

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

FEEL THE MEANING OF THE TRIP

DREAM
MY DREAM WILL NEVER STOP

高级语言程序设计

实验七 指针和数组

GO!
TAKE YOU ON A TRIP



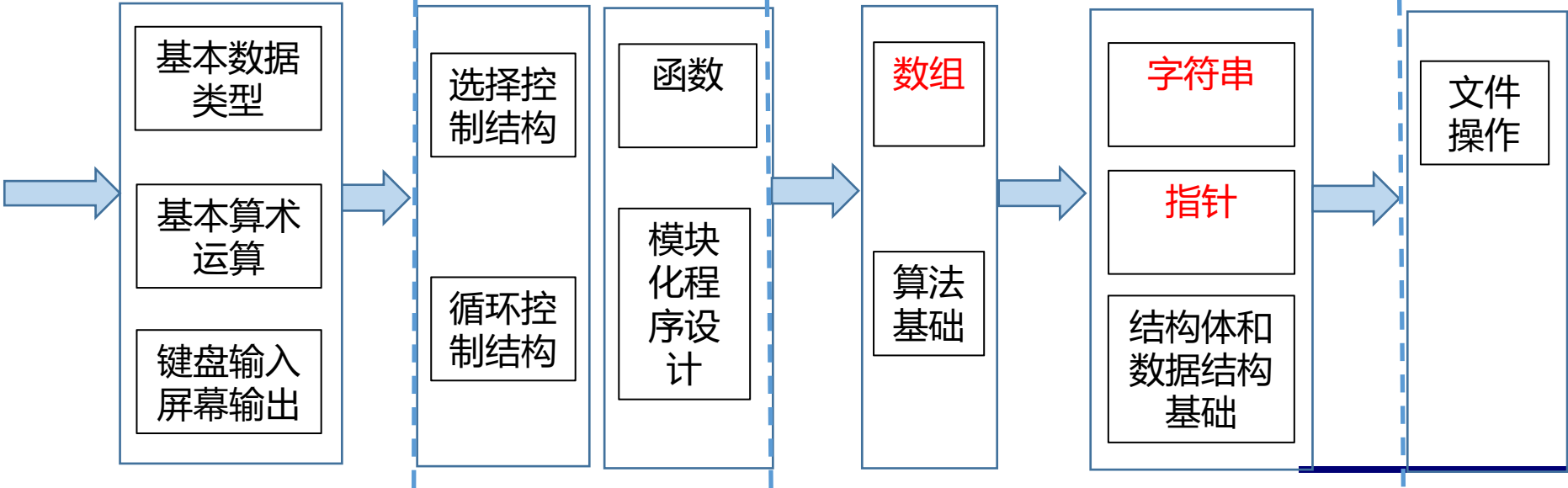
哈尔滨工业大学(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

探索 从未停止



实验课程安排与考核标准

编号	1	2	3	4	5	6	7	8	9
学时数	2	2	2	2	2	2	2	2	4
实验项目	初级编程	算数运算	循环控制	模块化程序设计	数组算法	字符串二维数组	指针	结构体	文件管理
分数	1	2	2	2	3	3	3	3	3
授课内容	集成开发环境	程序调试	编码规范	程序测试				软件文档	期末检查



