Recitation Class 07 for VG101

Date: 2012 / 11 / 05

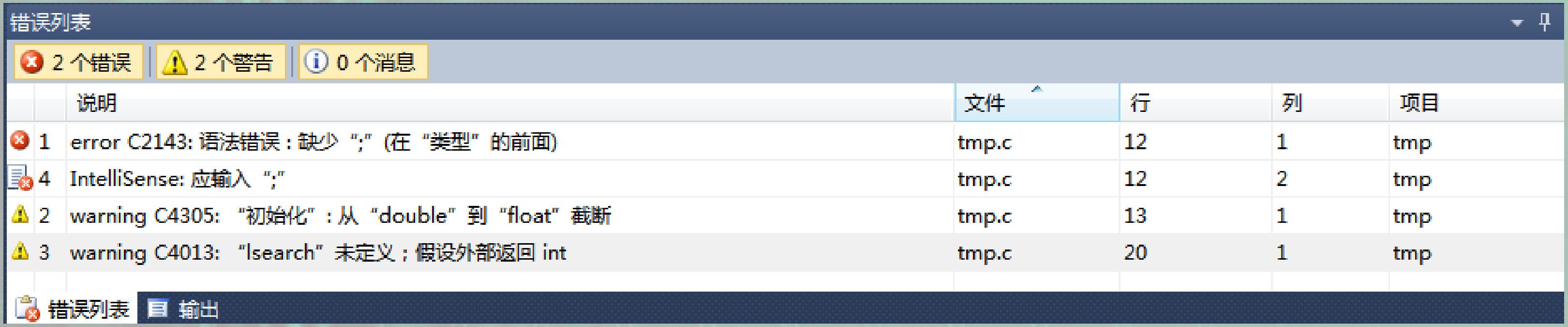
Wang Qian

feedback & announcement

- Take your flash disk during the lab time.
- LAB 05
  - Have a big picture of C.
- HWK04
  - Please pay attention to the usage of correct place holder.
  - Check the priority of the operators.
  - Only submit .c file instead of .sln or other files.
  - Please name it as .c instead of .cpp.

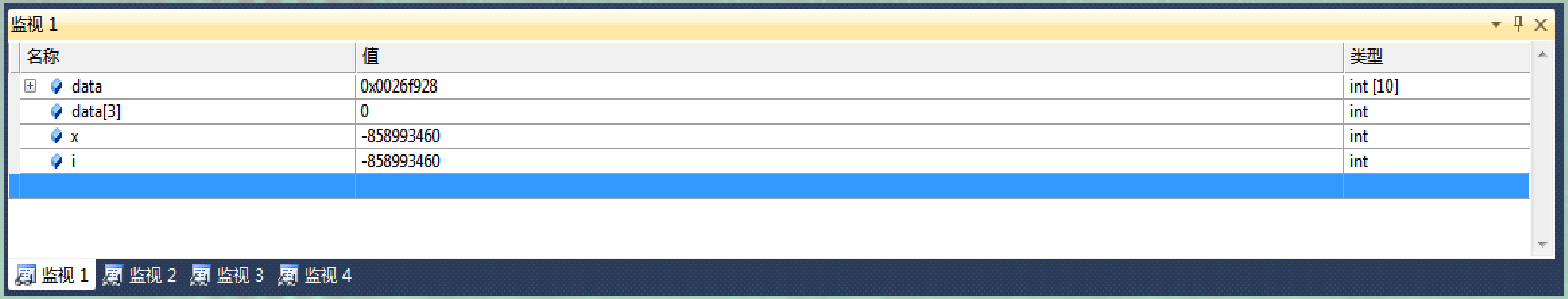
More about Debug

- Recall the keyboard shortcuts introduced before.
- Error list: ctrl + \ E (when compiling)



