

Welcome

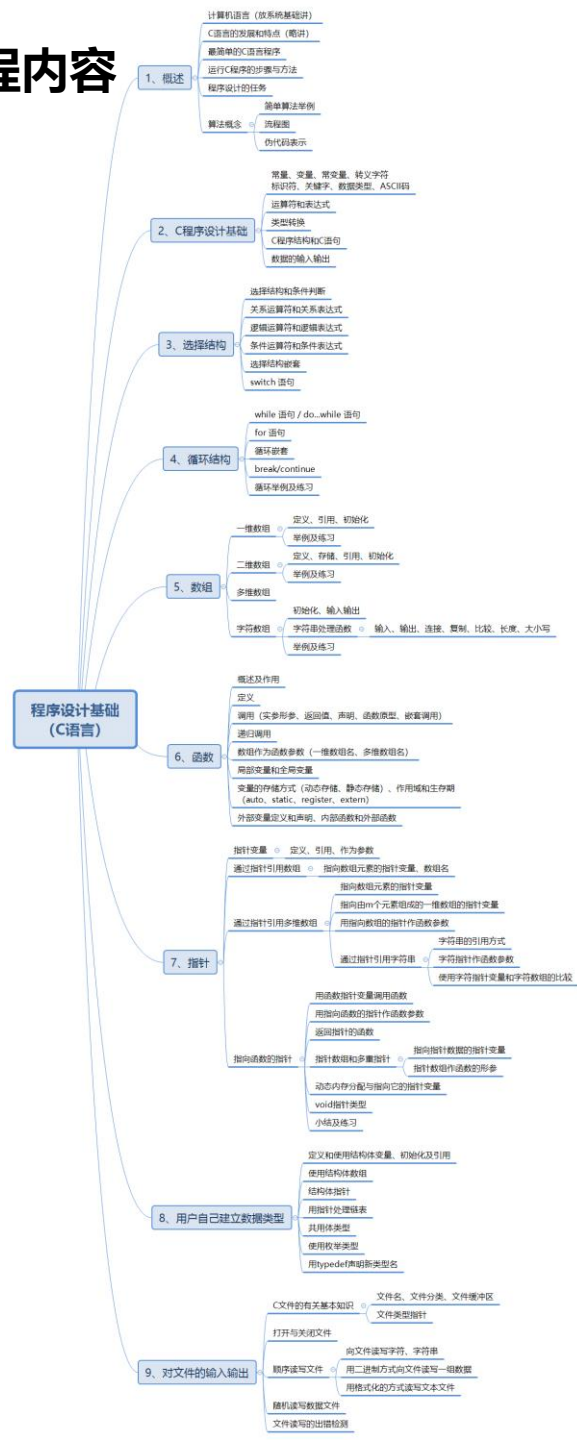
# 数据科学与大数据技术专业 程序设计基础(C语言)