目录

CONTENT

01

实验回顾

02

实验目的

03

实验原理

04

实验内容

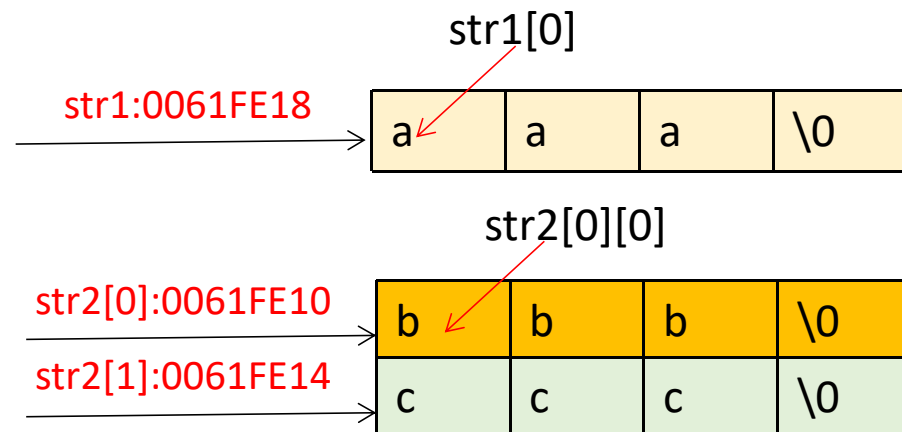
实验回顾

```
*main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4
5  int main()
6  {
7
8      char str1[4]={'\0'};
9      printf("Please input str1:\n");
10     scanf("%s",&str1);
11     printf("str1=%s\n",str1);
12
13     char str2[2][4]={'\0'};
14     for(int i = 0; i < 2; i++)
15     {
16         printf("\nPlease input str2[%d]:\n",i);
17         scanf("%s",&str2[i]);
18         printf("str2[%d]=%s\n",i,str2[i]);
19     }
20
21     return 0;
22 }
23
```

Logs & others

Code::Blocks x Search results x Cccc x Build log x Build messages x CppCheck/Ver+

File	Line	Message
=== Build: Debug in stucode (compiler: GNU GCC Compiler) ===		
In function 'main':		
D:\code\c\2...	10	warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[4]' [-Wformat=]
D:\code\c\2...	10	warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[4]' [-Wformat=]
D:\code\c\2...	17	warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[4]' [-Wformat=]
D:\code\c\2...	17	warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[4]' [-Wformat=]
=== Build finished: 0 error(s), 4 warning(s) (0 minute(s), 0 second(s)) ===		



```
D:\code\c\2022\stucode\bin\Debug\stucode.exe
Please input str1:
000000000061FE18 printf("%p\n",str1);
aaa
str1=aaa

Please input str2[0]:
000000000061FE10 printf("%p\n",str2[i]);
bbb
str2[0]=bbb

Please input str2[1]:
000000000061FE14
ccc
str2[1]=ccc

Process returned 0 (0x0)   execution time : 9.225 s
Press any key to continue.
```



