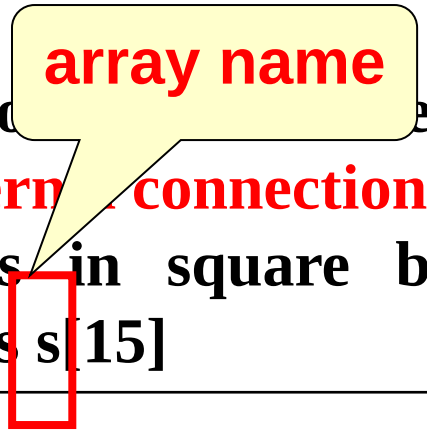
Computer programming

Lecture 5

- If there are 1000 students and each student has a score, the average score of these 1000 students needs to be calculated.
- Using $s_1, s_2, s_3, \ldots\ldots, s_{1000}$ to [illegible] grades of each student can reflect intern[illegible] connections.
- C language uses numbers in square brackets to represent subscripts, such as s[15]



- An array is a collection of ordered data. The arrangement of data in an array follows a certain pattern, with subscripts representing the sequence number of the data in the array
- Using an array name and index to uniquely identify elements in an array
- Each element in the array belongs to the same data type

The general form of declaring a one-dimensional array is:

Type operator array name [constant expression];

The naming convention for array names is the same as for variable names.

Example

```
int a [10];
int score[1000];
#Define N 10
long number [N];/* Defined a long integer array number
                   with 10 elements*/
int b [6 * N];/* Defined an integer array b with 60 elements*/
int a[4+6];  //legal
int n=10;
int a[n];      //illegal
```

```
#include <stdio.h>
 main()
{
    float salary[10];
    int i;
    for (i=0; i<=9; i++)
    {
        printf("%p\n",&salary [i]);
     }
     printf("%d\n",sizeof(salary));
     printf("%p\n",&salary);
}
```

```
000000000022FE20
000000000022FE24
000000000022FE28
000000000022FE2C
000000000022FE30
000000000022FE34
000000000022FE38
000000000022FE3C
000000000022FE40
000000000022FE44
40
000000000022FE20

Process returned 0 (0x0)   execution time : 0.021 s
Press any key to continue.
```

Salary+=1000; //wrong

The representation of referencing array elements is:

Array name [subscript]

Example

'a[i]'

int i=10

'a[i] ' , 'a[2*i] '

int a[10];

10 elements： a[0],a[1],a[2],…,a[9]

a[0]	a[1]	a[2]	a[3]	…	a[7]	a[8]	a[9]

[question] assigning values of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 to 10 array elements in sequence and then outputting them in reverse order

Solution ideas:

- **Define an array with a length of 10, which is defined as an integer**
- **The value to be assigned is from 0 to 9, which can be assigned using a loop**
- **Use a loop to output these 10 elements in descending order**

```
#include <stdio.h>
int main()
 {  int i,a[10];
    for (i=0; i<=9;i++)
       a[i]=i;
    for(i=9;i>=0; i--)
       printf("%d ",a[i]);
    printf("\n");
    return 0;
 }
```

Set the values of a[0]～a[9] to 0～9

a[0]a[1]a[2]a[3]a[4]a[5]a[6]a[7]a[8]a[9]

0	1	2	3	4	5	6	7	8	9

```
#include <stdio.h>
int main()
 {  int i,a[10];
     for (i=0; i<=9;i++)
         a[i]=i;
     for(i=9;i>=0; i--)
         printf("%d ",a[i]);
     printf("\n");
     return 0;
 }
```

```
9 8 7 6 5 4 3 2 1 0
```

Output a[9] firstly, output a[0] lastly

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
0	1	2	3	4	5	6	7	8	9

[question] Calculate the total score of 50 students' Chinese language scores and output it.

```
#include "stdio.h"
void main()
{
  int score[50]，sum=0;
  for(int i=0;i<49;i++)
     {
       scanf("%f",&score[i]);
       sum+=score[i];
       }
    printf("%d\n",sum);
 }
```

Initialization of one-dimensional arrays

Assign values to each array element while declaring the array

```
int a[10]={0,1,2,3,4,5,6,7,8,9};
int a[10]={0,1,2,3,4}; //equivalent to
int a[10]={0,1,2,3,4,0,0,0,0,0};
int a[10]={0,0,0,0,0,0,0,0,0,0}; //equivalent to
int a[10]={0};
int a[5]={1,2,3,4,5};    // can be written as
int a[ ]={1,2,3,4,5};
```

[question] using an array to calculate the first 20 terms of the Fibonacci sequence

```
#include <stdio.h>
int main()
 {  int i;  int f[20]={1,1};
    for(i=2;i<20;i++)
       f[i]=f[i-2]+f[i-1];
    for(i=0;i<20;i++)
   {   if(i%5==0) printf("\n");
```

