

网安4学分计导期末

题型	作答 / 题数
❌ 判断题	18 / 18
A. 单选题	17 / 17
A 填空题	3 / 3
D 程序填空题	3 / 3
f 函数题	3 / 4
</> 编程题	3 / 4

☒ ☒ ☒ ☒ ☒ ☒

☒ ☒ ☒ ☒ ☒ ☒

1-1 分数 1

单向链表中的每个结点都需要动态分配内存空间。

☒ T ☐ F

答案正确: 1 分

1-2 分数 1

对于缓冲文件系统，在进行文件操作时，系统自动为每一个文件分配一块文件内存缓冲区（内存单元）。

☒ T ☐ F

答案正确: 1 分

1-3 分数 1

结构体类型可用于表示复杂构造类型，假设想要跟踪图书馆中书本的动态，需要跟踪每本书的某些属性：BookID、Title、Author、Price，可定义Book结构体类型用来封装这些属性：

```
struct Book{
    int id;
    char title[50];
    char author[20];
    double price;
};
```

下面定义该类型的变量或数组是否正确？

```
struct Book b1,b2,c[10];
```

☒ T ☐ F

答案正确: 1 分 创建提问

1-4 分数 1

在if语句的三种形式中，如果要想在满足条件时执行一组(多个)语句，则必须把这一组语句用{}括起来组成一个复合语句。

☒ T ☐ F

答案正确: 1分 创建提问

1-5 分数 1

如果函数定义出现在函数调用之前，可以不必加函数原型声明。

☒ T ☐ F

答案正确: 1 分  创建提问 

1-6 分数 1

实参向形参进行数值传递时，数值传递的方向是单向的，即形参变量值的改变不影响实参变量的值。

☒ T ☐ F

答案正确: 1分  创建提问 

1-7 分数 1

在switch语句中，多个case可以共用一组执行语句。

☒ T ☐ F

答案正确: 1 分  创建提问 

1-8 分数 1

在C语言中，数组名是不可改变的，不能对它进行赋值运算。

☒ T ☐ F

答案正确: 1分 创建提问

1-9 分数 1

数组初始化时，初始值个数小于数组元素的个数，C语言自动将剩余的元素初始化为初始化列表中的最后一个初始值。

☐ T ☒ F

答案正确: 1 分 创建提问

1-10 分数 1

作者 zhanghd 单位 福州大学

else 总是与它上面最近的且尚未配对的 if 配对。

☐ T

☒ F

答案错误: 0 分 [创建提问](#)

图例

共 10 分

>

< 上一题

□ 单题作答

下一题 >

2022秋-计导期末考试

题目列表

提交列表

排名

题型

作答 / 题数

判断

10 / 10

A. 单选题

17 / 17

填空题

3 / 3

程序填空题

3 / 3

函数题

3 / 4

编程题

3 / 4

✓

✓

✗

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

2-5 分数 1

作者 张泳 单位 浙大城市学院

以下程序的输出结果是（ ）。

```
#include <stdio.h>
#include <stdlib.h>

struct NODE{
    int num;
    struct NODE *next;
};

int main(void)
{
    struct NODE *p, *q, *r;

    p = (struct NODE*)malloc(sizeof(struct NODE));
    q = (struct NODE*)malloc(sizeof(struct NODE));
    r = (struct NODE*)malloc(sizeof(struct NODE));
    p->num = 10;
    q->num = 20;
    r->num = 30;
    p->next = q;
    q->next = r;
    printf("%d\n", p->num + q->next->num);

    return 0;
}
```

