

## Experiment 7 Function (1)

### 1. Experimental Purpose

- (1) Familiar with the method of defining functions.
- (2) The method of declaring functions in practice.
- (3) Familiarize yourself with the correspondence between the actual parameters involved in calling a function and the way in which values are passed.
- (4) Learn to compile and run multi-file programs.

### 2. Experimental Contents

(1) Write a function to discriminate prime numbers, enter an integer in the main function, and output information on whether a prime number is present or not.

- ① Input analysis
- ② Delete the function declaration of the main function, compile it, and analyze the result of compilation
- ③ After changing the position of the main function to another function, there is no function declaration in the main function.
- ④ Keep the function of discriminant prime numbers, modify the main function, and require the output of 100-200 prime numbers.

Solution

①Code:

```
#include <stdio.h>
#include <math.h>
int main()
{
    int n,m,i;
    int sushu(int);
    printf("input a digit:\n");
    while(getchar()!='\0')
    {
        scanf("%d",&n);
        m=sushu(n);
        if(m==0)
            printf("prime number\n");
        else
            printf("Not a prime number\n");
    }

    return 0;
}

int sushu(int n)
{
    int h,i,flag=0;
```

```

    h=sqrt(n);
    if(n==1||n==0)
        flag=1;
    else
    { for(i=2;i<=h;i++)
        if(n%i==0)
        {    flag=1;
            break;
        }
    }
    return (flag);
}

```



```

C:\Documents and Settings\AHUCC\桌面\Debug\7.1.exe
input a digit:
17
是素数
34
不是素数
2
是素数
1
不是素数
0
不是素数

```

②After deletion, the result is the same as that of ①.

③After deletion, the result is the same as that of ①

④Code:

```

#include <stdio.h>
#include <math.h>
int main()
{
    int n,m,i=1;
    int sushu(int);
    printf("The prime number between 100 and 200 is: \n");
    for(n=100;n<=200;n++)
    {
        m=sushu(n);
        {
            if(m==0)
            {
                printf("%8d",n);
                i++;
            }
        }
    }
}

```

```

    }
}
if(i%5==0)
{
    printf("\n");
    i=1;
}
}
printf("\n");
return 0;
}

```

```

int sushu(int n)
{
    int h,i,flag=0;
    h=sqrt(n);
    for(i=2;i<=h;i++)
        if(n%i==0)
        {
            flag=1;
            break;
        }
    return (flag);
}

```

```

C:\Documents and Settings\AHUCC\桌面\Debug\7.1.exe
100~200间的素数为:
101    103    107    109
113    127    131    137
139    149    151    157
163    167    173    179
181    191    193    197
199
Press any key to continue_

```

(2) Write a function that copies the vowel letters from one string to another and outputs them.

1. Input program, compile and run program, analysis result.

Analyzing the way parameters are written in the function declaration. There are two forms.

(a) Parameters in a function declaration are written in exactly the same way as defining a function, for example:

```
void cpy(char s[],char c[]);
```

(b) The parameters in a function declaration are written in the same manner as they were defined, but the array names are omitted, for example:

```
Void CPY (char[], char[]);
```

Compile and run, respectively, and analyze the results.

If the size of the array is arbitrarily specified, it is not possible, for example:

Void cpy(char s[40], char[40])

①Code:

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

```
int main()
```

```
{
```

```
    void cpy(char[],char[]);
```

```
    char s[80],c[80];
```

```
    printf("input string:\n");
```

```
    gets(s);
```

```
    cpy(s,c);
```

```
    printf("The vowel letters are:%s\n",c);
```

```
    return 0;
```

```
}
```

```
void cpy(char s[],char c[])
```

```
{
```

```
    int i,j;
```

```
    for (i=0,j=0;s[i]!='\0';i++)
```

```
        if(s[i]=='a'||s[i]=='A'||s[i]=='e'||s[i]=='E'||s[i]=='i'||s[i]=='I'||s[i]=='O'||s[i]=='o'||s[i]=='u'||s[i]=='
```

```
U')
```

```
    {
```

```
        c[j]=s[i];
```

```
        j++;
```

```
    }
```

```
    c[j]='\0';
```

```
}
```



②Programming results in the other two forms are still the same.

③No effect in the actual operation.

