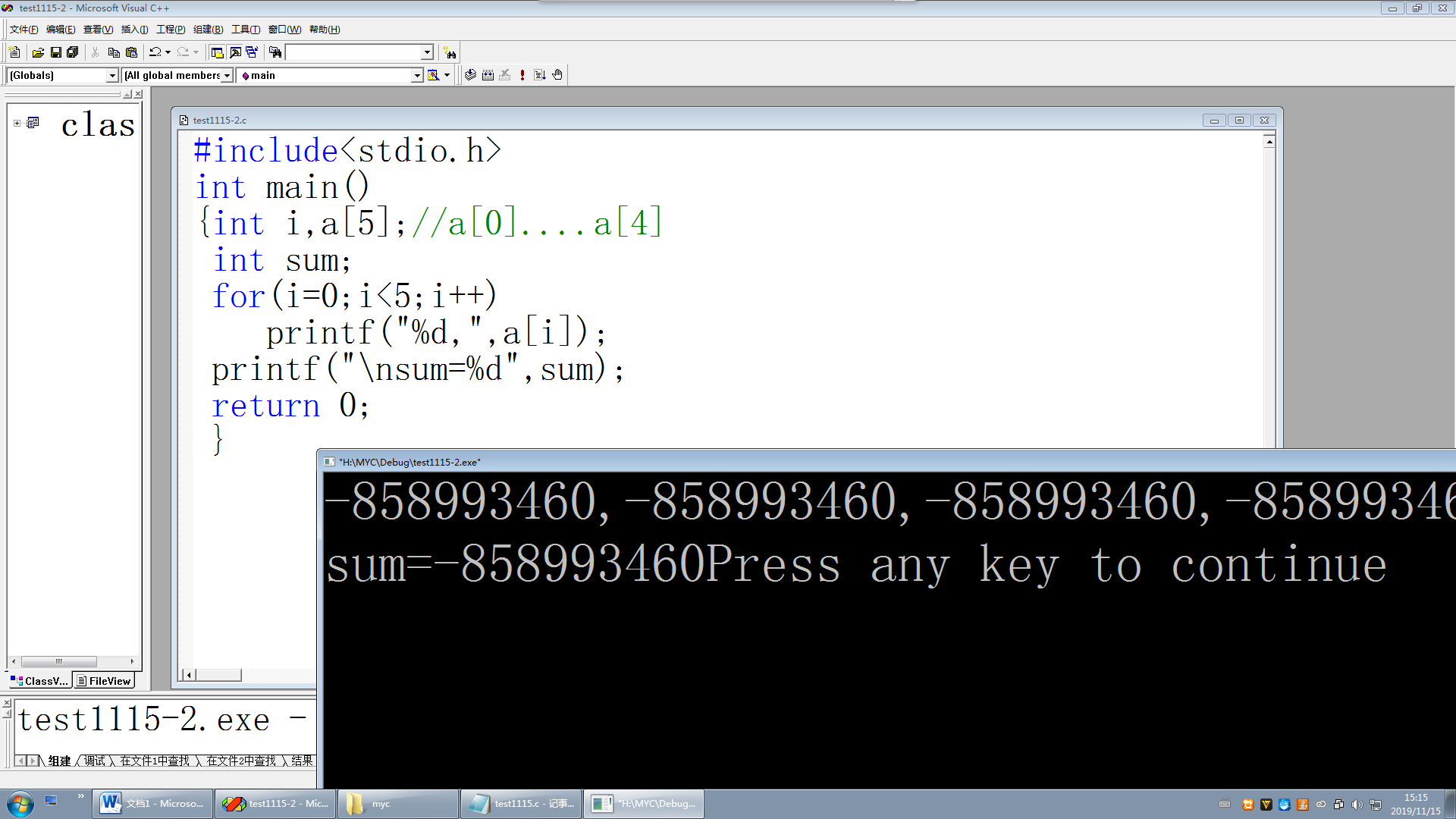