上海体育学院经济管理学院

Wu Ying



# 课程内容



## 第5周 程序设计概述

### 计算机语言及C语言

#### 最简单的C语言程序

#### 运行C程序的步骤和方法

#### 程序设计的任务

#### 算法概念及简单算法

#### 算法表示

#### 流程图

#### 伪代码

## 第6周 C程序设计基础

#### 常量、变量及数据类型

#### 常量变量常量变量转义字符

#### 标识符关键字数据类型

#### 运算符和表达式及类型转换

#### C程序结构和C语句

#### 数据的输入输出

## 第7周 选择结构

#### 选择结构和条件判断

#### 运算符和表达式

#### 关系运算符和关系表达式

#### 逻辑运算符和逻辑表达式

#### 条件运算符和条件表达式

#### 选择结构嵌套

#### switch 语句

## 第8周 循环结构

#### while语句和do...while语句

#### for语句

#### 循环嵌套, break, continue

#### 循环练习

## 第9周 数组 (一)

#### 一维数组定义、引用、初始化

#### 一维数组练习

#### 二维数组定义引用初始化

#### 多维数组及练习

## 第10周 数组 (二) 及函数 (一)

#### 字符数组初始化及输入输出

#### 字符串处理函数

#### 函数定义与调用

#### 函数返回值、声明、原型、嵌套调用

## 第11周 函数 (二)

#### 递归调用

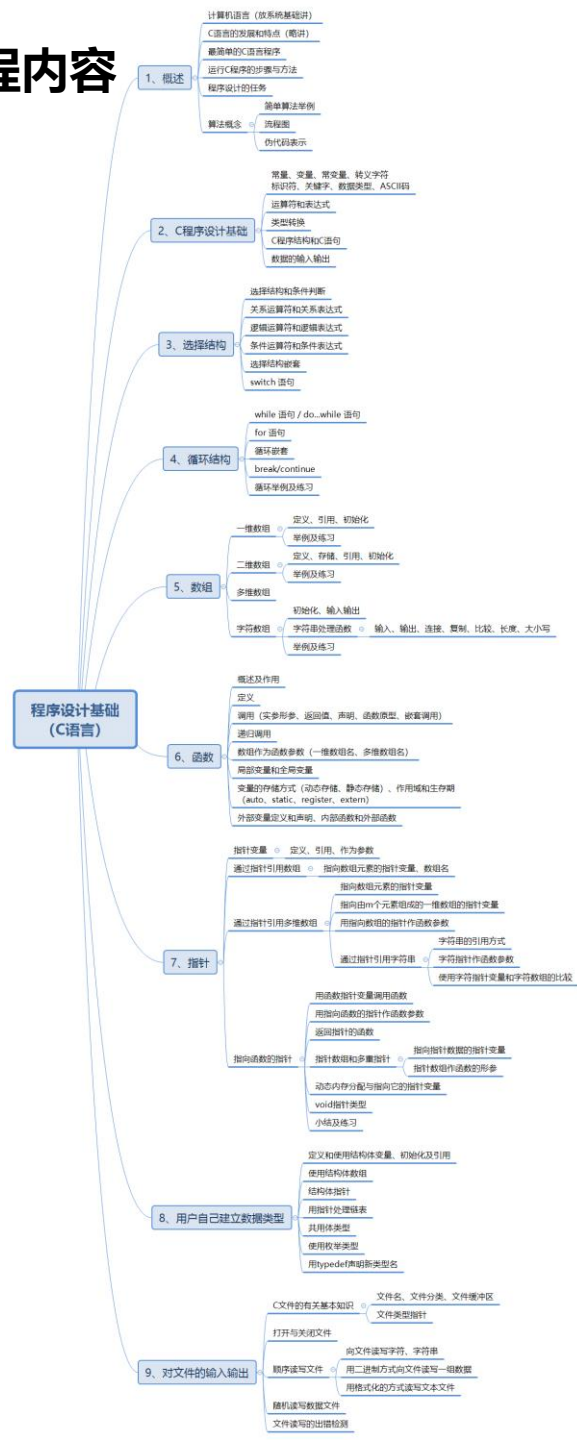
#### 数组作为函数参数

#### 变量作用域和生存期

#### 变量存储方式、内部函数与外部函数



# 课程内容



## 第12周 指针 (一)

指针变量

通过指针引用数组

### 通过指针引用多维数组 (I)

指向数组元素的指针变量

指向一维数组的指针变量

## 第13周 指针 (二)

### 通过指针引用多维数组 (II)

用指向数组的指针作为函数参数

### 通过指针引用字符串

字符串的引用方式

字符指针作函数参数

使用字符指针变量和字符串数组的比较

## 第14周 指针 (三)

### 指向函数的指针 (I)

用函数指针变量调用函数

用指向函数的指针作函数参数

返回指向函数的指针

### 指向函数的指针 (II)

#### 指针数组和多重指针

指向指针数据的指针变量

指针数组作函数的形参

动态内存分配与指向它的指针变量

## 第15周 指针 (四) 及结构体 (一)

### 指向函数的指针 (IV)

void指针类型

小结及练习

### 结构体

定义和使用结构体变量

使用结构体数组、结构体指针

## 第16周 结构体 (二) 及文件 (一)

用指针处理链表、共用体类型

枚举类型、使用typedef声明新类型

C文件的基本知识

打开与关闭文件

## 第17周 文件 (二)

顺序读写文件

随机读写数据文件

总复习练习

总复习练习



# 二维数组的初始化

(1)分行给二维数组赋初值。（最清楚直观）

```
int a[3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}};
```