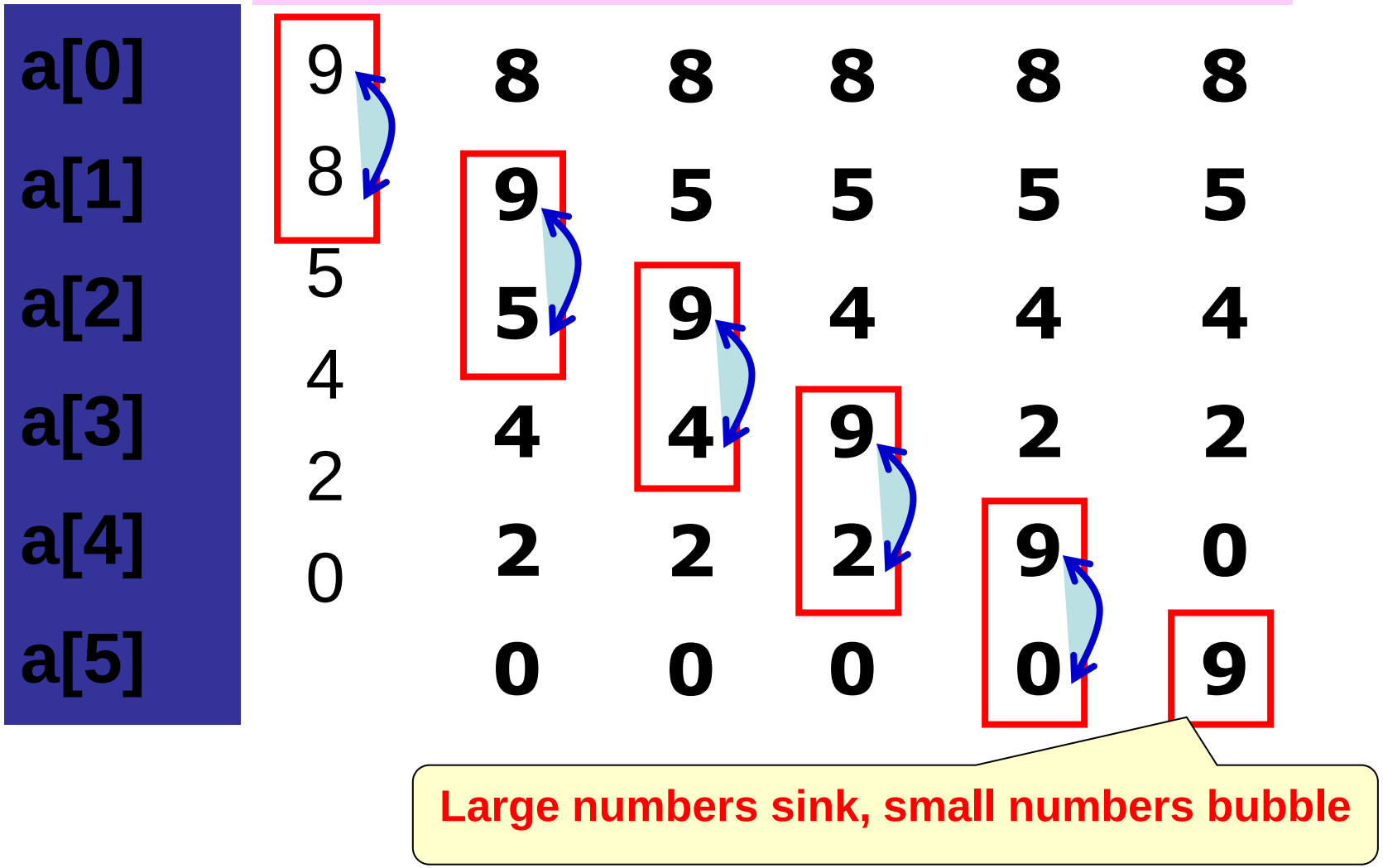
```
    1     1     2     3     5
    8    13    21    34    55
   89   144   233   377   610
  987  1597  2584  4181  6765
```

```
   return 0;
}
```