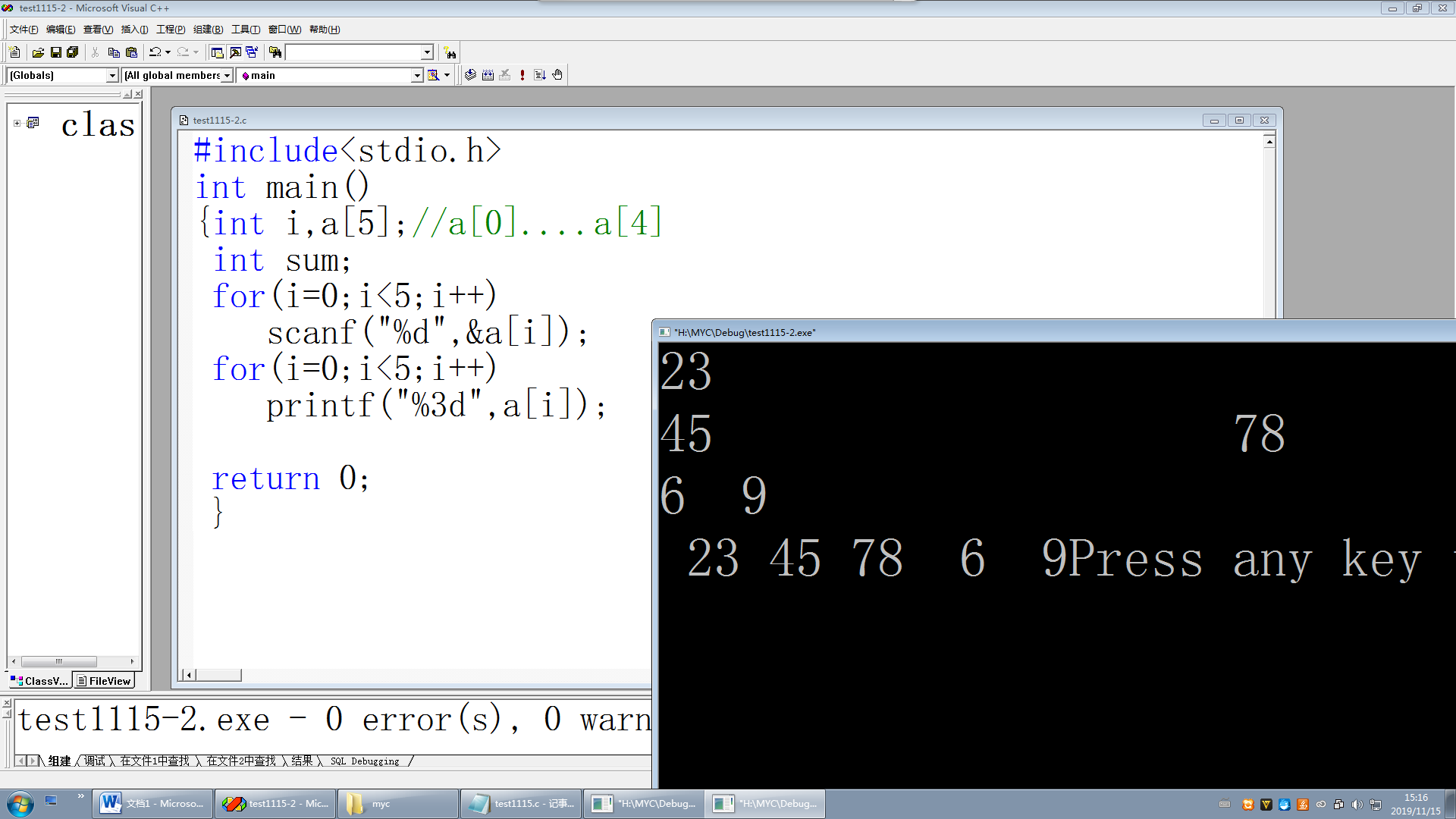
数组