(3) Enter the results of 10 different learning courses and use functions to achieve the following functions:

(1) Calculate the average score of each student;

(2) Calculate the average score of each course;

(3) Find out the students and courses corresponding to the highest score of all 50 scores.

Code:

```
#include <stdio.h>
#define N 5
#define M 10
float a[M][N];
float avers[N];
float averl[M];
int i,j,ms,ml;
float max=0;
int main()
{
    void shuru();
    void suan_avers();
    void suan_averl();
    void suan_max();
    int n;
    shuru();
    suan_avers();
    suan_averl();
    suan_max();
    n=0;
    printf("\n No.      cour1    cour2    cour3    cour4    cour5    aver\n\n");
    for (i=0;i<M;i++)
    {
        printf("NO   %2d   ",++n);
        for (j=0;j<N;j++)
        {
            printf("%8.2f",a[i][j]);
        }
        printf("%8.2f\n",avers[i]);
    }
    printf("average:%8.2f%8.2f%8.2f%8.2f%8.2f",averl[0],averl[1],averl[2],averl[3],averl[4]);
    printf("\nhighest:%8.2f NO.   %d   course   %d\n",max,ms,ml);
    return 0;
}
```

```
void shuru()
{
    for (i=0;i<M;i++)
    {
        printf("input score of student %d:\n",i+1);
        for (j=0;j<N;j++)
            scanf("%f",&a[i][j]);
        printf("\n");
    }
}
```

```
    }  
}
```

```
void suan_avers()  
{  
    for (i=0;i<M;i++)  
        avers[i]=(a[i][0]+a[i][1]+a[i][2]+a[i][3]+a[i][4])/5;  
}
```

```
void suan_averl()  
{  
    float sum=0;  
    for (i=0;i<N;i++)  
        {for(j=0;j<M;j++)  
            sum=sum+a[j][i];  
        averl[i]=sum/10;  
        sum=0;  
    }  
}
```

```
void suan_max()  
{  
    for(i=0;i<10;i++)  
        for (j=0;j<5;j++)  
            if (a[i][j]>max)  
                {max=a[i][j];ms=i+1;ml=j+1;}  
}
```

```
"C:\Users\THCMAZ\\Desktop\6和7\s.7.3\Debug\7.3.exe"
input score of student 1:
87 88 89 90 91

input score of student 2:
88 89 90 91 92

input score of student 3:
89 90 91 92 93

input score of student 4:
90 91 92 93 94

input score of student 5:
91 92 93 94 95

input score of student 6:
92 93 94 95 96

input score of student 7:
93 94 95 96 97

input score of student 8:
94 95 96 97 98

input score of student 9:
95 96 97 98 99

input score of student 10:
96 97 98 99 100

No.      cour1  cour2  cour3  cour4  cour5  aver
NO  1      87.00  88.00  89.00  90.00  91.00  89.00
NO  2      88.00  89.00  90.00  91.00  92.00  90.00
NO  3      89.00  90.00  91.00  92.00  93.00  91.00
NO  4      90.00  91.00  92.00  93.00  94.00  92.00
NO  5      91.00  92.00  93.00  94.00  95.00  93.00
NO  6      92.00  93.00  94.00  95.00  96.00  94.00
NO  7      93.00  94.00  95.00  96.00  97.00  95.00
NO  8      94.00  95.00  96.00  97.00  98.00  96.00
NO  9      95.00  96.00  97.00  98.00  99.00  97.00
NO 10      96.00  97.00  98.00  99.00 100.00  98.00

average:  91.50  92.50  93.50  94.50  95.50
highest: 100.00 NO. 10 course 5
Press any key to continue
```

(4) Use a function to first output the longest word in a line string. This line string is passed from the main function to the function.

- ① Place two functions in the same program file, compile and run as one file.
- ② Place the two functions in two program files, compile, connect and run as two files.