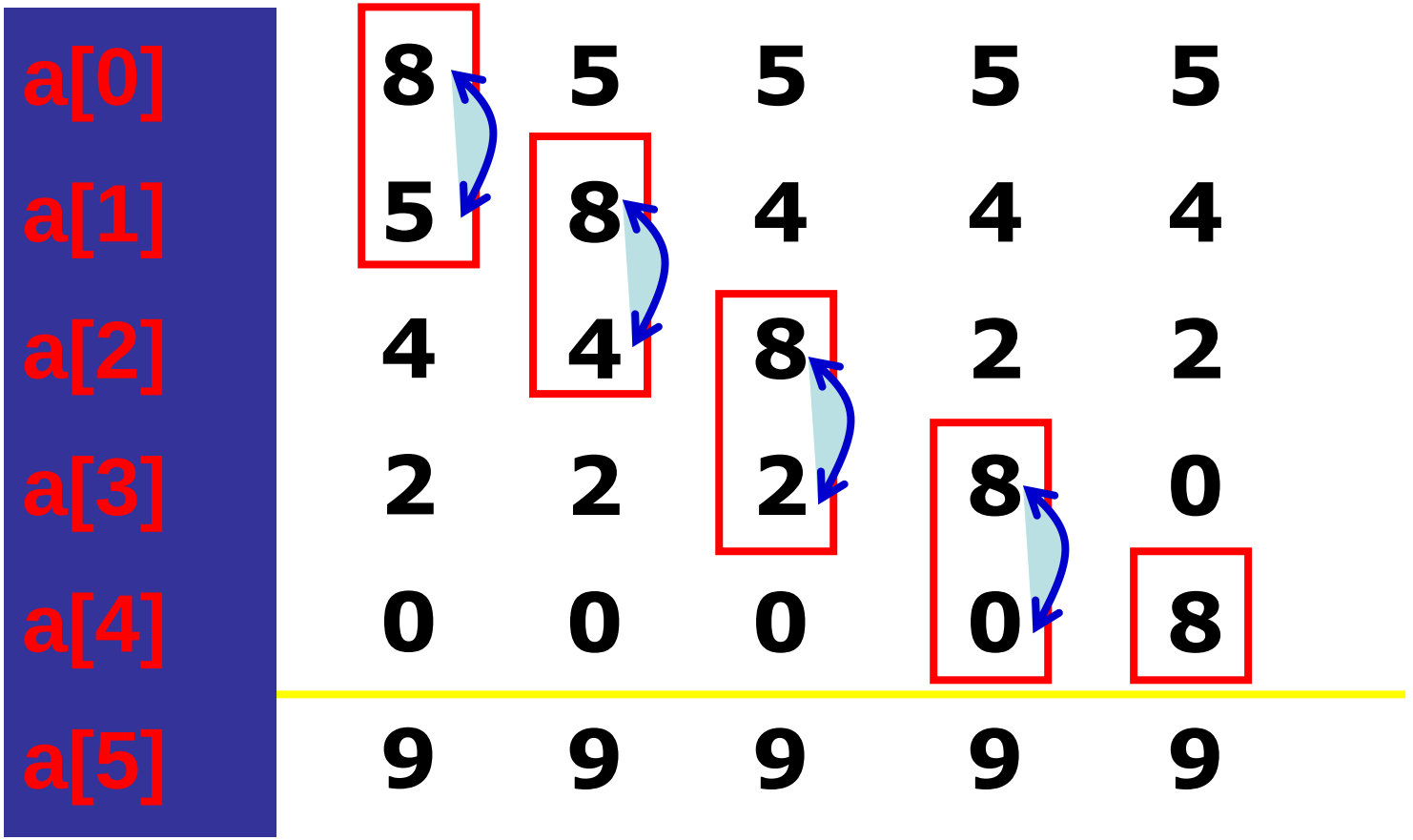
[question] There are six regions with an area that requires them to be arranged in descending order.

Solution ideas:
There are two types of sorting rules: one is "ascending", from small to large; Another type is' descending ', from largest to smallest
Abstract the question as: "Sort n numbers in ascending order "
Sort by bubble method