1. 复习
2. 数组的定义

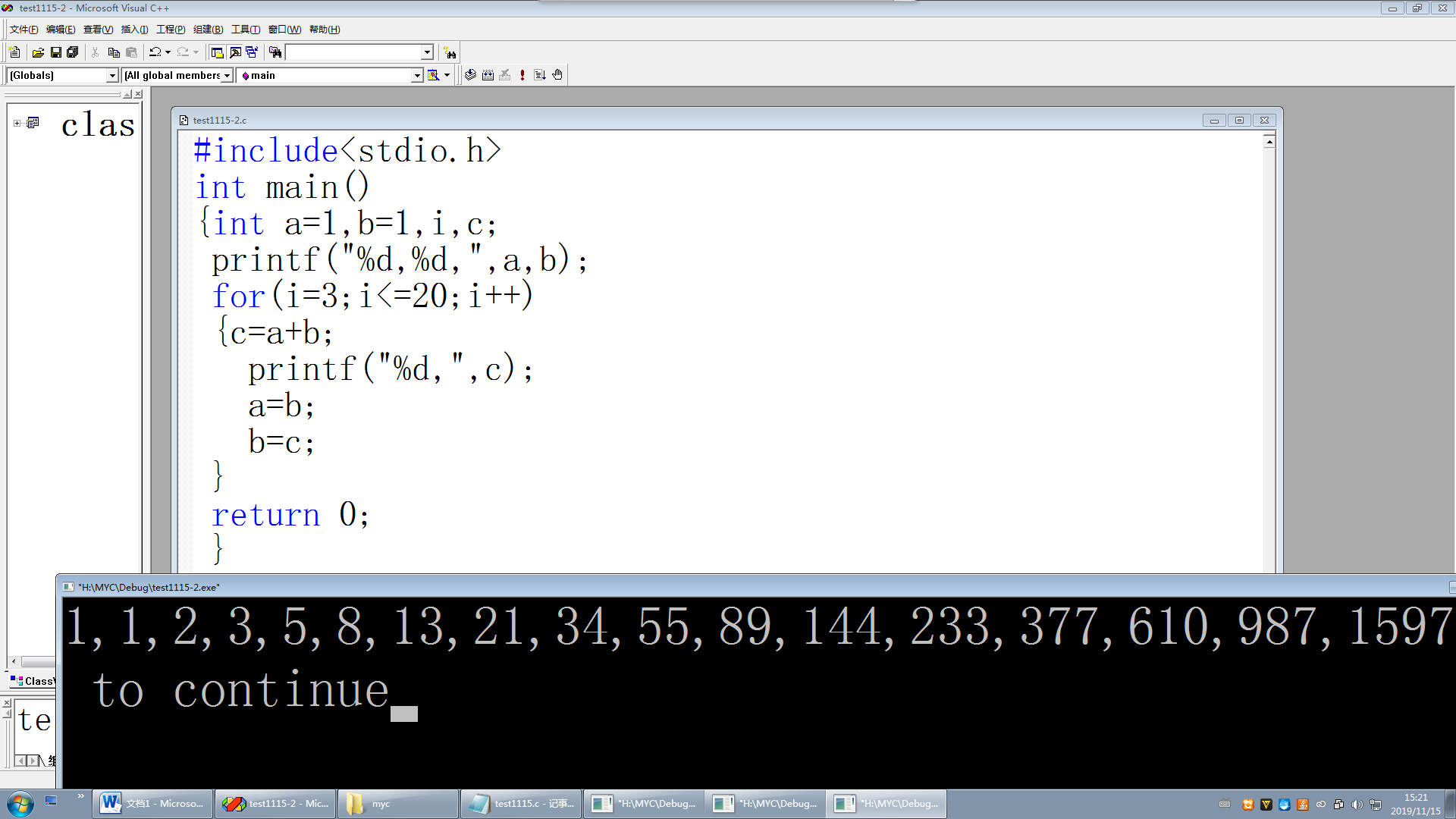


1. 算法应用
2. 输入输出数组元素



1. 计算斐波那契数列：1,1,2,3,5,8…输出前20项

法一：不用数组

 法二：用数组

#include<stdio.h>

int main()

{int f[20]={1,1},i;

for(i=2;i<20;i++)

f[i]=f[i-1]+f[i-2];

for(i=0;i<20;i++)

printf("%d,",f[i]);

return 0;

}

2、找出一行数中的最大者

#include<stdio.h>

int main()

{int a[12]={3,12,881,4,56,989,9,2,34},max,i;

max=a[0];

for(i=1;i<12;i++)

if (a[i]>max) max=a[i];//81

printf("max=%d",max);

return 0;

}

第二的改进，找出一行中最大值及下标。

#include<stdio.h>

int main()

{int a[12]={3,12,181,4,56,89,9,2,34},max,i,di;

max=a[0];di=0;

for(i=1;i<12;i++)

if (a[i]>max) {max=a[i];di=i;}//81

printf("最大值=%d，位于%d位置",max,di+1);

return 0;

}

1. 删除数组中的元素

算法：3,12,181,4,56,89,9,2,34（假设删除4）

3,12,181,4,56,89,9,2,34

3,12,181,56,56,9,2,34

3,12,181,56,9,9,2,34

3,12,181,56,9,2,2,34

3,12,181,56,9,2,34,34(输出时，末尾元素不输出，就可以了

补充算法，找元素，若找到，输出位置，没有找到，给出结论

#include<stdio.h>

int main()

{int a[12]={3,12,181,5,56,89,9,2,34,35,76,11},x,i,di;

scanf("%d",&x);//5

di=-1;

for(i=0;i<12;i++)

if (a[i]==x) di=i;//di=3;

if (di!=-1)

{printf("找到了,删除后如下:\n");

for(i=di;i<12-1;i++)

a[i]=a[i+1];

for(i=0;i<12-1;i++)

printf("%4d",a[i]);

}

else

printf("没有找到");

return 0;

}

1. 若已经有序，插入一个元素，仍然有序

分析算法：

1，3，**5**，7，9，11//插入数据4

1，3，5，7，9，11，11

1，3，5，7，9，9，11

1，3，5，7，7，9，11

1，3，5，5，7，9，11

1，3，**~~5~~**，5，7，9，11

基本实现：

#include<stdio.h>

int main()

{int a[13]={1,3,5,7,9,11,21,34,56,64,78,90},x,i,di;

scanf("%d",&x);//8

di=-1;

for(i=0;i<12;i++)

if (a[i]>=x) {di=i;break;}//di=3;

for(i=11;i>=di;i--)

a[i+1]=a[i];

a[di]=x;

for(i=0;i<13;i++)

printf("%4d",a[i]);

return 0;

}

上述算法无法插入最大值的数，改进如下：

#include<stdio.h>

int main()

{int a[13]={1,3,5,7,9,11,21,34,56,64,78,90},x,i,di;

scanf("%d",&x);//98

di=12;

for(i=0;i<12;i++)

if (a[i]>=x) {di=i;break;}//di=3;

for(i=11;i>=di;i--)

a[i+1]=a[i];

a[di]=x;

for(i=0;i<13;i++)

printf("%4d",a[i]);

return 0;

}

小练习：

定义五个元素的数组，用scanf依次输入，分别正序和逆序输出所有元素