# 输入

## 指定长度---直接scanf

int

char

double

## 给数字 然后给不定长的数据

int+int /char/string； int+矩阵

scanf(int)

for(读取后面的)

# 输入

## 指定长度---直接scanf

int

char

double

## 给数字 然后给不定长的数据

int+int /char/string； int+矩阵

scanf(int)

for(读取后面的)

int +多个句子

scanf（int）

for()+fgets()//要处理回车符

首先要处理scanf留下来的回车符号

然后fgets(arr[i],100,stdin)要这样写，麻烦死了，最左边是存的地方，中间是大小，右边的读取的文件指针。

之后还要处理整行读下来的回车符号str[strcspn(str, "\n")] = '\0';

for()+getchar();

就直接for去读就好了

## 不指定长度

创造一个足够大的二维数组去存

之后用while（scanf == 1）存就好了

string

操作符（char 和数字结合）

int

子网掩码

句子（单个句子和多个句子）

# 必记函数

## qsort

void qsort(void \*base, size\_t nmemb, size\_t size, int (\*compar)(const void \*, const void \*));

这里是各个参数的含义：

base：指向要排序的数组的指针。

nmemb：数组中元素的个数。

size：每个元素的大小（以字节为单位）。

compar：指向比较函数的指针，该函数用于比较两个元素的大小。

// 比较函数，用于升序排序

int compare(const void \*a, const void \*b) {

return (\*(int\*)a - \*(int\*)b);

}

// 使用 qsort 对数组进行排序

qsort(array, array\_size, sizeof(int), compare);

## fgets 连续读取

for (int i = 0; i < MAX\_LINES; i++)

{

if (fgets(str, sizeof(str), stdin) == NULL)

{

// 处理错误或退出循环

break;

}

}

## 宏函数字符交换

\#define SWAPCHAR(a, b) do { char \*temp = (a); (a) = (b); (b) = temp; } while(0)

char\* revstr(char \*s)

\{

int len = strlen(s);

int top = 0, end = len - 1;

while (top < end)

{

SWAPCHAR(s[top], s[end]);

top++;

end--;

}

return s;

\}

// 删除末尾的换行符

str[strcspn(str, "\n")] = '\0';

// 使用 revstr 函数反转字符串

strcpy(strings[i], revstr(str));

## 偷懒小技巧\_\_builtin\_popcount(int)； 这个函数可以直接输出int的二进制1的个数！

int +多个句子

scanf（int）

for()+fgets()//要处理回车符 str[strcspn(str, "\n")] = '\0';

## 不指定长度

string

操作符（char 和数字结合）

int

子网掩码

句子（单个句子和多个句子）