实验目的

- **掌握指针和字符串的使用；**
- **掌握一维数组和二维数组的使用；**
- **掌握指针，一维数组和二维数组作为函数参数**



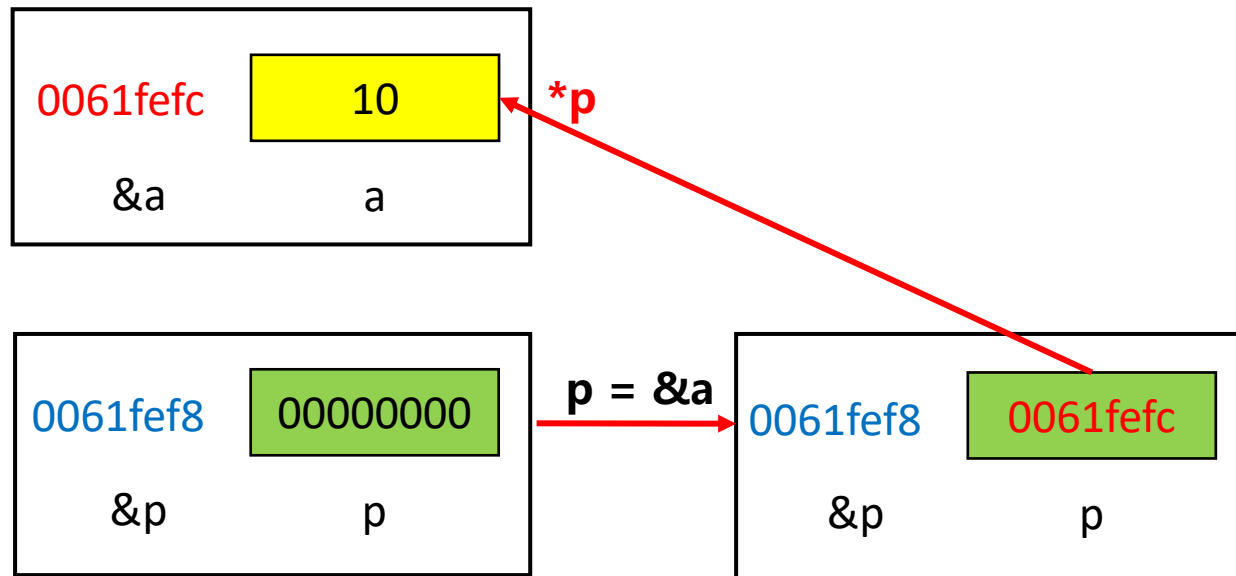
实验原理

➤ 什么是指针?

指针变量是专门用于存储变量的地址值的变量

```
int main()
{
    int a = 10;
    int * p = NULL;
    p = &a;
    printf("a is %d, &a is %p\n\n", a, &a);
    printf("p is %p, *p is %d, &p is %p\n", p, *p, &p);
    return 0;
}
```

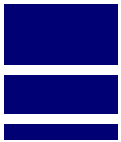
定义指针变量，并用NULL对其初始化。
永远不要使用未初始化的指针!!!



Watches (new)		
Function arguments		
Locals		
a	10	
p	0x0	
&a	(int *) 0x61fefc	int *
*&a	10	int
*p	Cannot access memory at address 0x0	int
&p	(int **) 0x61fef8	int **

p = &a

Watches (new)		
Function arguments		
Locals		
a	10	
p	0x61fefc	
&a	(int *) 0x61fefc	int *
*&a	10	int
*p	10	int
&p	(int **) 0x61fef8	int **

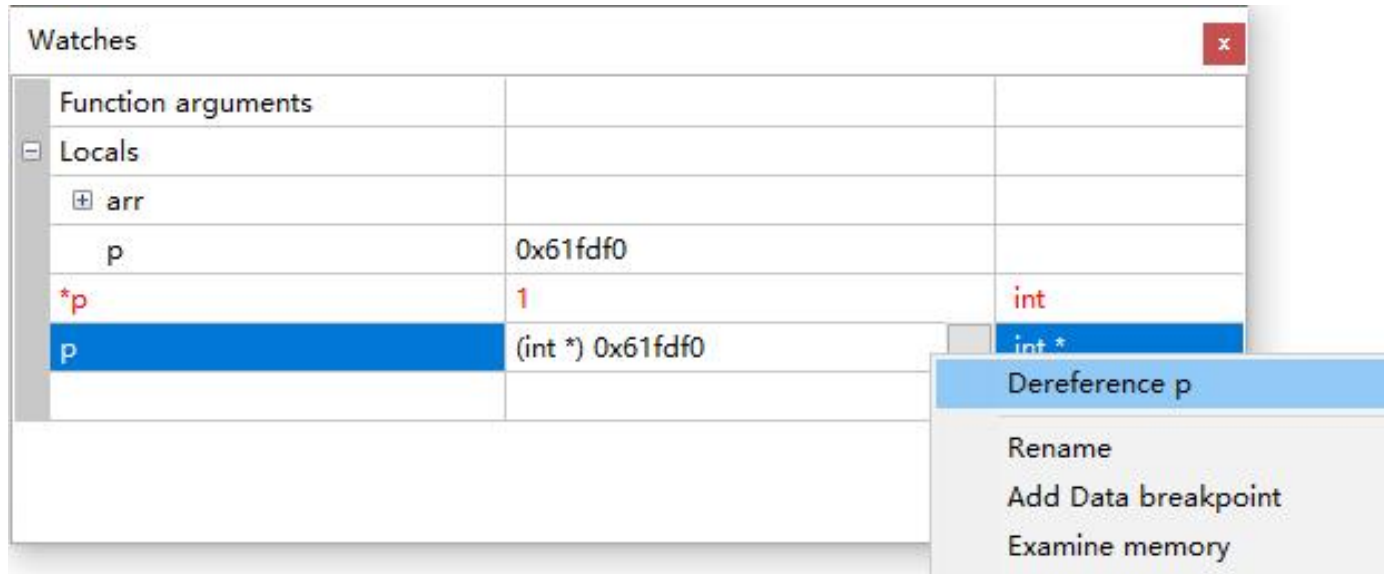


实验原理

➤ 如何查看指针指向的变量的值？

调试小技巧：

1. 直接在watches空行键入*p
2. 在watches空行键入p，在此行点击右键，选择 “Dereference p”