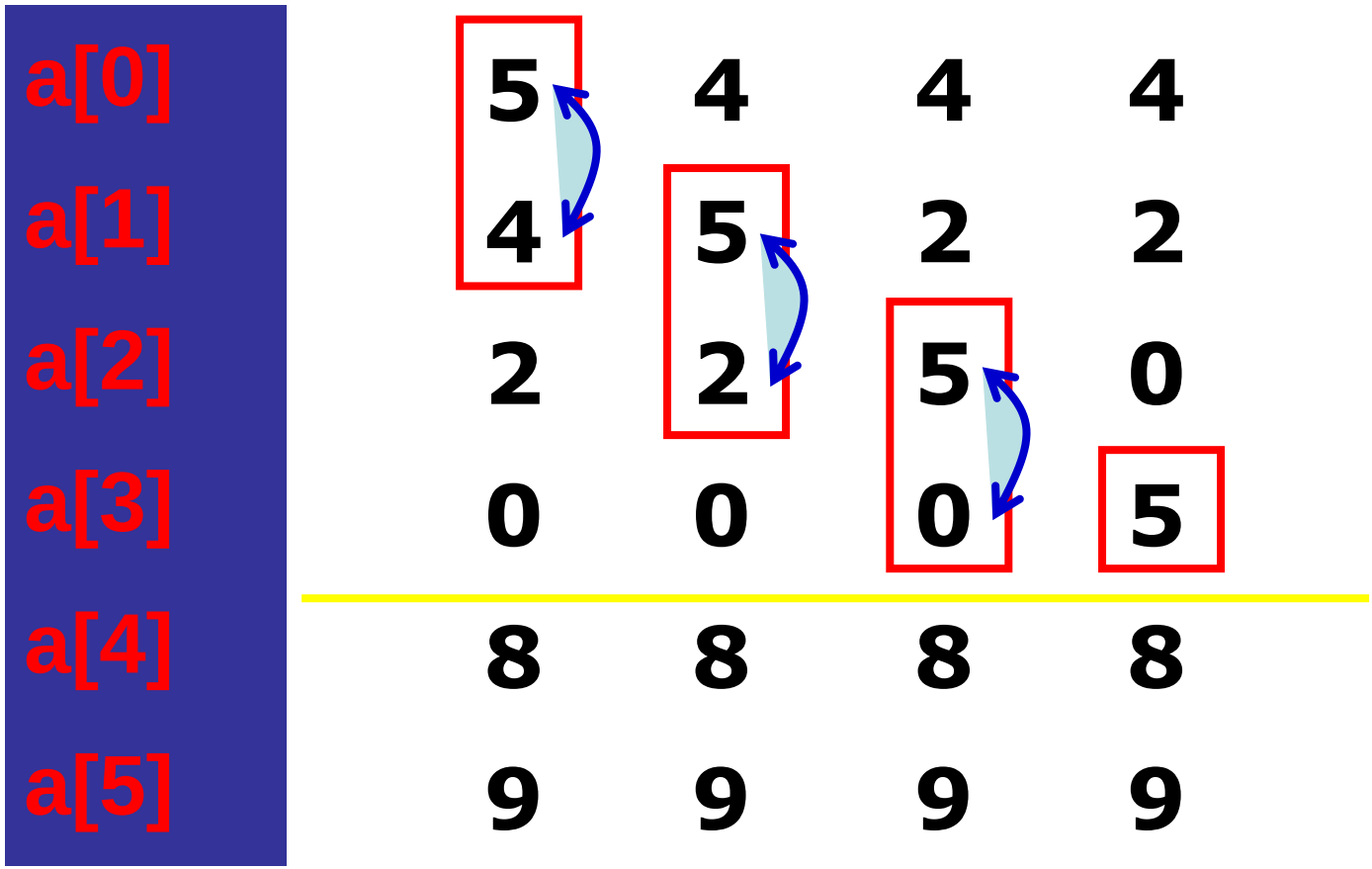
```
for(i=0;i<5;i++)
    if (a[i]>a[i+1])
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```



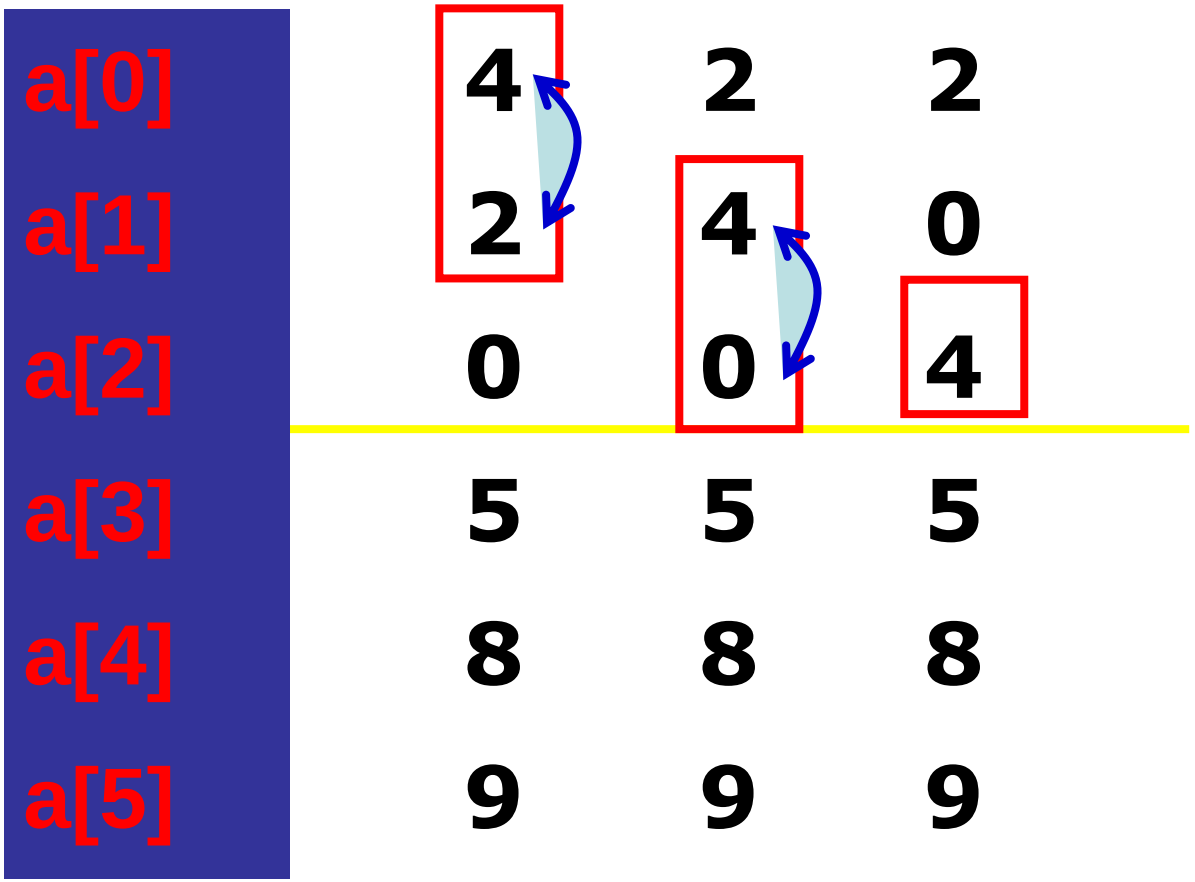
```
for(i=0;i<4;i++)
    if (a[i]>a[i+1])
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```