① Code:

```
#include <stdio.h>
#include <string.h>
int main()
{
```

```

    int alphabetic(char);
    int longest(char[]);
    int i;
    char line[100];
    printf("input one line :\n");
    gets(line);
    printf("The longest word is:");
    for (i=longest(line);alphabetic(line[i]);i++)
        printf("%c",line[i]);
    printf("\n");
    return 0;
}

```

```

int alphabetic(char c)
{
    if((c>='a' && c<='z') || (c>'A' && c<='Z'))
        return(1);
    else
        return(0);
}

```

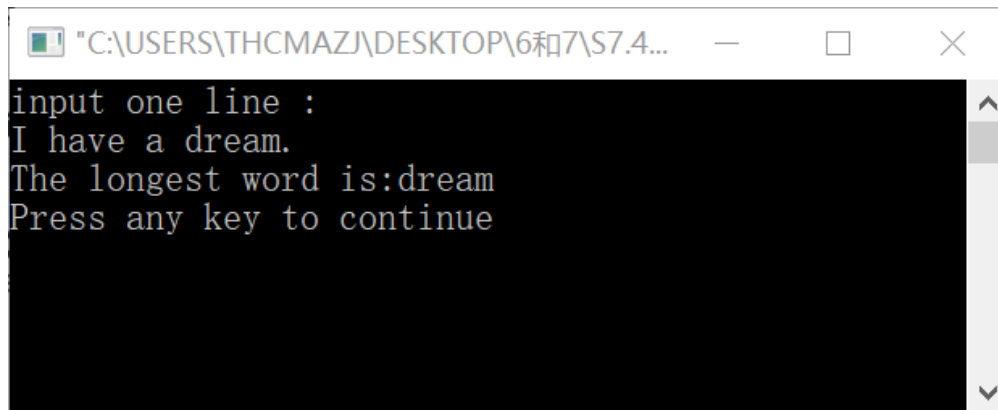
```

int longest(char string[])
{
    int len=0, i, length=0, flag=1, place=0, point;
    for(i=0; i<=strlen(string); i++)
        if(alphabetic(string[i]))
            if(flag)
            {
                point=i;
                flag=0;
            }
            else
                len++;
    else
    {
        flag=1;
        if(len>=length)
        {
            length=len;
            place=point;
            len=0;
        }
    }
    return(place);
}