☐ A. 10

☐ B. 20

☐ C. 30

☒ D. 40

答案正确: 1 分

2-2 分数 1

作者 张泳 单位 浙大城市学院

若fopen()函数打开文件失败，其返回值是（ ）。

☐ A. 1

☐ B. -1

☒ C. NULL

☐ D. ERROR

答案正确: 1 分

2-3 分数 1

作者 张泳 单位 浙大城市学院

若以"a+"方式打开一个已存在的文件，则以下叙述正确的是（ ）。

☐ A. 文件打开时，原有文件内容不被删除，位置指针移到文件末尾，可作添加和读操作

☐ B. 文件打开时，原有文件内容不被删除，位置指针移到文件开头，可作重写和读操作

☐ C. 文件打开时，原有文件内容被删除，只可作写操作

☒ D. 以上各种说法都不正确

答案错误: 0 分

2-4 分数 1

作者 王秀 单位 福州大学

在一个被调用函数中，关于 return 语句使用的描述，错误的是（ ）。

☐ A. 被调用函数中可以不用 return 语句

☐ B. 被调用函数可以使用多个 return 语句

☐ C. 被调用函数中，如果有返回值，就一定要有 return 语句

☒ D. 被调用函数中，一个 return 语句可以返回多个值给调用函数

答案正确: 1 分

创建提问

2-5 分数 1

作者 王秀 单位 福州大学

若x为浮点型，则表达式 $x=10/4$ 为（ ）。

☐ A. 2.5

☒ B. 2.0

☐ C. 2

☐ D. 3

答案正确: 1 分

创建提问

2-6 分数 1

作者 张高磊 单位 浙大城南学院

对于以下递归函数f，调用f(4)，其返回值为（）。

```
int f(int n)
{
    if (n) return f(n - 1) + n;
    else return n;
}
```

- ☒ A. 10
- ☐ B. 4
- ☐ C. 0
- ☐ D. 以上均不是

答案正确：1分 创建提问

2-7 分数 1

作者 wx 单位 福州大学

表达式_____的值是0。

- ☐ A. 3%5
- ☐ B. 3/5.0
- ☒ C. 3/5
- ☐ D. 3<5

答案正确：1分 创建提问

2-8 分数 1

作者 吴永芬 单位 中国人民解放军陆军工程大学

能正确表示“当x的取值在[-58,-40]和[40,58]范围内为真，否则为假”的表达式是（）。

- ☐ A. (x>= -58) && (x<= -40) && (x>=40) &&(x<=58)
- ☐ B. (x>= -58) || (x<= -40) || (x>=40) || (x<=58)
- ☒ C. (x>= -58) && (x<= -40) || (x>=40) && (x<=58)
- ☐ D. (x>= -58) || (x<= -40) && (x>=40) || (x<=58)

答案正确：1分 创建提问

2-9 分数 1

作者 usx程序设计类课程组 单位 绍兴文理学院

判断a、b中有且仅有1个值为0的表达式是()。

- ☒ A. !(a*b)&&a+b
- ☐ B. (a*b)&&a+b
- ☐ C. a*b==0
- ☐ D. a!=0 && !b

答案正确：1分 创建提问

2-10 分数 1

作者 wjl 单位 赤峰学院

若 int i=10; 执行下列程序后，变量 i 的正确结果是()。

```
1 switch (i) {
2     case 9: i+=1 ;
3     case 10: i+=1 ;
4     case 11: i+=1 ;
5     default : i+=1 ;
6 }
```

- ☐ A. 10
- ☐ B. 11
- ☐ C. 12
- ☒ D. 13

答案正确：1分 创建提问

2-11 分数 1

作者 王秀秀 单位 山东交通学院

若有定义int a[4][5]; 下列语句中（）是存在语法错误的。

2022秋-计导期末考试

题目列表

提交列表

排名

题型

作答 / 题数

判断题

10 / 10

A. 单选题

17 / 17

填空题

3 / 3

程序填空题

3 / 3

函数题

3 / 4

编程题

3 / 4

✓

✓

✗

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

✓

2-11 分数 1

作者 王秀秀 单位 山东交通学院

若有定义 `int a[4][5];` 下列语句中 () 是存在语法错误的。

☐ A. `*a`

☐ B. `*(*a+2)+3`

☐ C. `&a[2][5]`

☒ D. `**a`

答案正确: 1 分 创建提问

2-12 分数 1

作者 王秀秀 单位 山东交通学院

下面程序段执行后的输出结果是 ()。

```
char a[] = "language", *p;
p = a;
while ( *p != 'u' )
{
    printf( "%c", *p - 32 ); p++;
}
```

☐ A. LANGUAGE

☐ B. language

☒ C. LANG

☐ D. langUAGE

答案正确: 1 分 创建提问

2-13 分数 1

作者 李祥 单位 湖北经济学院

关于结构化程序设计

结构程序设计的三种结构是 _____。

☐ A. 顺序结构、选择结构、转移结构

☐ B. 分支结构、等价结构、循环结构

☐ C. 多分支结构、赋值结构、等价结构

☒ D. 顺序结构、选择结构、循环结构

答案正确: 1 分 创建提问

2-14 分数 1

作者 赖晓燕 单位 福建农林大学金山学院

下列程序执行后的输出结果是

```
1 main()
2 {
3     int x='f';
4     printf("%c \n", 'A'+(x-'a'+1));
5 }
```