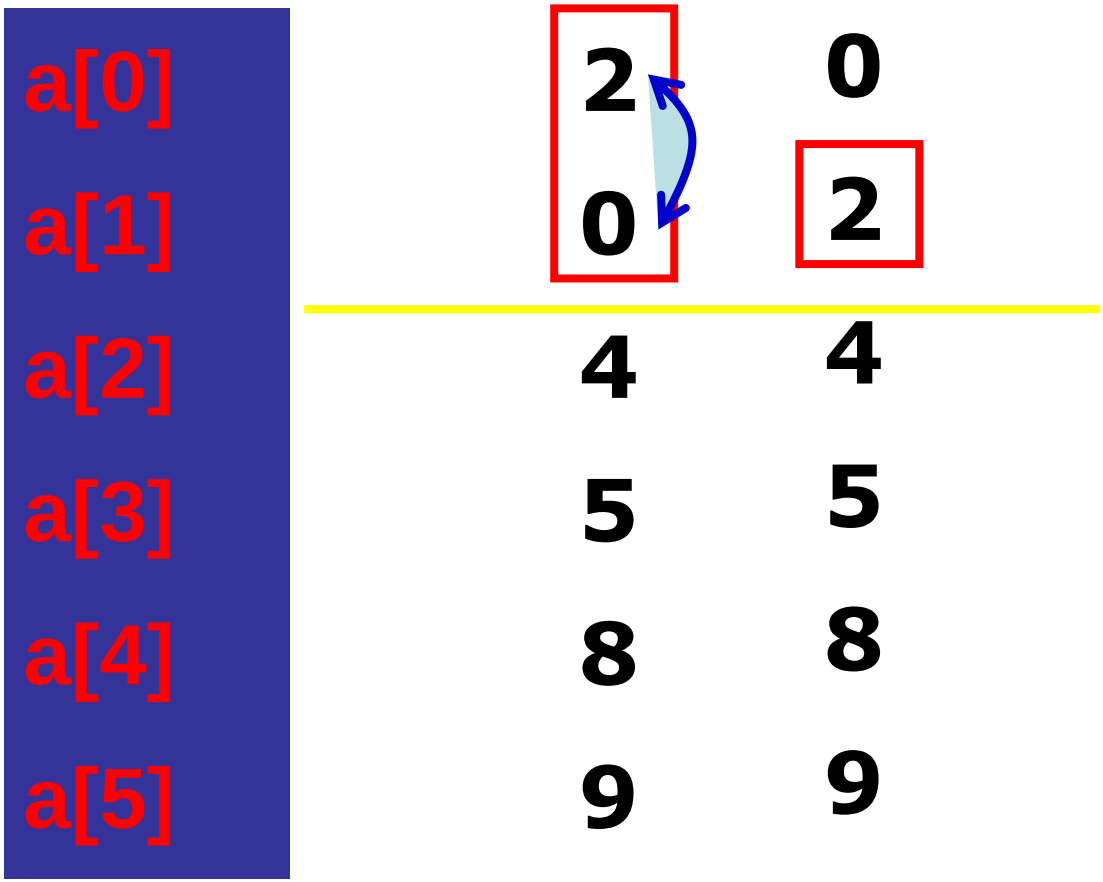
```
for(i=0;i<3;i++)
    if (a[i]>a[i+1])
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```



```
for(i=0;i<2;i++)
    if (a[i]>a[i+1])
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```



```
for(i=0;i<1;i++)
    if (a[i]>a[i+1])
    { t=a[i];a[i]=a[i+1];a[i+1]=t; }
```



```
for(i=0;i<5;i++)
    if (a[i]>a[i+1])
    { ……}
```

```
for(i=0;i<4;i++)
    if (a[i]>a[i+1])
    { ……}
```

……

```
for(i=0;i<1;i++)
    if (a[i]>a[i+1])
    { ……}
```

```
for(j=0;j<5;j++)
    for(i=0;i<5-j;i++)
        if (a[i]>a[i+1])
        { ……}
```

```
int a[10]
printf("in
for (i=0;
printf("\n");
for(j=0;j<9;j++)
    for(i=0;i<9-j;i++)
      if (a[i]>a[i+1])
        {t=a[i];a[i]=a[i+1];a[i+1]=t;}
printf("the sorted numbers :\n");
for(i=0;i<10;i++)  printf("%d ",a[i]);
printf("\n");
```

```
input 10 numbers :
34 67 90 43 124 87 65 99 132 26

the sorted numbers :
26 34 43 65 67 87 90 99 124 132
```

Two-dimensional array

	member 1	member 2	member 3	member 4	member 5	member 6
Team 1	2456	1847	1243	1600	2346	2757
Team 2	3045	2018	1725	2020	2458	1436
Team 3	1427	1175	1046	1976	1477	2018

