# 一、字符串、数组作业讲解

因为老师临时有事，不可抗拒因素，调整一下课程，今日课程是免费增加的，我们相会于视频。

我们讲解一下作业。

## 第1题

编写函数repeat(char,n) 实现重复char字符串n次

比如repeat("★",6) 返回"★★★★★★"

**方法1：**

我们现在发现，**只需要进行n次的相同操作：每次str末尾增加1颗★。**

str增加字符，使用连字符，或者用str.concat();

str += “★” 这条语句重复执行n次就可以了。

**方法2：**

**二分法。**

首先我们来看看刚才的算法，有没有什么效率问题？效率低下。

比如我们要制造16颗★，那么我们刚才的算法，要重复执行16次，其实没有必要。

因为★ → ★★ → 直接复制自己 → ★★★★ → 直接复制自己 → ★★★★★★★★ → 16颗星星

也就是说，我们没有必要一次都增加一颗星星，因为我们已经在第二次循环的时候，得到了★★，那为什么不直接复制自己得到4颗呢？？

现在的问题来了，我们的n不一定是2的某次方。比如我们输入16，可以按倍数来。比如用户输入13，怎么办？

其实有办法，因为我们发现，**任何数字都能够直接表示为1、2、4、8、16、32、64……的和**。

比如：

7 = 4 + 2 + 1;

9 = 8 + 1;

**13 = 8 + 4 + 1;**

29 = 16 + 8 + 4 + 1;

任何数字都能拆分为1、2、4、8、16、32、64……的和。

所以我们现在就能想到一种算法，就是把这个星星的数量，就在这个数字拆分的那个数值上，进行复制一倍。

比如我们现在要13颗星星，那么1颗（要） → 2颗（不累加） → 4颗（要） → 8颗（要）

伪代码：

循环开始之前，

结果串str “”

临时串char “★”

数量n

循环：

如果n是奇数，那么str要加上char；如果n是偶数，str不变。 （奇变偶不变）

n = parseInt(n / 2); //n越除越小

char = char + char; //每次循环，char都要倍增。

循环直到n为1截止。

试着遍历一下repeat(“★”,13)

循环开始之前：

str “”

char “★”

n 13

第1次循环：

n此时13是奇数，str就要加上char，所以str变为★

n变为6

char要复制自己一倍，char变为★★

第2次循环：

n此时6是偶数，str不变。str仍为★

n变为3

char要复制自己一倍，char变为★★★★

第3次循环：

n此时3是奇数，str就要加上char，所以str变为★★★★★

n变为1

char要复制自己一倍，★★★★★★★★

第4次循环：

n此时1是奇数，str就要加上char，所以★★★★★★★★★★★★★

n已经是1了，不需要再除了，退出循环，打完收工。

邵老师把这个方法，起了一个别叫好的名字，叫做“1248因数法”、“1248拆分法” 把任何一个数字拆成1、2、4、8、16……和。

## 第2题

在控制台输出图形

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

每本C语言的书上都会有的题目，当然JS也有公司当做面试题。

一共5行，比如行号是i，i从1开始，每行的内容： 5-i个空格 和 2i - 1个星星

重复空格5-i次，重复星星2i-1次。

## 第3题

编写函数rev(char) 实现将字符串char倒置

比如rev("spring") 返回"gnirps"

方法1：

经典面试题，经典解法。

就是for循环从最后一位开始取源字符串的字符，然后拼接到新的串中。

str.charAt(str.length - 1); 就是最后一位

str.charAt(str.length - 2) ;就是倒数第二位

方法2：

以后一看见倒置这个词，就要立即想到数组！

数组有一个东西，reverse()可以立即将数组倒置。

字符串 → 数组 → 倒置 → 字符串

str.split arr.reveer() arr.join(“”)

## **第4题**

将字符串"i come from beijing"倒置，即控制台输出"beijing from come i"

语句直接写在程序中，不需要用户输入

字符串 → 数组 → 倒置 → 字符串

|  |
| --- |
| 1. var str = "i come from beijing"; 2. //字符串 → 数组 3. var arr = str.split(" "); 4. //数组倒置 5. arr = arr.reverse(); 6. //数组 → 字符串 7. str = arr.join(" "); 8. alert(str); |

## 第5题

编写函数rightchange(char,n) 实现字符串char循环右移n位。

比如rightchange("abcdefg",2) 返回"fgabcde"

方法1：

就是大家想到的用数组，因为数组有pop、unshift();

|  |
| --- |
| 1. arr.unshift(arr.pop()) |

把最后一位挪到第1位

方法2：

原来：abcdefg 移动2

变为：fgabcde

fgabcde = fg + abcde

= abcdefg.substr(-2) + abcdefg.substr(0 , length - 2);

移动a 就是 原.substr(-a) + 原.substr(0 , length - a);

## 第6题

编写函数maxr(char) 返回字符串char中最长的连续重复字母

比如maxr("mmmiijjjjkkkkkkssptr") 返回"kkkkkk"

是一个新题型，如果你真的不知道算法，就真的不会做了。

指针法：

和C语言中的指针，不是一个事情。

我们这里的“指针”是虚拟的一个变量而已。

## 第7题

有10个学生的成绩存在数组中，请统计大于等于平均成绩的人数。

成绩直接以数组形式写在程序中，不需要用户输入。

见代码

## 第8题

有10个学生的成绩存在数组中，请统计最高分，最低分，并将最高分和最低分学生的下标输出到控制台

## 第9题

勾股定理：直角三角形中，两个直角边的平方和，等于斜边平方和。寻找三边的长度都不大于100，并且三边边长都是整数的可能，在控制台输出。

不涉及数组，也不涉及字符串。就是算法。

所以现在可以写3层for，也可以用设A验B法。