(2)可以将所有数据写在一个花括号内，按数组元素在内存中的排列顺序对各元素赋初值。

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

(3)可以对部分元素赋初值。

```
int a[3][4]={{1},{5},{9}}; ①
```

```
int a[3][4]={{1},{0,6},{0,0,11}}; ②
```

```
int a[3][4]={{1},{5,6}}; ③
```

```
int a[3][4]={{1},{},{9}}; ④
```

| ①       | ②        | ③       | ④       |
|---------|----------|---------|---------|
| 1 0 0 0 | 1 0 0 0  | 1 0 0 0 | 1 0 0 0 |
| 5 0 0 0 | 0 6 0 0  | 5 6 0 0 | 0 0 0 0 |
| 9 0 0 0 | 0 0 11 0 | 0 0 0 0 | 9 0 0 0 |

(4)如果对全部元素都赋初值(即提供全部初始数据)，则定义数组时对第1维的长度可以不指定，但第2维的长度不能省。

```
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12}; ≡ int a[][4]={1,2,3,4,5,6,7,8,9,10,11,12};
```

在定义时也可以只对部分元素赋初值而省略第1维的长度，但应分行赋初值。

```
int a[][4]={{0,0,3},{},{0,10}};
```

# 字符数组的初始化

```
char c[10]={ 'l',' ','a','m',' ','h','a','p','p','y'};    //把10个字符依次赋给c[0] ~ c[9]这10个元素
```

初值个数**大于数组长度**，则出现语法错误。

初值个数**小于数组长度**，则只将初始字符赋给前面的元素，其余元素自动赋值为**空字符(即'\0')**

```
char c[10]={ 'c',' ','p','r','o','g','r','a','m'};
```

| c[0] | c[1] | c[2] | c[3] | c[4] | c[5] | c[6] | c[7] | c[8] | c[9] |
|------|------|------|------|------|------|------|------|------|------|
| c    | ␣    | p    | r    | o    | g    | r    | a    | m    | \0   |

如果提供的初值个数与预定的数组长度相同，在定义时可以省略数组长度，系统会自动根据初值个数确定数组长度。

```
char c[]={ 'l',' ','a','m',' ','h','a','p','p','y'};    //数组c的长度自动定为10
```

```
char diamond[5][5]={{' ',' ','*'},{' ','*',' ','*'},{'*',' ',' ',' ','*'},{' ','*',' ','*'},{' ',' ','*'}};
```

```
      *
     * *
    *  *
   *  *
  *
```

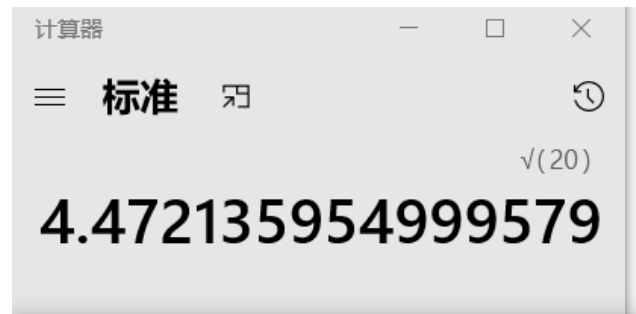
【16】输出以下图案

```
  *
 ***
*****
*****
*****
 ***
  *
```

【13】用迭代法求  $x = \sqrt{a}$  求平方根的迭代公式为  $x_{n+1} =$

$$\frac{1}{2} \left( x_n + \frac{a}{x_n} \right)$$

设  $x_0 = \frac{a}{2}$  , 用以上公式求下一项x 直到前后两次求出的x的差的绝对值小于  $10^{-5}$  为止, 得到  $\sqrt{a}$



Microsoft Visual Studio 调试控制台

```
输入大于0的数: -7
输入大于0的数: -3
输入大于0的数: -9
输入大于0的数: 20
平方根为:4.472136

C:\Users\HP\Desktop\CPra
按任意键关闭此窗口. . . .
```

# 字符数组与字符串

# 字符串和字符串结束标志

0 1 2 3 4 5 6 7 8 9 10

```
char c[]={"I am happy"};
```

```
或 char c[]="I am happy";
```