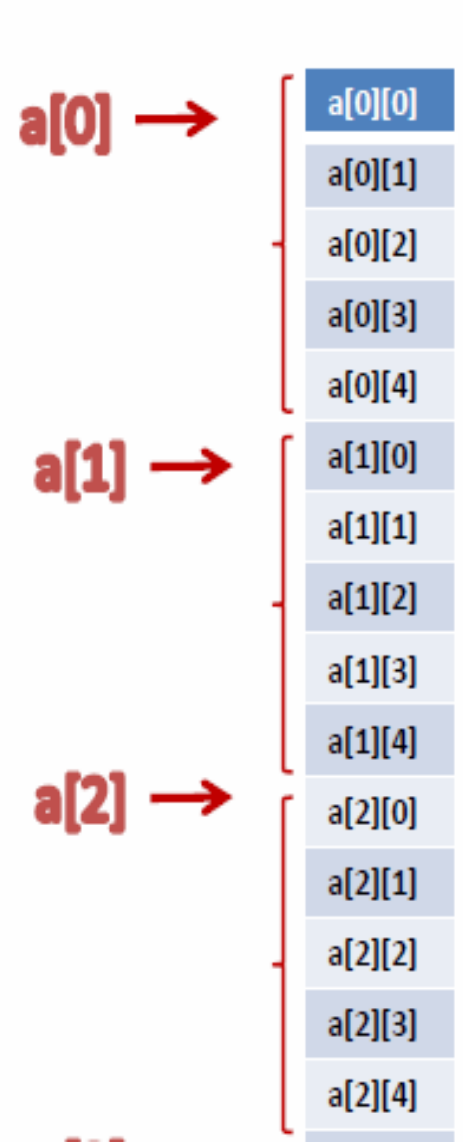
float pay[3][6];

The general form of a two-dimensional array definition is

Type operator array name [constant expression] [constant expression];

```
example
float a [3] [4];
#define R 4
#defint C 7
float b [R] [C];
```

a[0]	a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1]	a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2]	a[2][0]	a[2][1]	a[2][2]	a[2][3]



[question] A study group consists of 5 people, each with exam scores for 4 courses. Calculate the average and total average scores for each subject in the group.

```
int i, j;  //2D array subscript
int sum = 0;  //The total score of the current subject
int average;  //Total Average Score
int aver[4];  //Average score of each subject
int score[5][4]; // save the grades of each student in various subjects
```

Name	Chinese	Mathematics	Physics	English
李俊	80	90	65	77
马丽	90	67	78	67
周云	93	56	70	90
王强	70	60	80	80
李煜	55	60	47	60

The general format to access element of 2D array :

array name [first dimension index] [second dimension index]

example

a[0][4]

a[3*i+4] [j+1] Both i and j are integer variables and have been assigned values.

```
b[1][2]=a[2][3]/2
for(i=0;i<m;i++)
    { printf("%d,%d\n",a[i][0],a[0][i]);}
```

Initialization of two-dimensional arrays

```
int a[3][4]={{1,2,3,4},{5,6,7,8},
                    {9,10,11,12}};
int a[3][4]={1,2,3,4,5,6,7,8,9,10,11,12};
int a[3][4]={{1},{5},{9}}; //equivalent to
int a[3][4]={{1,0,0,0},{5,0,0,0}, {9,0,0,0}};
char C[3][4] = {'a', 'c'}; //equivalent to
char a[3][4]={{'a', '\0', '\0', '\0'},{'c', '\0', '\0', '\0'},
                                {'\0', '\0', '\0', '\0'}};
int a[ ][4]={1,2,3,4,5,6,7,8,9,10,11,12};
int a[3][4]={{},{},{}};//wrong
```

Assign values after defining an array

Case 1: If the values of array elements are generated during program operation, they can be assigned according to the situation after definition.

```
example
int a[5][5];
a[0][0]=1; a[0][1]=2; a[0][2]=3; a[0][3]=4; a[0][4]=5;
a[1][0]=6; a[1][1]=7; a[1][2]=8; a[1][3]=2; a[1][4]=4;
.....
```

Case 2: The value of the array element is entered by the user through the keyboard, due to the large number of array elements, a loop can be used to traverse the array and complete the assignment

```
example:
for (int i=0; i<4; i++)
    {for (int j=0; j<4; j++)
        {scanf ('% d',&a [i] [j]);}}
```

[example] swapping the rows and columns of a two-dimensional array and saving the result in another two-dimensional array

Solution ideas:

- **Two arrays can be defined: array a consists of 2 rows and 3 columns, storing a specified number of 6**
- **Array b has 3 rows and 2 columns, and is initially unassigned**
- **Store the element a [i] [j] in the array a into the element b [j] [i] in the array b**
- **Complete with nested for loops**

```
#include <stdio.h>
void main()
 {  int a[2][3]={{1,2,3},{4,5,6}};
    int b[3][2],i,j;
    printf("array a:\n");
    for (i=0;i<=1;i++)
    { for (j=0;j<=2;j++)
      {  printf("%5d",a[i][j]);
         b[j][i]=a[i][j];
       }
      printf("\n");
     }
   printf("array b:\n");
   for (i=0;i<=2;i++)
   {  for(j=0;j<=1;j++)
         printf("%5d",b[i][j]);
      printf("\n");
   }}
```