☒ A. G

☐ B. H

☐ C. I

☐ D. J

答案正确: 1 分 创建提问

2-15 分数 1

作者 cui 单位 成都信息工程大学

在以下一组运算符中,优先级最高的运算符是()。

☐ A. <=

☐ B. =

☒ C. %

☐ D. &&

答案正确: 1 分 创建提问

2-16 分数 1

作者 王秀 单位 福州大学

如果要求在 `if` 后一对括号中的表达式在表示 `a` 不等于 0 的时候的值为“真”，则能正确表示这一关系的表达式为 ()。

☐ A. `a<>0`

☐ B. `!a`

☐ C. `a=0`

☒ D. `a`

答案正确: 1 分 创建提问

答案正确: 1 分

创建提问

2-17 分数 1

作者 王秀 单位 福州大学

设有程序段

```
int m=20; while (m=0) m=m++;
```

则下面描述中正确的是 ()。

☐ A. while 循环执行 10 次

☐ B. 循环是无限循环

☒ C. 循环体语句一次也不执行

☐ D. 循环体语句执行一次

答案正确: 1 分

创建提问

共 17 分

图例

单题作答

上一题

下一题

2022秋-计导期末考试

题目列表

提交列表

排名

题型

作答 / 题数

判断題

10 / 10

A. 单选题

17 / 17

填空题

3 / 3

程序填空题

3 / 3

函数题

3 / 4

编程题

3 / 4

×

✓

✓

4-3 分数 2

作者 张咏 单位 浙大城市学院

下列函数用于将链表中各结点的数据依次输出。

```
struct student {
    long data;
    struct student *next;
};
void print(struct student *head)
{
    struct student *p ;
    p=head;
    if(head != NULL)
        do {
            printf("%ld\n", p->data);
            p=p->next 1 分 ;
        } while ( 1 分 );
}
```

部分正确: 1 分

4-4 分数 2

作者 张咏 单位 浙大城市学院

若有定义 `int a = 10, *p1 = &a, **p2 = &p1;`, 则表达式 `**p2` 的值是 10 2 分。

答案正确: 2 分

4-5 分数 2

作者 张咏 单位 浙大城市学院

若有定义 `double *p;` 请写出利用 `malloc` 函数使 `p` 指向一个双精度型的动态存储单元的完整语句为

```
p= (double*)malloc(sizeof(double)) 2 分。
```

答案正确: 2 分

共 6 分

图例

单题作答

上一题

下一题

2022秋-计导期末考试

题目列表

提交列表

排名

题型

作答 / 题数

判断题

10 / 10

A. 单选题

17 / 17

A. 填空题

3 / 3

程序填空题

3 / 3

函数题

3 / 4

编程题

3 / 4

✓

✓

✓

5-2 最小的字符串

分数 3

作者 张咏

单位 浙大城市学院

输出多个字符串中最小的字符串。

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

int main()
{
    const char *st[] = {"bag", "good", "This", "are", "Zoo", "park"};
    const char *smin;
    int i;

    smin= st[0] 1分 ;
    for(i = 1; i < 6; i++){
        if(strcmp( st[i],smin 1分 ) < 0){
            smin = st[i];
        }
    }
    printf("%s\n", smin 1分 );

    return 0;
}
```

答案正确: 3分

5-2 指针数组的访问

分数 2

作者 smilesnaill

单位 北京邮电大学

编写程序输出language数组元素所指向的5个字符串, 请填与程序中相应语句。

```
#include<stdio.h>
int main(void)
{
    char * language[] = {"BASIC", "FORTRAN", "PROLOG", "JAVA", "C++"};
    char ** q;
    int k;

    for(k = 0 ; k < 5 ; k++) {
        q = language+k 2分 ;
        printf("%s", *q);
    }

    return 0;
}
```

答案正确: 2分

5-3 查找最高分

分数 2

作者 smilesnaill

单位 北京邮电大学

输入10个成绩, 查找最高分并输出。