用一个字符串(字符串的两端用双引号)作为字符数组的初值。

|   |  |   |   |  |   |   |   |   |   |    |
|---|--|---|---|--|---|---|---|---|---|----|
| I |  | a | m |  | h | a | p | p | y | \0 |
|---|--|---|---|--|---|---|---|---|---|----|

**注意**

- 数组c的长度不是10, 而是11。因为字符串常量的最后由系统加上一个'\0'。

```
char c[]={'I', ' ', 'a','m', ' ', 'h','a','p','p','y','\0'};
```

≠

```
char c[]={'I', ' ', 'a','m', ' ', 'h','a','p','p','y'};
```

```
char c[10]={"China"};
```

数组c的前5个元素为: 'C','h','i','n','a',第6个元素为'\0', 后4个元素也自动设定为空字符。

|   |   |   |   |   |    |    |    |    |    |
|---|---|---|---|---|----|----|----|----|----|
| C | h | i | n | a | \0 | \0 | \0 | \0 | \0 |
|---|---|---|---|---|----|----|----|----|----|



# 字符数组的输入输出

(1) 用格式符"%c"输入或输出一个字符。

```
#include <stdio.h>
int main()
{
    char c[15]={'I',' ','a','m',' ','a',' ','s','t','u','d','e','n','t','.'};
    int i;
    for(i=0;i<15;i++)
        printf("%c",c[i]);
    printf("\n");
    return 0;
}
```

(2) 用"%s"格式符，对字符串(string)输入输出。

```
#include <stdio.h>
int main()
{
    char c[]="China";
    printf("%s\n",c);
    return 0;
}
```

## 课后练习13 两个字符串连接起来，不用strcat函数

|   |   |   |   |   |    |   |   |   |   |    |    |    |    |    |    |    |
|---|---|---|---|---|----|---|---|---|---|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| I |   | a | m |   | \0 |   |   |   |   |    |    |    |    |    |    |    |

|   |   |   |   |   |    |   |   |   |   |    |
|---|---|---|---|---|----|---|---|---|---|----|
| 0 | 1 | 2 | 3 | 4 | 5  | 6 | 7 | 8 | 9 | 10 |
| h | a | p | p | y | \0 |   |   |   |   |    |



# 使用字符串处理函数

```
#include<string.h>
```

# 小结

puts(字符数组)

gets\_s(字符数组)

strcat(字符数组1, 字符数组2)

strcpy(字符数组1, 字符串2)

strncpy(字符数组1, 字符串2)

strcmp(字符串1, 字符串2)

strlen(字符数组)

strlwr(字符串)

strupr(字符串)

## 注意

- 常用的字符串处理函数是C语言编译系统提供的公共函数，C库函数
- 不同的编译系统提供的函数数量和函数名、函数功能都不尽相同，必要时查询库函数手册。
- `#include <string.h>`

<https://www.runoob.com/cprogramming/c-function-strcmp.html>



# 测字符串长度的函数

## strlen(字符数组)

**size\_t strlen(const char \*str)**

计算字符串 **str** 的长度，直到空结束字符，不包括空结束字符。

size\_t是unsigned int，在编译系统里定义为  
typedef unsigned int size\_t;

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[10]="China";
    printf("%d,%d\n",strlen(str),strlen("China
"));
}
```

```
C:\WINDOWS\system32\cmd.exe
5, 5
请按任意键继续. . .
```

# 转换为大小写的函数

## strlwr(字符串)

作用：将字符串中大写字母换成小写字母。 `char *strlwr(char *str);`

## strupr(字符串)

作用：将字符串中小写字母换成大写字母。 `char *strupr(char *str);`

```
1  #define _CRT_SECURE_NO_WARNINGS
2  #include<stdio.h>
3  #include<math.h>
4  #include<string.h>
5
6  int main()
7  {
8      char str[] = "HTTP://SUS.EDU.CN";
9      printf("%s\n", _strlwr(str));
10     printf("%s\n", str);
11     return 0;
12 }
```

Microsoft Visual Studio 调试控制台

```
http://sus.edu.cn
http://sus.edu.cn
```





# 字符数组应用举例

【例6.8】输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。

txt：用于存放字符串。

i：计数器，用于遍历字符串中的每个字符。

word：用于判断是否开始一个新单词。若word=0表示未出现新单词，如出现了新单词，就把word置成1。

num：用于统计单词数。

Microsoft Visual Studio 调试控制台

