

## Programming Design of Loop Structure in Experiment 5

### 1. Experimental Purpose

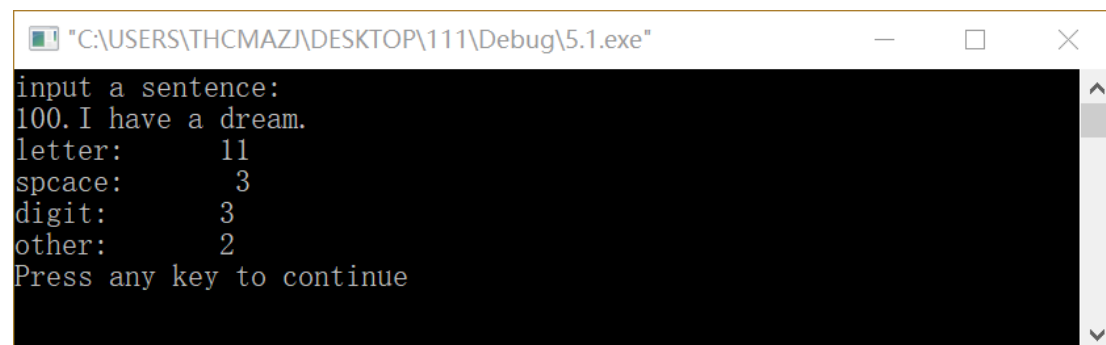
- (1) Familiarize yourself with how to implement loops with while, do...while, and for statements.
- (2) Master how to implement some common algorithms (such as exhaustion, iteration, recursion, etc.) by means of loops in program design.
- (3) Further study of the debugging program.

### 2. Experimental Contents

- (1) Enter a line of characters to count the number of English letters, spaces, numbers and other characters.

Code:

```
#include <stdio.h>
int main()
{
    int letter=0,space=0,digit=0,other=0;
    char c;
    printf("input a sentence:\n");
    while ((c=getchar())!='\n')
    {
        if ((c>='a'&& c<='z') || (c>='A'&& c<='Z'))
            letter++;
        else if(c==' ')
            space++;
        else if(c>='0'&&c<='9')
            digit++;
        else
            other++;
    }
    printf("letter:%8d\nspace:%8d\ndigit:%8d\nother:%8d\n",letter,space,digit,other);
    return 0;
}
```



```
"C:\USERS\THCMAZJ\DESKTOP\111\Debug\5.1.exe"
input a sentence:
100.I have a dream.
letter:      11
space:       3
digit:       3
other:       2
Press any key to continue
```

Modify the program so that it counts case.

Code:

```

#include <stdio.h>
int main()
{
    int upper=0,lower=0,space=0,digit=0,other=0;
    char c;
    printf("input a sentence:\n");
    while ((c=getchar())!='\n')
    {
        if (c>='A'&& c<='Z')
            upper++;
        else if (c>='a'&& c<='z')
            lower++;
        else if(c==' ')
            space++;
        else if(c>='0'&&c<='9')
            digit++;
        else
            other++;
    }
    printf("upper case:%8d\nlower case:%8d\nspace:%8d\ndigit:%8d\nother:%8d\n",upper,lower,space,digit,other);
    return 0;
}

```



```

"C:\USERS\THCMAZ\DESKTOP\111\Debug\5.1.exe"
input a sentence:
100.I have a dream.
upper case:      1
lower case:     10
space:           3
digit:           3
other:           2
Press any key to continue

```

(2)Output of all "daffodils" The so-called "number of daffodils" refers to a three-digit number whose digit cube is equal to the number itself, for example, 153 is a number of daffodils because  $153 = 1^3 + 5^3 + 3^3$ .