	说明	文件	行	列	项目
1	error C2143: 语法错误 : 缺少 “;” (在 “类型” 的前面)	tmp.c	12	1	tmp
4	IntelliSense: 应输入 “;”	tmp.c	12	2	tmp
2	warning C4305: “初始化” : 从 “double” 到 “float” 截断	tmp.c	13	1	tmp
3	warning C4013: “lsearch” 未定义；假设外部返回 int	tmp.c	20	1	tmp

- Watch: ctrl + alt + w 1,2,3,4 (when debugging)



名称	值	类型
data	0x0026f928	int [10]
data[3]	0	int
x	-858993460	int
i	-858993460	int

Array

- A sequence of data with the same type.
  - Recall the vector and matrix in Matlab.
- Declaration
  - data_type name[size]
- Initialization
  - Recall the importance of initialization.
  - data_type name[] = {value,value,value,…}
  - data_type name[size] = {value,value,value,…}
  - Fill the remaining locations with 0 automaticly

Array

- Attention:
  - The index will begin and will only begin from 0.
  - The index of the last element is “size - 1”.
  - Check the boundary when using the array.
  - Has a very close relation with pointer.
  - For higher dimension, we use separate square brackets.
  - e.g. int data[10][20];

String

- Each element is a character.
- Declaration
  - char name[size];
- Initialization
  - char stra[] = "Hello";
  - char strb[] = {"Hello"};
  - char strc[] = {'H','e','l','l','o','\0'};
  - ~~char strd[] = {'H','e','l','l','o'};~~
- Only the first three ones can be regarded as strings.
- '\0', whose ASCII is 0, should be the last element.
- '\0' means false.

String

- Input:
  - – gets(stra);
  - – scanf("%s%s",stra,strb);
- Output:
  - – puts(stra);
  - – printf("%s\n",stra);
  - – printf(stra);
- Quiz:
  - – Record a whole sentence in a string.
  - – Record separate words in separate strings.
  - – Print two sentences in one line.

String

- #include <string.h>
- strcmp(stra,strb)
  - strcmp("123","12345") == -1
  - strcmp("12345","123") == 1
  - strcmp("123","123") == 0
- strcat(stra,strb);
- strcpy(stra,strb);
- strlen("12345") == 5
  - Though the last element is “\0”, the answer is 5 instead of 6.

Algorithm

- Program = Algorithm + Data Structure
- Please review the following algorithms covered before:
  - Bubble sort
  - Check the prime (and how to optimize)
  - Find the gcd and lcm (compare with brute-force)
- Other algorithms you should know:
  - Linear search
  - Binary search
  - Selection sort
  - Merge sort (you don't need to implement it)
  - Newton's method (Calculus 5th ed. Section 4.9)

Search Algorithm

- Task:
  - Find a particular element in an array (first appearance).
- Strategy:
  - Linear search: easiest and not very fast
  - Binary search: much faster than the previous one
  - Hash search: difficult to analyze but easy to implement
  - Binary search tree: a complicated data structure
  - (May provide the source code for you directly during the lab after you learn pointer.)

Linear Search

- Basic idea:
  - check each element one by one.
- Pseudo-code:

```
FOR i := 1 to N DO
    IF (data[i] = x) THEN
        The location is determined.
    END IF
END FOR
```

Linear Search

- How to use array as the input argument?
- Recall the usage of for loop and if statement.

```
int lSearch(int data[],int len,int x)
{
        int i;
        for (i = 0; i < len; i++)
                if (data[i] == x)
                        return i;
        return -1;
}
```

Binary Search

- Basic idea:
  - compare the target with the middle one.
- Note:
  - Should be sorted at first.
- Pseudo-code:

```
WHILE (left < right) DO
    mid = (left + right) / 2;
    IF (x > data[mid]) THEN
        left := mid + 1;
    ELSEIF (x <= data[mid]) THEN
        right := mid;
    END IF
END WHILE
```