```
#include <stdio.h>
int * getMax(int score[ ], int n);
int main(void)
{
    int i, score[10], *p;

    for(i = 0; i < 10; i++){
        scanf("%d", &score[i]);
    }

    p = getMax(score,10) 1分 ;
    printf("%d", *p);

    return 0;
}

int * getMax(int score[ ], int n)
{
    int i, max, pos = 0;

    max = score[0] ;
    for(i = 0 ; i < n ; i++){
        if(score[i] > max){
            max = score[i];
            pos = i ;
        }
    }

    return score+pos 1分 ;
}
```

答案正确: 2分

共 7 分

图例

单题作答

下一题

6-1 数组插入

分数 10

现有两个数组A和B，你的任务是根据指定位置将B中全部元素插入到A中去。

输入格式：
第一行为三个用空格分隔的整数m,n, pos，依次代表数组A中元素的个数，数组B中的元素的个数和插入的位置；
第二行为m个用空格分隔的整数，为A中元素；
第三行为n个用空格分隔的整数，为B中元素。

输出格式：
只有一行，为m+n个用空格分隔的整数，为插入后A中元素。
测试用例保证m,n均大于0，且0<=pos<=m, m+n<=100。

完成这个题目你需要设计两个函数。

函数接口定义：

```
1 void create( int array[] , int num );
```

该函数的功能是将num个整数读入数组array中，其中array和num都是用户传入的参数。array为用于保存整数的数组名；num为读入元素的个数。函数没有返回值。

```
1 void insert( int A[] , int B[] , int lenA , int lenB , int pos );
```

A、B、lenA、lenB和pos都是用户传入的参数。其中A和B为待操作数组；lenA和lenB分别代表数组A和数组B中元素的个数；pos代表插入的起始位置。函数没有返回值。
该函数的功能是将数组B中的全部元素依次插入到数组A中。插入的起始位置为pos，例如：pos等于3，则B[0]插入后处于A[3]的位置，B[1]插入后处于A[4]的位置，后续元素依此类推。插入的同时数组A中从pos开始的各个元素要依次后挪。

裁判测试程序样例：

```
1 #include<stdio.h>
2
3 #define      MAX      100
4
5 void insert( int A[] , int B[] , int lenA , int lenB , int pos );
6 void create( int array[] , int num );
7
8 int main()
9 {
10     int      A[MAX] , B[MAX] ;
11     int      i , m , n , pos ;
12
13     scanf(  "%d%d%d" , &m , &n , &pos ) ;
14     create( A , m ) ;
15     create( B , n ) ;
16     insert( A , B , m , n , pos ) ;
17
18     for ( i = 0 ; i < m + n ; i ++ )
19     {
20         if ( i == m + n - 1 ) printf("%d\n" , A[i]) ;
21         else printf("%d " , A[i]) ;
22     }
23
24     return 0;
25 }
26
27
28 /* 请在这里填写答案 */
```

输入样例：

```
9 3 3
1 2 3 4 5 6 7 8 9
11 12 13
```

输出样例：

```
1 2 3 11 12 13 4 5 6 7 8 9
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

6-2 递归求最小值

分数 10

请写一个递归函数，该函数的功能为查找一个整数数组中从某个位置开始后边所有整数的最小值。

输入格式：

共2行，第一行为一个整数n(0<n<=100)，代表数组中共n个整数；第二行为用空格分隔的n个整数，为待查找整数。

输出格式：

共n+1行，前n行依次为后1个整数的最小值、后2个整数的最小值，以此类推直到全部n个整数的最小值。最后一行为全部n个整数的最小值。具体格式见样例。

测试用例保证所有整数可以用int类型存储。

** 需要注意的是前n行要在递归函数中输出，最后一行由主函数输出。**

注意：此题要求递归求解，且不允许使用全局变量、静态变量等等。

函数接口定义：

```
1 int findMin(int numbers[], int first, int n) ;
```

其中 numbers、first 和 n 都是用户传入的参数。numbers 为待查找数组；first 为查找的起始位置；n 为数组中元素的总个数。函数须返回查找到的最小值。

裁判测试程序样例：

```
1 #include<stdio.h>
2
3 #define      MAX      100
4
5 int findMin(int numbers[], int first, int n) ;
6
7 int main()
8 {
9     int      i , n , numbers[MAX];
10
11     scanf( "%d" , &n ) ;
12     for ( i = 0 ; i < n ; i++ )
13         scanf( "%d" , &numbers[i] ) ;
14     printf( "min = %d\n" , findMin( numbers , 0 , n ) ) ;
15
16     return 0;
17 }
18
19
20 /* 请在这里填写答案 */
```