Code:

```

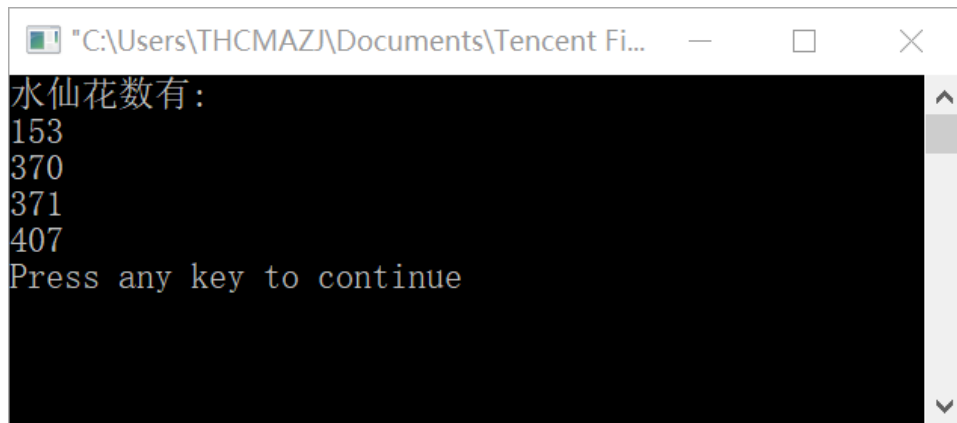
#include <stdio.h>
int main()
{
    int n,a,b,c;
    printf("The number of daffodils is:");
    for (n=100;n<1000;n++)

```

```

{
    a=n/100;
    b=(n-a*100)/10;
    c=n-a*100-b*10;
    if(n==a*a*a+b*b*b+c*c*c)
        printf("\n%d",n);
}
printf("\n");
return 0;
}

```



```

"C:\Users\THCMAZJ\Documents\Tencent Fi...
水仙花数有:
153
370
371
407
Press any key to continue

```

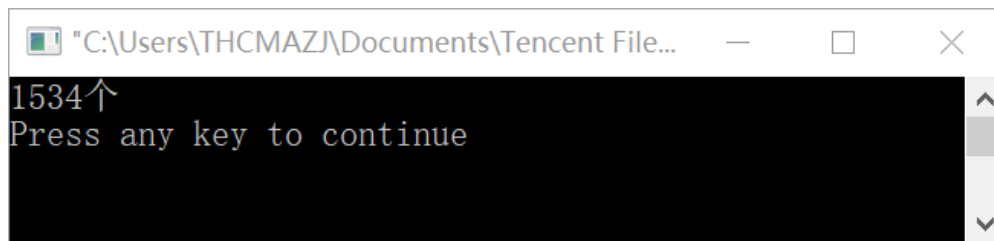
(3)Monkey eating peach problem. The monkey picked several peaches on the first day and ate half of them immediately. It was not addicted to them, but ate one more. The next morning he ate half of the remaining peaches and one more. Every morning afterwards, I ate half and one half of the rest of the previous day. When I wanted to eat again the morning of the 10th day, I saw there was only one peach left. See how many peaches were picked on the first day.

Code:

```

#include <stdio.h>
int main()
{
    int n,i;
    n=1;
    for(i=1;i<=9;i++)
        n=(n+1)*2;
    printf("%d 个\n",n);
    return 0;
}

```

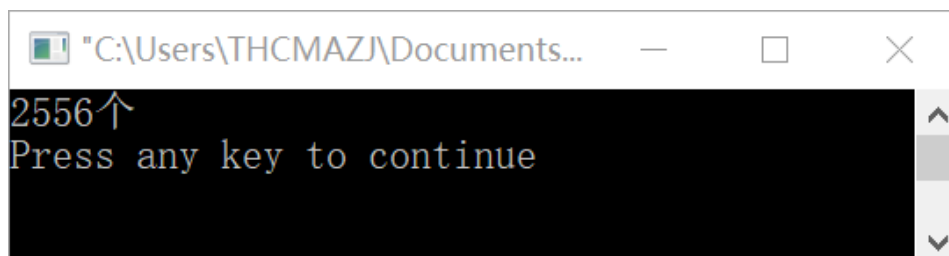


```
"C:\Users\THCMAZ\Documents\Tencent File..."
1534个
Press any key to continue
```

After getting the results, change the subject to Monkey eating half the rest of the previous day each day, then eat two more. Please modify the program and run it to check if the results are correct.

Code:

```
#include <stdio.h>
int main()
{
    int n,i;
    n=1;
    for(i=1;i<=9;i++)
        n=(n+2)*2;
    printf("%d 个\n",n);
    return 0;
}
```



```
"C:\Users\THCMAZ\Documents..."
2556个
Press any key to continue
```

(4) The Newton iteration method is used to find the root of the equation  $2x^3=4x^2+3x-6=0$  near 1.5.