实验原理

➤ 如何规范地使用指针？

指针必须先初始化后再使用！

指针变量在创建的同时应当被初始化，要么将指针设置为NULL，要么让它指向合法的内存。

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char *p1;
7      printf("p1 = %p\n", p1);
8      scanf("%s", p1);
9
10     return 0;
11 }
12
```

Logs & others

File Line Message

D:\exp7demo... == Build: Debug in exp7demo (compiler: GNU GCC Compiler) ==

D:\exp7demo... In function 'main':

warning: 'p1' is used uninitialized in this function [-Wuninitialized]

== Build finished: 0 error(s), 1 warning(s) (0 minute(s), 1 second(s))

== Run: Debug in exp7demo (compiler: GNU GCC Compiler) ==


D:\exp7demo\bin\Debug\exp7demo.exe

p1 = 0000000000000010

hello

Process returned -1073741819 (0xC0000005)

Press any key to continue.



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char *p2 = NULL;
7      char name[6];
8      p2 = name;
9      printf("p2 = %p\n", p2);
10     scanf("%s", p2);
11     printf("%s", p2);
12
13     return 0;
14 }
```

D:\codetest\testout\bin\Debug\testout.exe


p2 = 0060FEF6

lily

lily

Process returned 0 (0x0)

Press any key to continue.



```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      char *p2 = NULL;
7      printf("p2 = %p\n", p2);
8      scanf("%s", p2);
9
10     return 0;
11 }
12
```

Logs & others

File Line Message

D:\exp7demo... == Build: Debug in exp7demo (compiler: GNU GCC Compiler) ==

D:\exp7demo... gcc.exe -Wall -g -c D:\exp7demo\main.c -o obj\Debug\main.o

gcc.exe -o bin\Debug\exp7demo.exe obj\Debug\main.o

Output file is bin\Debug\exp7demo.exe with size 54.10 KB

Process terminated with status 0 (0 minute(s), 1 second(s))

0 error(s), 0 warning(s) (0 minute(s), 1 second(s))


D:\exp7demo\bin\Debug\exp7demo.exe

p2 = 0000000000000000

hello

Process returned -1073741819 (0xC0000005)

Press any key to continue.





实验原理

➤ 如何用指针操纵一维数组?

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

int main()
{
    int i;
    int score[3] = {70,80,90};

    for ( i =0; i<3; i++)
        printf("score[%d] = %d\n",i,score[i]);
    printf("\n");