输入样例：

```
9
1 2 3 4 5 6 7 8 9
```

输出样例：

```
9
8
7
6
5
4
3
2
1
min = 1
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

6-3 奇怪的编码 分数 5

X公司最近研究出了一种新的编码方式，这种编码是将一串大写字母转化为数字。下面是每个字母所代表的数字。
B, F, P, V 代表 1
C, G, J, K, Q, S, X, Z 代表 2。
D, T 代表 3
L 代表 4
M, N 代表 5
R 代表 6。
另外字母A, E, I, O, U, H, W,Y不代表任何数字。你需要将每个字母转化为所代表的数字，但注意，如果连续几个字母都代表同一个数字，那么你只需要转化成一个数字就可以。比如PF,P和F都代表1，所以你只需要输出一个1，而不是两个1。不代表数字的字母不用转化。

输入格式
第一行为整数T(0<T<10)代表T组测试数据
每行一个只含大写字母的字符串，串长不超过20。
输出格式
共T乘2行，每两行依次对应一行输入；其中第一行输出所转化后的数字编码，如果没有任何编码，输出一个空行；第二行输出转化后编码的长度。
你需要设计一个函数来完成此编码功能。

函数接口定义：

```
1 int code( char s[] ) ;
```

其中 s 是用户传入的参数，代表待编码字符串；函数的返回值为编码后的长度。注意，你的编码函数要输出编码后的结果，而编码后的长度在主函数里输出了，所以无需在编码函数中输出。

裁判测试程序样例：

```
1
2 #include<stdio.h>
3
4 int code( char s[] ) ;
5
6 int main()
7 {
8     int T;
9     char s[64] ;
10
11     scanf( "%d" , &T );
12     while( T-- )
13     {
14         scanf("%s",s);
15         printf( "%d\n" , code(s) );
16     }
17
18     return 0;
19 }
20
21
22
23 /* 请在这里填写答案 */
```

输入样例：

```
3
KHAWN
PFISTER
BOBBY
```

输出样例：

```
25
2
1236
4
11
2
```

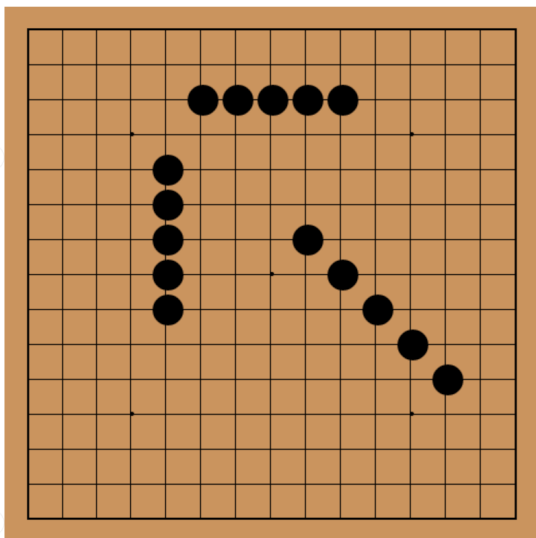
代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

6-4 五子棋一 分数 10

全屏浏览题目 切换布局

作者 scs 单位 北京邮电大学

五子棋是世界智力运动会竞技项目之一，是一种两人对弈的纯策略型棋类游戏。通常双方分别使用黑白两色的棋子，下在棋盘直线与横线的交叉点上，先形成5子连线者获胜。假设棋盘为19X19路的棋盘，棋盘的左上角的坐标为（1，1），它右边的一个点为（1，2），再右边一个点为（1，3）；左上角下边的一个点为（2，1）；依此类推最右下角的坐标为（19，19）。现给出5个点的坐标，你写一段程序判断这5个点是否在横向或者纵向上紧密的连成一排（也就是五子棋获胜的两种情况）。



输入：

共5行，每行为两个整数，依次代表一颗棋子的行坐标和列坐标（测试数据保证合法）。

输出：

共6行，

第1行，如果这五颗子在棋盘上的横向或纵向上紧密的连成一排，则输出WIN，否则输出NO。

后5行，无论第1行输出什么，后5行均将输入的5个坐标排序后输出，每行一个坐标。同一行内先输出行坐标后输出列坐标，中间用一个空格分隔；5个坐标输出的顺序为行坐标小的先输出，如果行坐标相同则列坐标小的先输出。具体见样例。

为了实现此功能，你要设计的两个函数如下：

函数接口定义：