After you get the correct results, modify the program to change the initial x value from 1.5 to 100, 1000, 10000, run again, observe the results, and analyze whether different initial x values have an impact on the results.

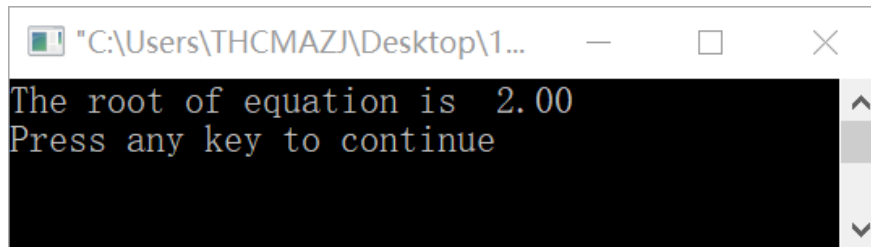
Code:

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x1,x0,f,f1;
    x1=10000;
    do
    {
```

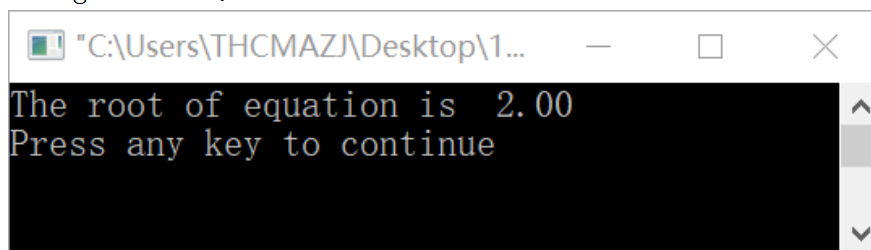
```

    x0=x1;
    f=((2*x0-4)*x0+3)*x0-6;
    f1=(6*x0-8)*x0+3;
    x1=x0-f/f1;
}
while (fabs(x1-x0)>=1e-5);
printf("The root of equation is %5.2f\n",x1);
return 0;
}

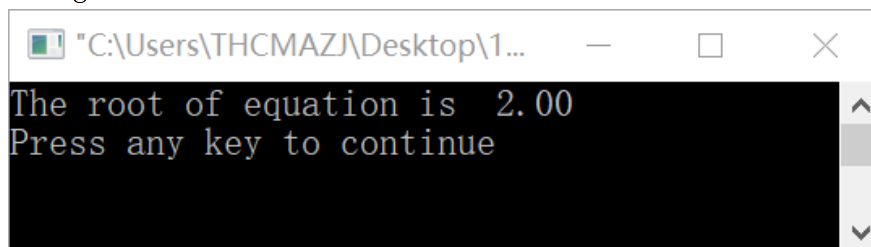
```



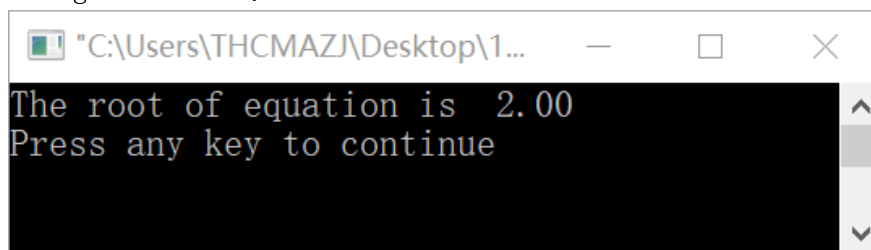
Change to 100:



Change to 1000:



Change to 10000:



Modify the program so that it outputs the number of iterations and the results of each iteration, and analyze whether different initial x values have an effect on the number of iterations.

Code:

