

第六章 字符数组与字符串

第 7 4 课 `memset` 函数和 `fill` 函数

《信息学奥赛一本通·编程启蒙 C++版》

我们可以用 `memset` 函数按字节对内存块进行初始化。

```
void *memset(void *s, int c, size_t n);
```

- `s` 指向要填充的内存块。
- `c` 是要被设置的值。
- `n` 是要被设置该值的字符数。
- 返回类型是一个指向存储区 `s` 的指针。

由于 `memset` 函数是按字节对内存块进行初始化的，因此只能将 `int` 数组初始化为 0 和 -1，因为 `int` 类型占 4 个字节，例如：

```
int x; memset(x, 1, sizeof(x));
```

`x` 的值为 $16843009 = (000000010000000010000000100000001)_2$ ，而不是 1。

二、fill 函数

如果要给容器填充指定更加灵活的元素，可以使用 fill 函数。

```
fill(iterator beg, iterator end, value);
```

- 向容器中填充元素
- beg 开始迭代器
- end 结束迭代器
- value 填充的值

如果要往二维数组里全部填充上 1，就可以使用如下代码：

```
int a[5][5]; fill(a[0], a[0]+sizeof(a), 1);
```

【例 74.1】 倒杨辉三角

【题目描述】

杨辉三角是如下三角形：

```

          1
        1 1
      1 2 1
    1 3 3 1
  1 4 6 4 1
1 5 10 10 5 1
  6 15 20 15 6 1
    .....

```

可以发现杨辉三角的每个位置的值是左上角的值加上右上角的值，请输出倒杨辉三角。

【输入格式】

输入一行一个整数 $n(1 \leq n \leq 5)$ ，表示选择的行数。

【输出格式】

输出 n 行倒的杨辉三角（每一行相邻的数之间用一个空格隔开）。

【样例输入】

4

【样例输出】

```

1 3 3 1
 1 2 1
   1 1
    1

```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int r,c,k,n,ny,sum;
4. int a[11][11];
5. int main(){
6.     cin>>n;
7.     for (int i=1;i<=n;i++){
8.         for (int j=1;j<=i;j++){
9.             if(j==1||j==i) a[i][j]=1;
10.            else a[i][j]= a[i-1][j-1]+a[i-1][j];
11.        }
12.    }
13.    for (int i=n;i>=1;i--){
14.        for (int k=1;k<=n-i;k++) cout<<' ';
15.        for (int j=1;j<=i;j++){
16.            cout<<a[i][j]<<' ';
17.        }
18.        cout<<endl;
19.    }
20.    return 0;
21.}
```

【例 74.2】 古风排版

【题目描述】

中国的古人写文字，是从右向左竖向排版的。本题就请你编写程序，把一段文字按古风排版。

【输入格式】

输入在第一行给出一个正整数 N ，是每一列的字符数。第二行给出一个长度不超过 **1000** 的非空字符串，以回车结束。 $1 \leq N \leq 1000$ 。

【输出格式】

按古风格式排版给定的字符串，每列 N 个字符（除了最后一列可能不足 N 个）。

【样例输入】

4

This is a test case

【样例输出】

```
asa T
st ih
e tsi
ce s
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n,l,m,p;
4. string s;
5. char a[1005][1005];
6. int main(){
7.     memset(a,' ',sizeof(a));
8.     cin>>m;getchar();
9.     getline(cin,s);
10.    l=s.size();
11.    n=(l-1)/m+1;
12.    for(int i=1;i<=n;i++)
13.        for(int j=1;j<=m;j++){
14.            if(p==l) break;
15.            a[i][j]=s[p];
16.            p++;
17.        }
18.    for(int j=1;j<=m;j++){
19.        for(int i=n;i>=1;i--){
20.            cout<<a[i][j];
21.        }
22.        cout<<endl;
23.    }
24.    return 0;
25.}
```

【例 74.3】 彼得·潘框架

【题目描述】

彼得·潘框架”是一种装饰文字，每一个字母都是由一个菱形框架。一个彼得·潘框架看起来像这样（x 是字母，#是框架）

```
..#..  
.#.#.  
#.X.#  
.#.#.  
..#..
```

然而，只是一个框架会有些沉闷，所以我们每遇到三个字母会把第三个字母用温迪框架把它框起来。温迪框架看起来像这样

```
..*..  
.*.*.  
*.X.*  
.*.*.  
..*..
```

当温迪和彼得·潘的框架重叠时，温迪框架覆盖在上面。

【输入格式】

一行包含至多 15 个大写英文字母。

【输出格式】

输出使用彼得·潘

【样例输入】

DOG

【样例输出】