```

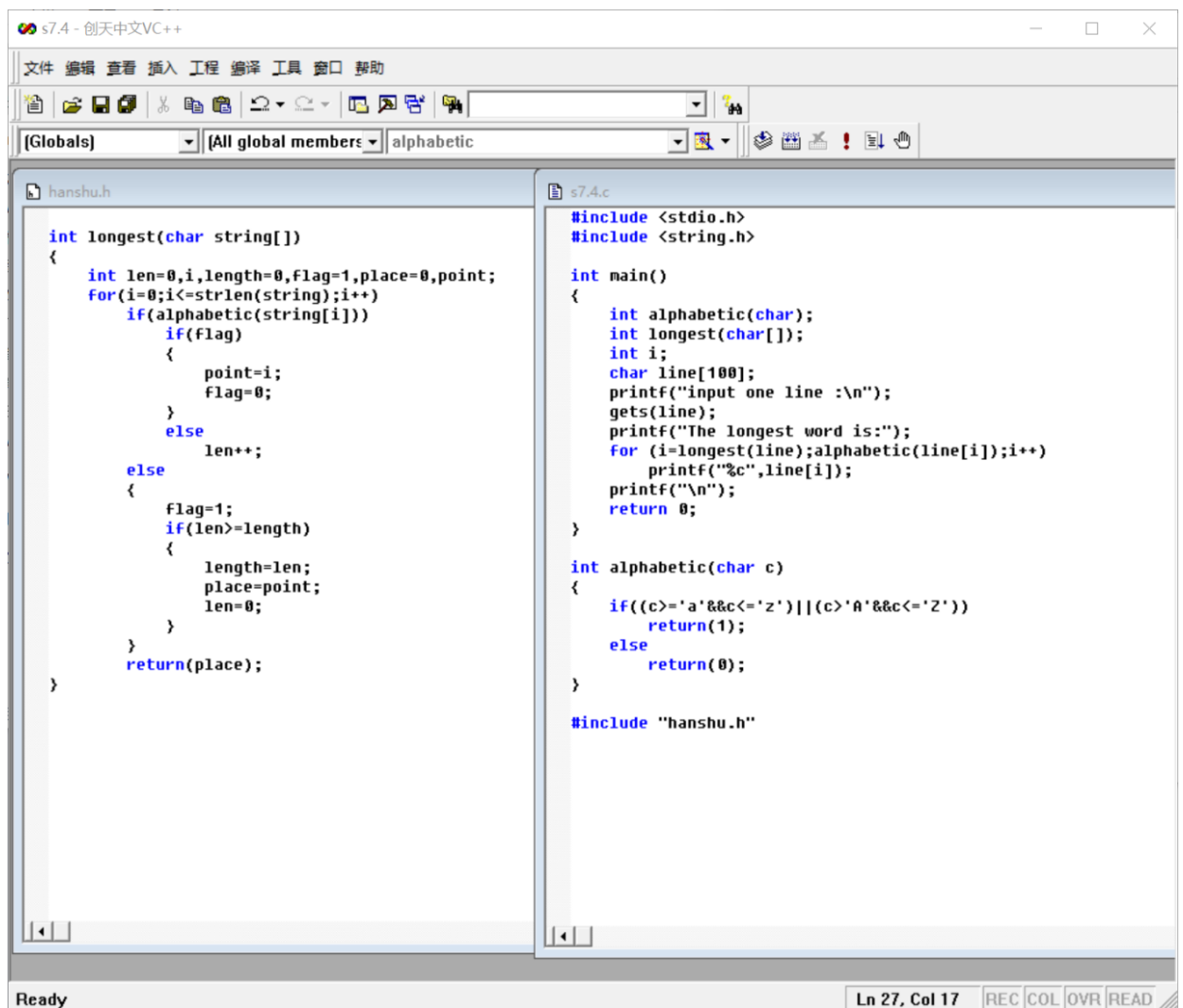


}



```
input one line :
I have a dream.
The longest word is:dream
Press any key to continue
```

②Code:



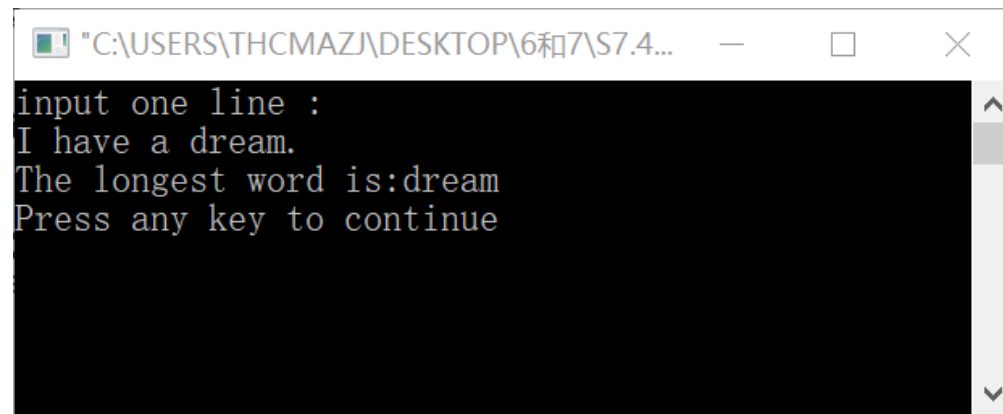
```
hanshu.h
int longest(char string[])
{
    int len=0,i,length=0,flag=1,place=0,point;
    for(i=0;i<=strlen(string);i++)
        if(alphabetic(string[i]))
            if(flag)
            {
                point=i;
                flag=0;
            }
            else
                len++;
    else
    {
        flag=1;
        if(len>length)
        {
            length=len;
            place=point;
            len=0;
        }
    }
    return(place);
}

s7.4.c
#include <stdio.h>
#include <string.h>

int main()
{
    int alphabetic(char);
    int longest(char[]);
    int i;
    char line[100];
    printf("input one line :\n");
    gets(line);
    printf("The longest word is:");
    for (i=longest(line);alphabetic(line[i]);i++)
        printf("%c",line[i]);
    printf("\n");
    return 0;
}

int alphabetic(char c)
{
    if((c>='a'&&c<='z')||(c>='A'&&c<='Z'))
        return(1);
    else
        return(0);
}

#include "hanshu.h"
```



A screenshot of a Windows command prompt window. The title bar at the top shows the file path "C:\USERS\THCMAZ\DESKTOP\6和7\S7.4..." followed by standard window controls (minimize, maximize, close). The command prompt area has a black background with white text. The text displayed is: "input one line :", "I have a dream.", "The longest word is:dream", and "Press any key to continue". A vertical scrollbar is visible on the right side of the text area.

```
input one line :  
I have a dream.  
The longest word is:dream  
Press any key to continue
```