Binary Search

- Recall the usage of while loop and if statement.

```
int bSearch(int data[],int len,int x)
{
    int left = 0,right = len - 1,mid;
    while (left < right)
    {
        mid = (left + right) >> 1;
        if (x > data[mid])
            left = mid + 1;
        else if (x <= data[mid])
            right = mid;
    }
    if (x == data[left])
        return left;
    else
        return -1;
}
```

Sort Algorithm

- Task:
  - Sort a sequence of elements in order according to one rule.
- Strategy:
  - Bubble sort: slowest but easy to understand (covered before)
  - Selection sort: also slow but still easy to understand
  - Merge sort: should understand the idea at least
  - Count sort: useful when sorting integers of a certain range
  - Bucket sort: useful for uniformly distributed elements
  - (May provide the source code for you directly during the lab after you learn pointer.)

Selection Sort

- Basic idea:
  - For each iteration, find the smallest element in the remaining elements. Then add it to the end of the sorted list.
- Pseudo-code:

```
FOR i := 1 to N-1 DO
    idx := i;
    FOR j := i+1 to N DO
        IF (data[j] < data[idx]) THEN
            idx := j;
        END IF
    END FOR
    swap data[i] and data[idx]
END FOR
```

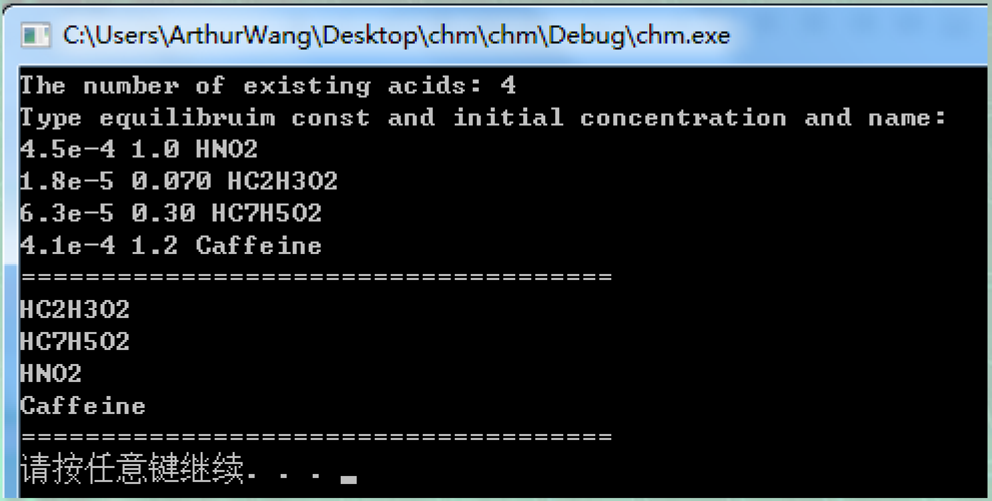
Selection Sort

- How to design a function for this?
  - We will be able to understand this after we learn pointer.

```
void sSort(int data[],int len)
{
    int i,j,idx,tmp;
    for (i = 0; i < N-1; i++)
    {
        idx = i;
        for (j = i+1; j < N; j++)
            if (data[j] < data[idx])
                idx = j;
        tmp = data[i];
        data[i] = data[idx];
        data[idx] = tmp;
     }
}
```

Quiz

- Given name, initial concentration and the equilibrium constant of several weak acids, print the name of each weak acid from the one with the lowest concentration of hydrion to the highest.
- Suppose that $\frac{C_i}{K_a} > 400$



C:\Users\ArthurWang\Desktop\chm\chm\Debug\chm.exe

```
The number of existing acids: 4
Type equilibruim const and initial concentration and name:
4.5e-4 1.0 HNO2
1.8e-5 0.070 HC2H3O2
6.3e-5 0.30 HC7H5O2
4.1e-4 1.2 Caffeine
=======================================
HC2H3O2
HC7H5O2
HNO2
Caffeine
=======================================
请按任意键继续. . .
```