```
..#...#...*..  
.#.#.#.#.*.*.  
#.D.#.O.*.G.*  
.#.#.#.#.*.*.  
..#...#...*..
```

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. string s;
4. long long len,n,m,
5. dx[8]={-2,-1,0,1,2,1,0,-1},
6. dy[8]={0,1,2,1,0,-1,-2,-1};
7. char a[6][100];
8. int main(){
9.     cin>>s;
10.    len=s.size();
11.    n=5;
12.    m=4*len+1;
13.    fill(a[0],a[0]+sizeof(a) ,'.');
14.    for(int i=1,p=3;i<=len;i++,p+=4){
15.        if(i%3==0)continue;
16.        a[3][p]=s[i-1];
17.        for(int j=0;j<8;j++)
18.            a[3+dx[j]][p+dy[j]]='#';
19.    }
20.    for(int i=3;i<=len;i+=3){
21.        int p=4*i-1;
22.        a[3][p]=s[i-1];
23.        for(int j=0;j<8;j++)
24.            a[3+dx[j]][p+dy[j]]='*';
25.    }
26.    for(int i=1;i<=n;i++){
27.        for(int j=1;j<=m;j++)cout<<a[i][j];
28.        cout<<endl;
29.    }
30.    return 0;
31.}
```

练 74.1 垂直直方图

【题目描述】

输入 4 行全部由大写字母组成的文本，输出一个垂直直方图，给出每个字符出现的次数。注意：只用输出字符的出现次数，不用输出空白字符，数字或者标点符号的输出次数。

【输入格式】

输入包括 4 行由大写字母组成的文本，每行上字符的数目不超过 80 个。

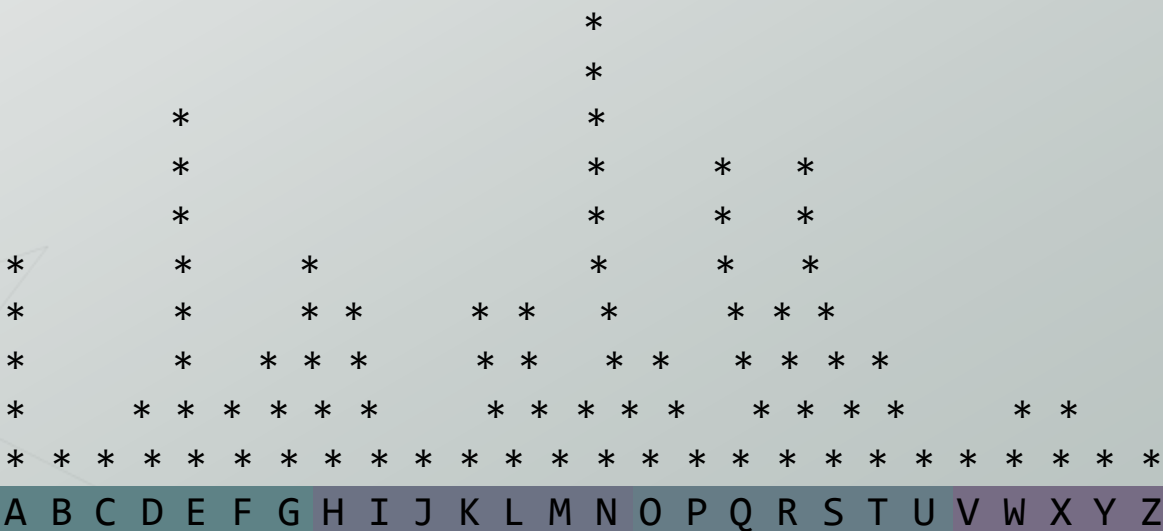
【输出格式】

输出包括若干行。其中最后一行给出 26 个大写英文字母，这些字母之间用一个空格隔开。前面的几行包括空格和星号，每个字母出现几次，就在这个字母的上方输出一个星号。注意：输出的第一行不能是空行。

【样例输入】

THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
THIS IS AN EXAMPLE TO TEST FOR YOUR
HISTOGRAM PROGRAM.
HELLO!

【样例输出】



【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. string s;
4. int a[256],n;
5. int main(){
6.     for(int i=1;i<5;i++){
7.         getline(cin,s);
8.         int len=s.size();
9.         for(int j=0;j<len;j++) if(isupper(s[j])) a[s[j]]++;
10.    }
11.    n=*(max_element(a,a+256));
12.    char b[400][30];
13.    memset(b,' ',sizeof(b));
14.    for(char i='A';i<='Z';i++){
15.        b[n][i-'A']=i;
16.        for(int j=n-1,k=a[i];k>0;k--,j--) b[j][i-'A']='*';
17.    }
18.    for(int i=0;i<=n;i++){
19.        for(int j=0;j<26;j++) cout<<b[i][j]<<" ";
20.        cout<<endl;
21.    }
22.    return 0;
23.}
```

谢谢！