```
ab c de f gh.  
There are 5 words in this line.
```

|   |   |   |  |   |   |  |   |  |   |   |   |    |  |  |  |  |  |  |  |
|---|---|---|--|---|---|--|---|--|---|---|---|----|--|--|--|--|--|--|--|
| a | b | c |  | d | e |  | f |  | g | h | . | \0 |  |  |  |  |  |  |  |
|---|---|---|--|---|---|--|---|--|---|---|---|----|--|--|--|--|--|--|--|

txt

txt[i]

i: 0 ~ txt[i]==\0

word=0



# 字符数组应用举例

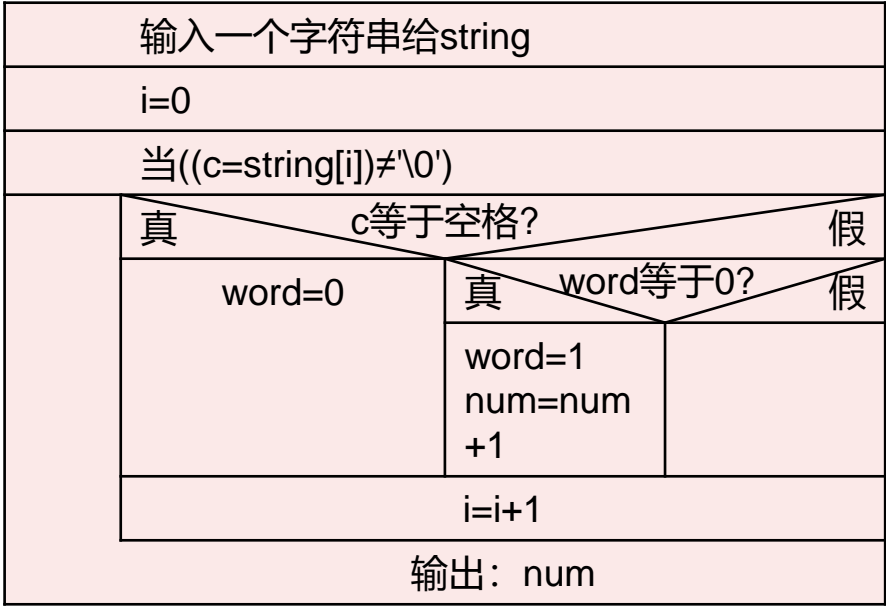
【例6.8】输入一行字符，统计其中有多少个单词，单词之间用一个空格分隔开。

string：用于存放字符串。

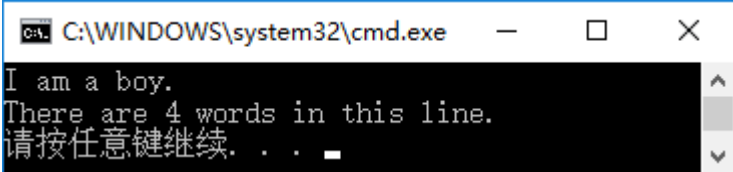
i：计数器，用于遍历字符串中的每个字符。

word：用于判断是否开始了一个新单词的标志。若word=0表示未出现新单词，如出现了新单词，就把word置成1。

num：用于统计单词数。