```
1 PIECE * getMemory( int n );
```

该函数的功能为创建一个长度为 `n` 的PIECE型（该类型定义见裁判测试程序样例）动态一维数组，并返回动态数组第一个元素的地址。其中 `n` 是用户传入的参数，代表数组的长度。函数的返回值为该数组的首元素地址。

```
1 int check( PIECE * pieces , int n );
```

该函数的功能为判断给定的数组内的5个棋子是否符合题目要求同时按题目要求将各个棋子排好序。其中 `pieces` 和 `n` 是用户传入的参数，`pieces` 为指向保存棋子的数组的指针，`n` 代表数组的内元素的个数。如果符合题目要求则函数返回1，否则函数返回0。

裁判测试程序样例：

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct piece
5 {
6     int x;
7     int y;
8 } PIECE; // 保存一个棋子坐标的结构
9
10 PIECE * getMemory( int n );
11 int check( PIECE * pieces , int n );
12
13 int main()
14 {
15     PIECE *pieces ;
16     int i ;
17     const int num = 5 ;
18
19     pieces = getMemory( num );
20     if ( pieces == NULL )
21         return -1 ; // 这里的return是因为没有获得内存而直接结束程序。
22
23     for( i = 0 ; i < num ; i++ )
24     {
25         scanf( "%d%d" , &pieces[i].x, &pieces[i].y );
26     }
27
28     if ( check( pieces , num ) ) printf("WIN\n");
29     else printf("NO\n");
30
31     for( i = 0 ; i < num ; i++ )
32     {
33         printf( "%d %d\n" , pieces[i].x, pieces[i].y );
34     }
```

```
34     }
35
36     free( pieces ) ;
37
38     return 0;
39 }
40
41
42
43 /* 请在这里填写答案 */
```

输入样例:

```
1 1
1 2
1 3
1 4
1 5
```

输出样例:

```
WIN
1 1
1 2
1 3
1 4
1 5
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

题目描述

7-1 球员分组 分数 10

全屏浏览题目 切换布局

作者 scs 单位 北京邮电大学

众所周知，2022年卡塔尔世界杯于2022年11月20日至12月18日在卡塔尔境内8座球场举行。最后阿根廷队通过点球大战取胜法国队，赢得大力神杯。世界杯期间各球队除了比赛，训练也很重要。某球队在一次训练中教练把球员分成3组进行分组对抗。分组的依据是球员的号码，号码可以整除3的一组，号码除以3余1的一组，号码除以3余2的一组。现请你写一段程序帮助教练员分组。

输入格式:

共两行，第一行为一个整数n(0<n<=100)，代表球员个数，第二行为n个用空格分隔的整数，代表每个球员号码。

输出格式:

只有一行，为n个整数，代表分组后的结果。每个整数之间用1个空格分隔，最后一个整数后无空格，为一个换行符。输出顺序为：最前边输出号码能被3整除的，中间输出号码除以3余1的，最后输出号码除以3余2的。如果两个号码除以3的余数相同，则按原顺序输出，也就是说在输入中在前边的数在输出中仍然在前边。测试用例保证所有整数均可以用int存储。

输入样例:

```
11
1 2 3 4 5 6 7 8 9 10 11
```

输出样例:

```
3 6 9 1 4 7 10 2 5 8 11
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

7-2 按钮灯矩阵二

分数 5

全屏浏览题目 切换布局

作者 scs 单位 北京邮电大学

有一个由按钮灯(该物体即是按钮也是灯)组成的矩阵，每次按一下某个按钮灯后，该按钮灯及该灯的左上、右上、左下、右下的按钮灯的状态都会改变一次。即，如果灯原来是点亮的，就会被熄灭；如果灯原来是熄灭的，则会被点亮。注意当按一次在矩阵角上的按钮灯时只能改变2个按钮灯的状态；当按一次在矩阵边缘上的按钮灯时只能改变3个按钮灯的状态；现请你写一段程序，根据按钮灯矩阵的初始状态及按按钮的过程给出按钮灯矩阵的最终状态。