Process each element in a row of a

Process a column element in A

Output each element of a

Assign the value of element a to the corresponding element b

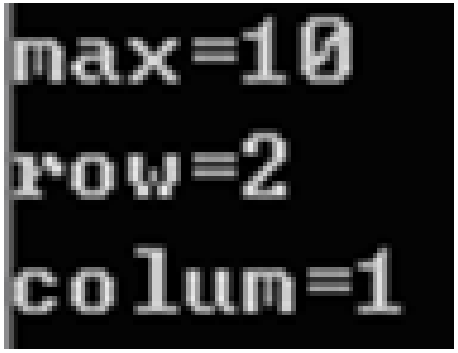
Output each element of b

[question] There is a 2-dimensional array with 3 rows and 4 columns, which program to find the value of the element with the highest value, as well as its row and column numbers.

Solution ideas:

- **declare that a variable stores the value of a certain element**
- **compare other elements with it. The larger value is saved, and the saved value continues to be compared with other elements.**
- **Until all elements participate in the comparison, the final saved value is the maximum value.**

```
#include <stdio.h>
void main()
{
  int i,j,row=0,colum=0,max;
  int a[3][4]={{1,2,3,4},{9,8,7,6},
                         {-10,10,-5,2}};
  max=a[0][0];
  for (i=0;i<=2;i++)
    {
      for (j=0;j<=3;j++)
      {
            if (a[i][j]>max)
               { max=a[i][j];  row=i;  colum=j; }
       }
     }
  printf("max=%d\nrow=%d\ncolum=%d\n",max,row,colum);
}
```



character array

```
char c[10];
c[0]='I';   c[1]=' ';
c[2]='a';   c[3]='m';
c[4]=' ';   c[5]='h';
c[6]='a';  c[7]='p';
c[8]='p';  c[9]='y';
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
I		a	m		h	a	p	p	y

```
char c[10]={'I',' ','a','m',' ','h','a','p','p','y'};
```

c[0]	c[1]	c[2]	c[3]	c[4]	c[5]	c[6]	c[7]	c[8]	c[9]
I		a	m		h	a	p	p	y

```
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

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

[question] Output a known string.

Solution ideas:

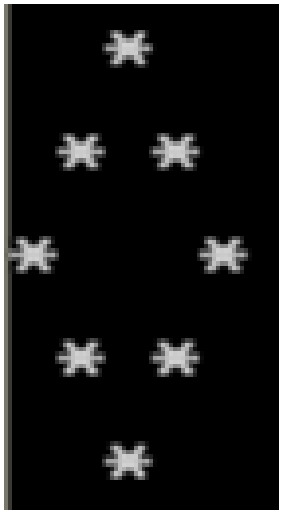
- **Define a character array and assign initial values to it**
- **Output the characters in this character array one by one using a loop**

```
#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;
}
```

```
I am a student.
```

[question] Please output a diamond with a star on the screen

```
#include <stdio.h>
int main()
{ char diamond[][5]={{' ',' ','*'},
            {' ','*',' ','*'},{'*',' ',' ',' ','*'},
             {' ','*',' ','*'},{' ',' ','*'}};
  int i,j;
  for (i=0;i<5;i++)
    {for (j=0;j<5;j++)
        printf("%c",diamond[i][j]);
     printf("\n");
    }
  return 0;
}
```



The relationship between strings and character arrays

- In C language, strings are processed as character arrays
- The concern is the effective length of the string rather than the length of the character array
- In order to determine the actual length of a string, the C language specifies the string end flag '\ 0'

```
char c[]={”I  am  happy”};
    can be written as
char c[]=”I  am  happy”;
      equivalent to
char c[11]={”I  am  happy”};
```

```
char c[10]={”China”};
    can be written as
char c[10]=”China”;
```

Starting from c [5], all element values are \0

```
printf(”%s”,c);
```

There are two methods for inputting and outputting character arrays:

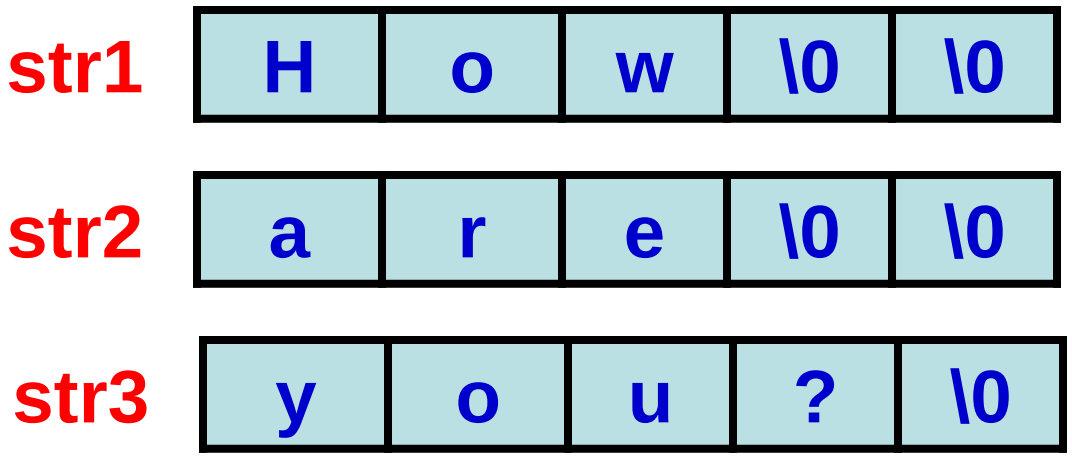
1. **Character by character input and output (% c)**
2. **Input and output the entire string at once (% s)**