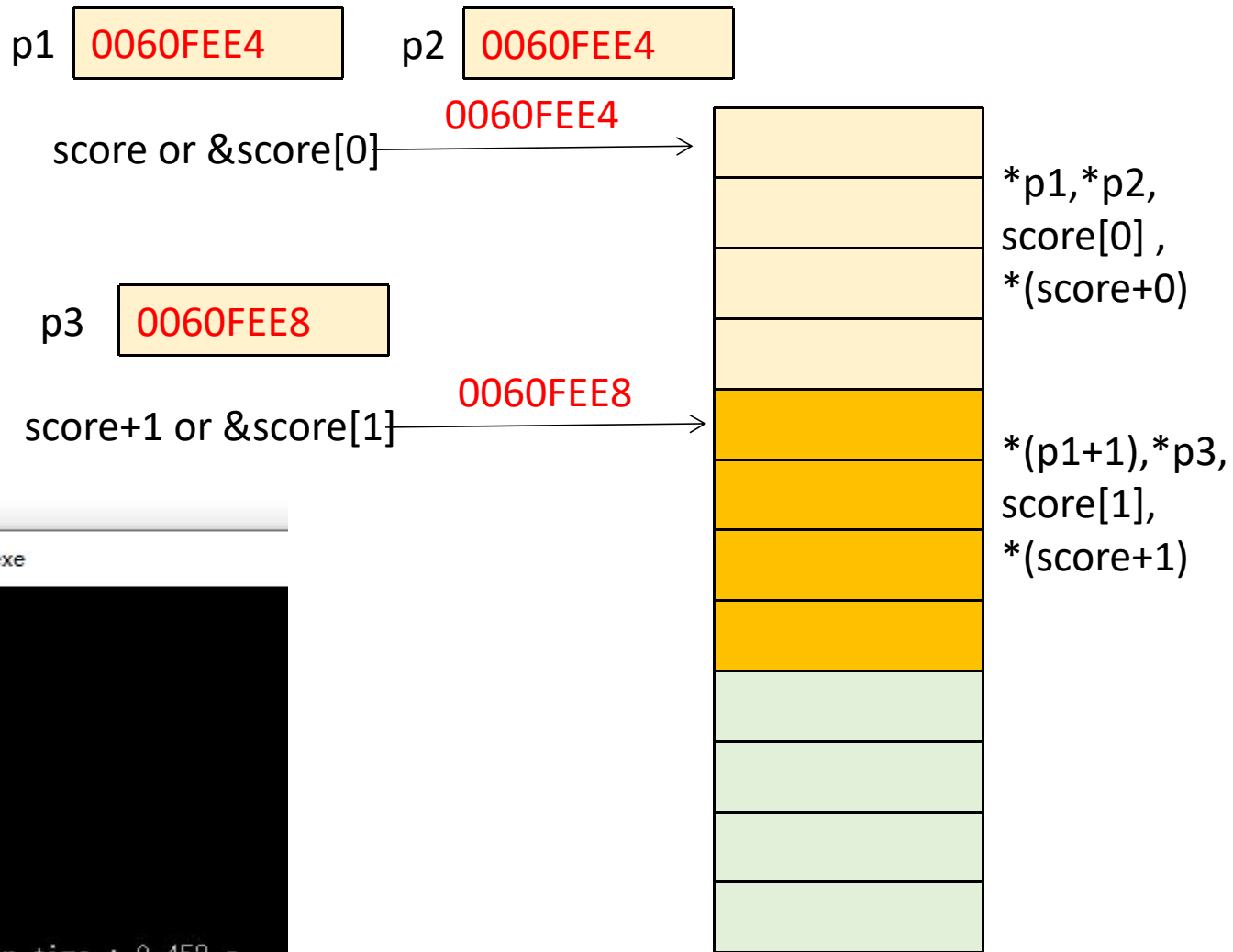
    int *p1 = score;
    int *p2 = &score[0];
    int *p3 = p1+1;

    printf("p1 = %p\n",p1);
    printf("p2 = %p\n",p2);
    printf("p3 = %p\n",p3);
    printf("\n");

    printf("*p1 = %d\n",*p1);
    printf("*p2 = %d\n",*p2);
    printf("*p3 = %d\n",*p3);