```
#include <stdio.h>
int main()
{
    char string[81];
    int i,num=0,word=0;
    char c;
    gets(string); //输入一个字符串给字符数组string
    for(i=0;(c=string[i])!='\0';i++) //只要字符不是'\0'就循环
        if(c==' ') word=0; //若是空格字符，使word置0
    else if(word==0) //如果不是空格字符且word原值为0
    {
        word=1; //使word置1
        num++; //num累加1，表示增加一个单词
    }
    printf("There are %d words in this line.\n",num); //输出单词数
    return 0;
}
```





# 作业

---

- ① 【例6.8】 输入一行字符，统计其中有多少个单词，单词之间用空格分隔开。
- ② 课后练习13 两个字符串连接起来，不用strcat函数
- ③ 【例6.9】 有3个字符串,要求找出其中 “最大” 者。

# 作业

④【P165 2】用选择法对10个整数排序 选择排序

|   |
|---|
| 8 |
| 5 |
| 2 |
| 6 |
| 9 |
| 3 |
| 1 |
| 4 |
| 0 |
| 7 |

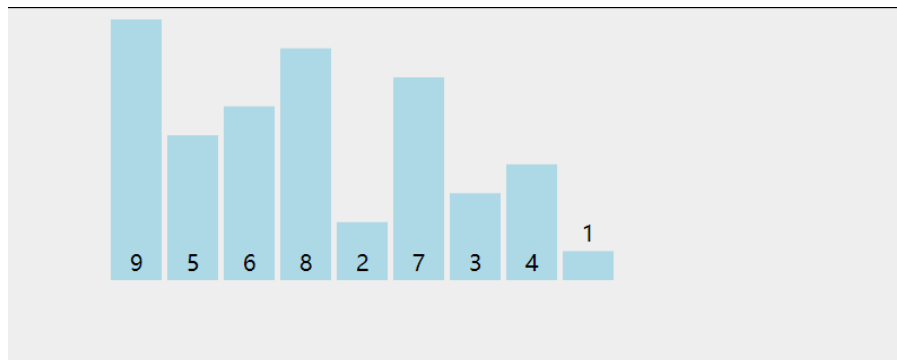
# 作业

④【P165 2】用选择法对10个整数排序

选择排序

|  |   |
|--|---|
|  | 8 |
|  | 5 |
|  | 2 |
|  | 6 |
|  | 9 |
|  | 3 |
|  | 1 |
|  | 4 |
|  | 0 |
|  | 7 |

冒泡排序



# 作业

⑤ 【P165 7】 输出“魔方阵”。所谓魔方阵是这样的方阵，它的每一行、每一列的对角线之和均相等。例如三阶魔方阵，要求输出  $1 - n^2$  的自然数构成的魔方阵， $n$  为奇数。

- 1) 1 放在第一行中间一列;  $[0][n/2]$
- 2) 从 2 开始直到  $n \times n$  为止，各数依次按此规则存放：

$(k - 1) \% N == 0 \rightarrow \text{row}++;$

行-1，如果行为0， $\text{row}=N-1$ ，或者 $\text{row}=\text{row}-1$

$\text{row} = (\text{row} + N) \% N;$

列+1，

$\text{col} = \text{col} \% N$

|   |   |   |
|---|---|---|
| 8 | 1 | 6 |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

|    |    |    |    |    |
|----|----|----|----|----|
| 17 | 24 | 1  | 8  | 15 |
| 23 | 5  | 7  | 14 | 16 |
| 4  | 6  | 13 | 20 | 22 |
| 10 | 12 | 19 | 21 | 3  |
| 11 | 18 | 25 | 2  | 9  |



```

2  #define N 5                //定义符号常量，魔方阵的阶
3  #include<stdio.h>
4  int main()
5  {
6      int a[N][N] = { 0 }; //数组元素初始化为0
7      int row, col;
8      int k;                //魔方阵的数
9
10     //建立魔方阵
11     row = 0;    col = (N - 1) / 2;
12     a[row][col] = 1; // 1 放在第一行的中间
13
14     //依次存放 2 至 N*N
15     for (k = 2; k <= N * N; k++)
16     {
17         if (0 == (k - 1) % N)
18         {
19             row++;
20         }
21         else
22         {
23             row--;
24             row = (row + N) % N; // 上一个数的行号减1为0，row=N-1
25
26             col++;
27             col = col % N; // 上一个数的列号增1若为N，取余为0，回到第一列
28         }
29         a[row][col] = k;
30     }
31
32     //输出魔方阵
33     for (row = 0; row < N; row++)
34     {
35         for (col = 0; col < N; col++)
36             printf("%5d", a[row][col]);
37
38         printf("\n");

```

Microsoft Visual Studio 调试控制台

|    |    |    |    |    |
|----|----|----|----|----|
| 17 | 24 | 1  | 8  | 15 |
| 23 | 5  | 7  | 14 | 16 |
| 4  | 6  | 13 | 20 | 22 |
| 10 | 12 | 19 | 21 | 3  |
| 11 | 18 | 25 | 2  | 9  |