输入格式:

第一行为两个正整数m,n，分别代表矩阵的行数和列数(0<m,n<10)
接下来是m行n列个整数，代表按钮灯的初始状态（1代表灯是点亮的，0代表灯是熄灭的）；
接下来一行为一个整数t(0<t<20)，代表共按了t次按钮；
最后t行，每行为用空格分隔的两个整数，分别代表每次按下的按钮灯的坐标。如 1 1 代表按了最左上角的按钮灯， 1 2 代表按了第1行第2列的按钮灯等等，以此类推。

输出格式:

为一个mXn的整数矩阵，代表按钮灯的最终状态（1代表灯是点亮的，0代表灯是熄灭的）。注意灯的状态之间用1个空格分隔，但行末没有空格。具体见样例。

输入样例:

```
5 6
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
1
4 5
```

输出样例:

```
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 1 0 1
0 0 0 0 1 0
0 0 0 1 0 1
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

题目描述

7-3 烽火台一

分数 5

全屏浏览题目 切换布局

作者 scs 单位 北京邮电大学

烽火台，又称烽燧，俗称烽堠、烟墩、墩台。古时用于点燃烟火传递重要消息的高台，是古代重要军事防御设施，是为防止敌人入侵而建的，遇有敌情发生，则白天施烟，夜间点火，台台相连，传递消息。是最古老但行之有效的消息传递方式，它可以说是现代通信的雏形。

现有烽火台若干，但它们之间的通信已不用这种古老的方式了。为了保证所有烽火台之间通信的可靠性，设计者为每对烽火台之间都铺设了一对专门的通信线路。假设所有的烽火台都在一条直线上，请你写一段程序计算一下这些通信线路的总长度。

样例解释：

坐标为1的烽火台向外的通信线路总长：1+2+3+4=10；

坐标为5的烽火台向外的通信线路总长：4+3+2+1=10；

坐标为3的烽火台向外的通信线路总长：2+1+1+2=6；

坐标为2的烽火台向外的通信线路总长：1+1+2+3=7；

坐标为4的烽火台向外的通信线路总长：3+2+1+1=7

通信线路总长为：10+10+6+7+7 = 40

输入格式：

共两行，第一行为一个整数n(1<=n<=10000)，代表同n个烽火台；第二行为n个用空格分隔的整数(这些整数均大于0且不相等)，代表每个烽火台的坐标。

输出格式：

只有一行，为一个整数，代表通信线路总长度。测试用例保证所有整数都可以用int存储，包括总长度。

输入样例：

```
5
1 5 3 2 4
```

输出样例：

```
40
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

题目描述

7-4 烽火台二

分数 5

全屏浏览题目 切换布局

作者 scs 单位 北京邮电大学

烽火台，又称烽燧，俗称烽堠、烟墩、墩台。古时用于点燃烟火传递重要消息的高台，是古代重要军事防御设施，是为防止敌人入侵而建的，遇有敌情发生，则白天施烟，夜间点火，台台相连，传递消息。是最古老但行之有效的消息传递方式，它可以说是现代通信的雏形。

现有烽火台若干（哈哈，它们就是烽火台一里的那些烽火台）。现在的问题是，烽火台有了，通信也得到有效的保障，但没有足够的哨兵。所以只能在部分烽火台上放置观察哨。为了提高观察效率上级设计了任意两个哨兵间的最小距离最大的哨兵安排策略。现请你写一个程序实现这个策略。

样例解释：

3个哨兵分别放在1、4和8（或1，4，9），它们之间的最小距离为3。其他方案最小距离均小于3，所以3是要求的最大值。

输入格式：

只有两行，第一行为两个用空格分隔的整数n和m(2<=m<=n<=10000)，依次代表烽火台的个数和哨兵的个数；第二行为n个用空格分隔的整数，代表烽火台的坐标

（依然假设所有烽火台在同一直线上），这些整数均大于等于0小于等于10⁹，且互不相等。

输出格式：

只有一行，为一个整数，代表所求得的最大值。测试用例保证所有整数都可以用int存储。

输入样例：

```
5 3
1 2 8 4 9
```

输出样例：

```
3
```

代码长度限制	16 KB
时间限制	400 ms
内存限制	64 MB