    return 0;
}
```

```
D:\codetest\testouot\bin\Debug\testouot.exe
score[0] = 70
score[1] = 80
score[2] = 90
p1 = 0060FEE4
p2 = 0060FEE4
p3 = 0060FEE8
*p1 = 70
*p2 = 70
*p3 = 80
Process returned 0 (0x0)   execution time : 0.452 s
Press any key to continue.
```





实验内容 – 编程题1

【编程题1】 大数相加

使用**一维数组**实现两个正整数相加，输出它们的和。两个正整数位数最少为20位，不超过50位，**注意**两个加数的位数不一定相等。

【输入】 两个正整数

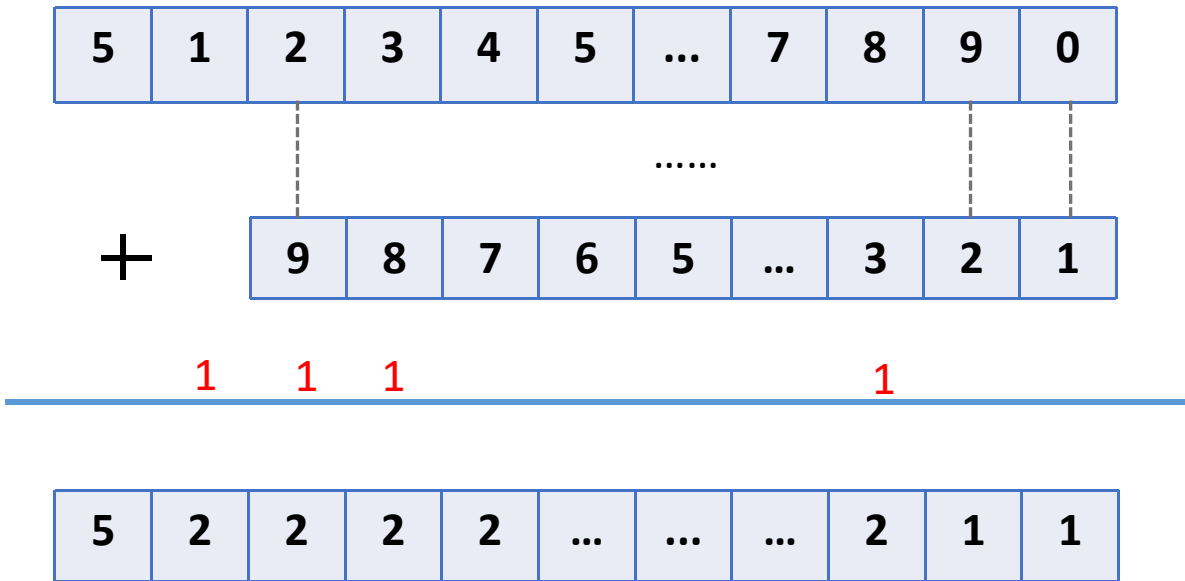
【输出】 两个正整数的和

【样例输入】

123456789341341234567
11135439503091234567

【样例输出】

134592228844432469134





实验内容 – 编程题2

【编程题2】 字符串数字提取

将用户输入的由数字字符和非数字字符组成的字符串中的数字提取出来（字符串长度 ≤ 100 ）。

例如：输入asd123,34fgh_566kkk789，则产生的数字分别是123、34、789。

【输入】 由数字字符和非数字字符组成的字符串

【输出】 "%10d\n"

【样例输入】

Abc123def456ghi111bbbccc99go100

【样例输出】

123
456
111
99
100

```
D:\code\c\2022\stucode\bin\Debug\stucode.exe
Abc123def456ghi111bbbccc99go100
123
456
111
99
100
Process returned 0 (0x0)   execution time : 0.626 s
Press any key to continue.
```



实验内容 – 编程题3

【编程题3】矩阵的转置

将输入的 $N*M$ 的矩阵转化成 $M*N$ 的矩阵， N 和 M 由用户输入。

【输入】

第一行，两个正整数，中间用空格间隔，分别表示矩阵
行和列元素个数，之后输入 $N*M$ 矩阵，元素之间用空格
间隔。

【输出】

$M*N$ 矩阵，元素之间用空格间隔。

【样例输入】

3 4
3 -1 -1 2
-5 1 3 -4
1 -5 3 -3

【样例输出】

3 -5 1
-1 1 -5
-1 3 3
2 -4 -3

$$A = \begin{bmatrix} 3 & -1 & -1 & 2 \\ -5 & 1 & 3 & -4 \\ 1 & -5 & 3 & -3 \end{bmatrix}$$



$$A^T = \begin{bmatrix} 3 & -5 & 1 \\ -1 & 1 & -5 \\ -1 & 3 & 3 \\ 2 & -4 & -3 \end{bmatrix}$$

青 / 春 / 不 / 老 / 梦 / 想 / 永 / 在

FEEL THE MEANING OF THE TRIP

DREAM

MY DREAM WILL NEVER STOP

请同学们开始实验



哈爾濱工業大學(深圳)
HARBIN INSTITUTE OF TECHNOLOGY, SHENZHEN

GO!
TAKE YOU ON A TRIP

探索 从未停止