] = ascii[i + 1]; ascii[i + 1] = t2;
38             }
39         }
40     }
41 }
```

Problem D: 统计字符的出现次数

```
43         // output
44         for (i = 0; i < 256; i++)
45         {
46             if (count[i] == 0)
47                 break;
48             printf("%c:%d ", ascii[i], count[i]);
49         }
50         printf("\n");
51     }
52
53     return 0;
54 }
```

Problem E: 厦大GPA

题目描述

厦门大学的GPA（绩点）计算规则一直是同学们非常关心的问题。每门考试成绩为百分制，则分数与绩点对应关系如下：

90~100 4.0

85~89 3.7

81~84 3.3

78~80 3.0

75~77 2.7

72~74 2.3

68~71 2.0

64~67 1.7

60~63 1.0

0~59 0.0

某位同学一共参加了 4 门考试，给定四门考试的总分，请问在最优情况下，4门考试绩点的和最高是多少？

输入

输入4门考试的总分 n ， $0 \leq n \leq 400$

输出

输出最优情况下，4门考试绩点之和的最高值。结果保留一位小数。

Problem E: 厦大GPA

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int x, n;
6      int low[11] = { 0, 60, 64, 68, 72, 75, 78, 81, 85, 90 };
7      int up[11] = { 59, 63, 67, 71, 74, 77, 80, 84, 89, 100 };
8      float gpa[11] = { 0, 1.0, 1.7, 2.0, 2.3, 2.7, 3.0, 3.3, 3.7, 4.0 };
9      int i, j, k, m, t;
10     float max_gpa;
11
```

Problem E: 厦大GPA

```
12 while (scanf("%d", &n) != EOF)
13 {
14     max_gpa = 0.0;
15     for (i = 0; i < 10; i++)
16         for (j = 0; j < 10; j++)
17             for (k = 0; k < 10; k++)
18                 for (m = 0; m < 10; m++)
19                     {
20                         if ((low[i] + low[j] + low[k] + low[m]) <= n
21                             && n <= (up[i] + up[j] + up[k] + up[m]))
22                             {
23                                 if (max_gpa < (gpa[i] + gpa[j] + gpa[k] + gpa[m]))
24                                 {
25                                     max_gpa = gpa[i] + gpa[j] + gpa[k] + gpa[m];
26                                 }
27                             }
28                     }
29
30     printf("%.1f\n", max_gpa);
31 }
32 return 0;
33 }
```

Problem F: 三角形

题目描述

三角形需满足两边之和大于第三边。

假设有N根木棒，现在要挑选其中的三根，问能拼出的三角形的最大周长是多少？

如果不能拼成任何一个三角形，输出-1。

输入

先输入正整数N，表示木棒的个数， $1 \leq N \leq 1000$ 。再输入N个数，表示每一根木棒的长度。木棒的长度都是不超过100000的正整数。

输出

输出能拼出三角形的最大周长。如果不能拼成三角形，则输出-1。

样例输入

```
3 1 1 100
7 1 9 9 90 2 2 4
```

样例输出

```
-1
22
```

Problem F: 三角形

```
1  #include <stdio.h>
2
3  void sort(int a[], int n)
4  {
5      int i, j, tmp;
6      for (j = 0; j < n - 1; j++)
7      {
8          for (i = 0; i < n - 1 - j; i++)
9          {
10             if (a[i] > a[i + 1])
11             {
12                 tmp = a[i]; a[i] = a[i + 1]; a[i + 1] = tmp;
13             }
14         }
15     }
16     return;
17 }
```



```

19 int main()
20 {
21     int n, i, f;
22     int a[1005];
23     while (scanf("%d", &n) != EOF)
24     {
25         f = 0;
26         for (i = 0; i < n; i++)
27         {
28             scanf("%d", &a[i]);
29         }
30         sort(a, n);
31         for (i = n - 1; i >= 2; i--)
32         {
33             if (a[i - 1] + a[i - 2] > a[i])
34             {
35                 f = 1;
36                 printf("%d\n", a[i - 1] + a[i - 2] + a[i]);
37                 break;
38             }
39         }
40         if (f == 0)    printf("-1\n");
41     }
42     return 0;
43 }

```