- **The output character does not include the closing character ' \0 '**
- **When outputting a string with %s, the output item in the printf function is a character array name, not an array element name**
- **If a character array contains multiple ' \0', the output ends when the first ' \0' is encountered**
- **input a string using the scanf function, the input item c in the scanf function is a defined character array name, and the input string should be shorter than the length of the defined character array**

char c [6]; scanf ("% s", c); China ↙

The system automatically adds a ' \0' after China

```
char str1[5],str2[5],str3[5];
scanf("%s%s%s",str1,str2,str3);
How are you? ↙
```



string-handling functions

In the C function library, some functions are provided specifically for handling strings, making it easy to use

1.Puts function - a function that outputs strings

general form is:

puts (character array)

The function is to output a string to the terminal

```
char str [20]="China";
puts (str);
```

Output China

2. gets function - a function that inputs a string

general form is:

gets (character array)

The function is to input a string into a character array

```
char str [20];
gets (str);
```

computer ↙

3. strcat function - string concatenation function

general form is:

strcat (character array 1, character array 2)

Its function is to connect two strings, connect string 2 to string 1, and place the result in character array 1

```
char str1[30]=”People”;
char str2[]=”China”;
printf(”%s”, strcat(str1,str2));
```

output：PeopleChina

4. strcpy and strncpy functions - string copying

general form is:

strcpy (character array 1, string 2)

The function is to copy string 2 into character array 1

```
Char str1 [10], str2 []="China";
Strcpy (str1, str2);
```

str1	C	h	i	n	a	\0	\0	\0	\0	\0

```
char str1[10],str2[]=”China”;
strcpy(str1,str2);
```

array name

```
char str1[10],str2[]=”China”;
strcpy(str1,str2); //equivalent to
strcpy(str1,”China”);
```

```
char str1[10],str2[]=”China”;
strcpy(str1,str2);
```

Array name or string constant

```
char str1[10],str2[]=”China”;
   str1=”China”;   //wrong
   str1=str2;        //wrong
```

Strcpy and strncpy functions - string copying

You can use the strncpy function to copy the first n characters in string 2 into character array 1

Strncpy (str1, str2, 2);

The function is to copy the first two characters in str2 into str1, replacing the original first two characters in str1The number of copied characters n should not exceed the original characters in str1

5.Strcmp function - string comparison function

general form is

Strcmp (string 1, string 2)

The function is to compare string 1 and string 2

```
Strcmp (str1, str2);
Strcmp ("China", "Korea");
Strcmp (str1, "Beijing");
```

The rule for string comparison is to compare two strings one by one from left to right, until different characters appear or '\0' is encountered

If all characters are the same, two strings are considered equal

If different characters appear, the comparison result of the first pair of different characters shall prevail

”A”<”B”　　”a”>”A”　　”computer”>”compare”
”these”>”that”　”1A”>”$20”　　”CHINA”>”CANADA”
”DOG”<”cat”　”Tsinghua”>”TSINGHUA”

The comparison result is brought back by the function value
If string 1=string 2, then the function value is 0
If string 1>string 2, then the function value is a positive integer
If string 1<string 2, then the function value is a negative integer

```
if(str1>str2)  printf(”yes”);  //wrong
if(strcmp(str1,str2)>0)
     printf(”yes”);    //correct
```

6. strlen function - a function for measuring the length of a string

general form is:

strlen (character array)

It is a function that tests the length of a string

The value of the function is the actual length in the string

```
char str[10]=”China”;
printf(”%d”,strlen(str));
```

The output result is 5

You can also directly test the length of string constants

```
strlen(”China”);
```

7. strlwr function - Convert to lowercase function

general form is

strlwr (string)

The function replaces uppercase letters in a string with lowercase letters

8. strupr function - Convert to uppercase function

general form is

strupr (string)

The function replaces lowercase letters in a string with uppercase letters

[question] There are three strings in the following example: China, Japan, and India. require finding the largest one

[question] There are three strings in the following example: China, Japan, and India. require finding the largest one

```
#include<stdio.h>
#include<string.h>
int main ( )
{char str[3][10]; char string[10]; int i;
  for (i=0;i<3;i++)    gets (str[i]);
  if (strcmp(str[0],str[1])>0)
            strcpy(string,str[0]);
  else
            strcpy(string,str[1]);
  if (strcmp(str[2],string)>0)
     strcpy(string,str[2]);
  printf("\nthe largest:\n%s\n",string);
  return 0;
}
```