```

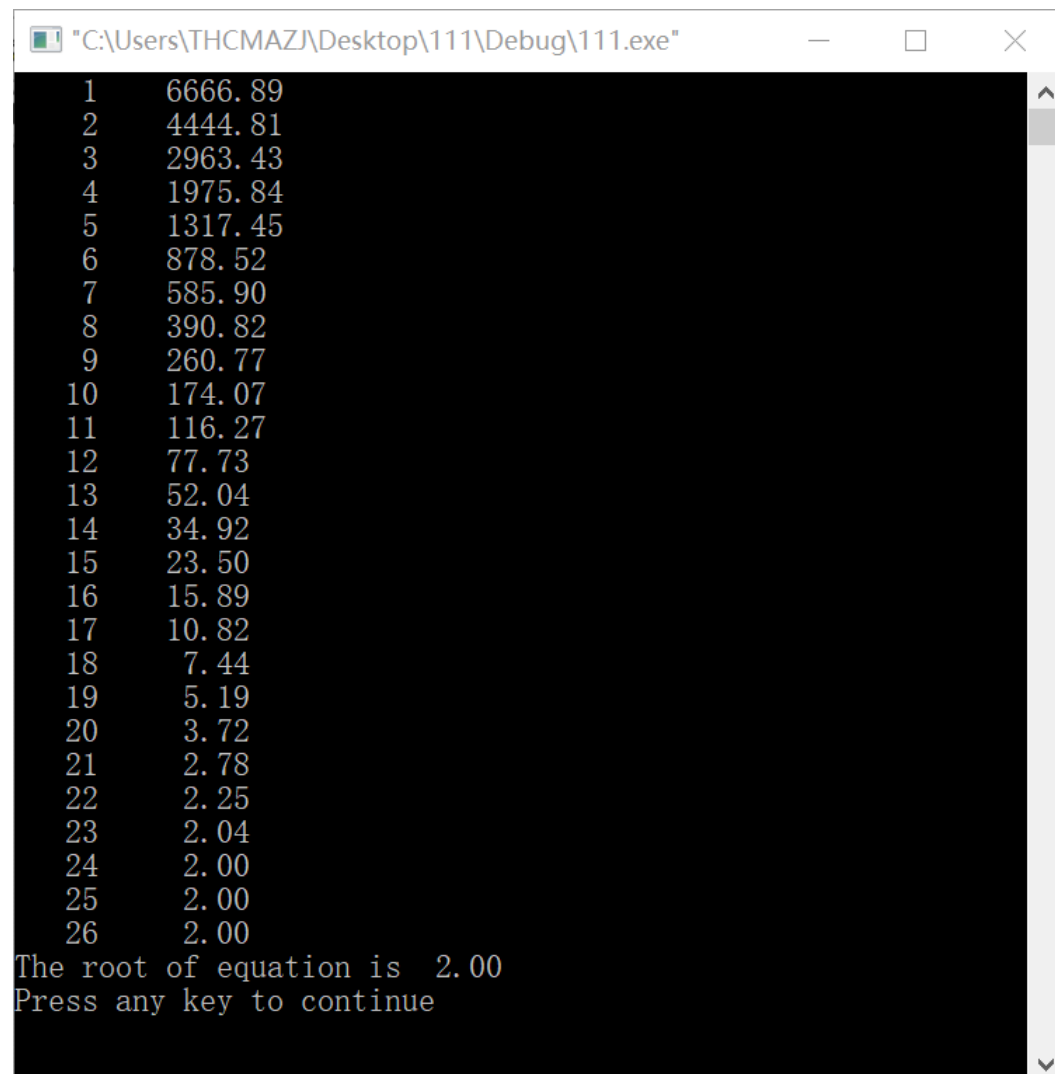
#include <stdio.h>
#include <math.h>
int main()

```

```

{
    double x1,x0,f,f1,n=0;
    x1=10000;
    do
    {
        x0=x1;
        f=((2*x0-4)*x0+3)*x0-6;
        f1=(6*x0-8)*x0+3;
        x1=x0-f/f1;
        n++;
        printf("%5.f      %5.2f\n",n,x1);
    }
    while (fabs(x1-x0)>=1e-5);
    printf("The root of equation is %5.2f\n",x1);
    return 0;
}

```



```

C:\Users\THCMAZJ\Desktop\111\Debug\111.exe
1      6666.89
2      4444.81
3      2963.43
4      1975.84
5      1317.45
6      878.52
7      585.90
8      390.82
9      260.77
10     174.07
11     116.27
12     77.73
13     52.04
14     34.92
15     23.50
16     15.89
17     10.82
18     7.44
19     5.19
20     3.72
21     2.78
22     2.25
23     2.04
24     2.00
25     2.00
26     2.00
The root of equation is 2.00
Press any key to continue

```