

## 第七次-17377191-段秋阳

---

### 1. P229, 6 (主函数完成输入输出即可，注意公式的化简);

```
//P229 6
#include <stdio.h>
#include <stdlib.h>

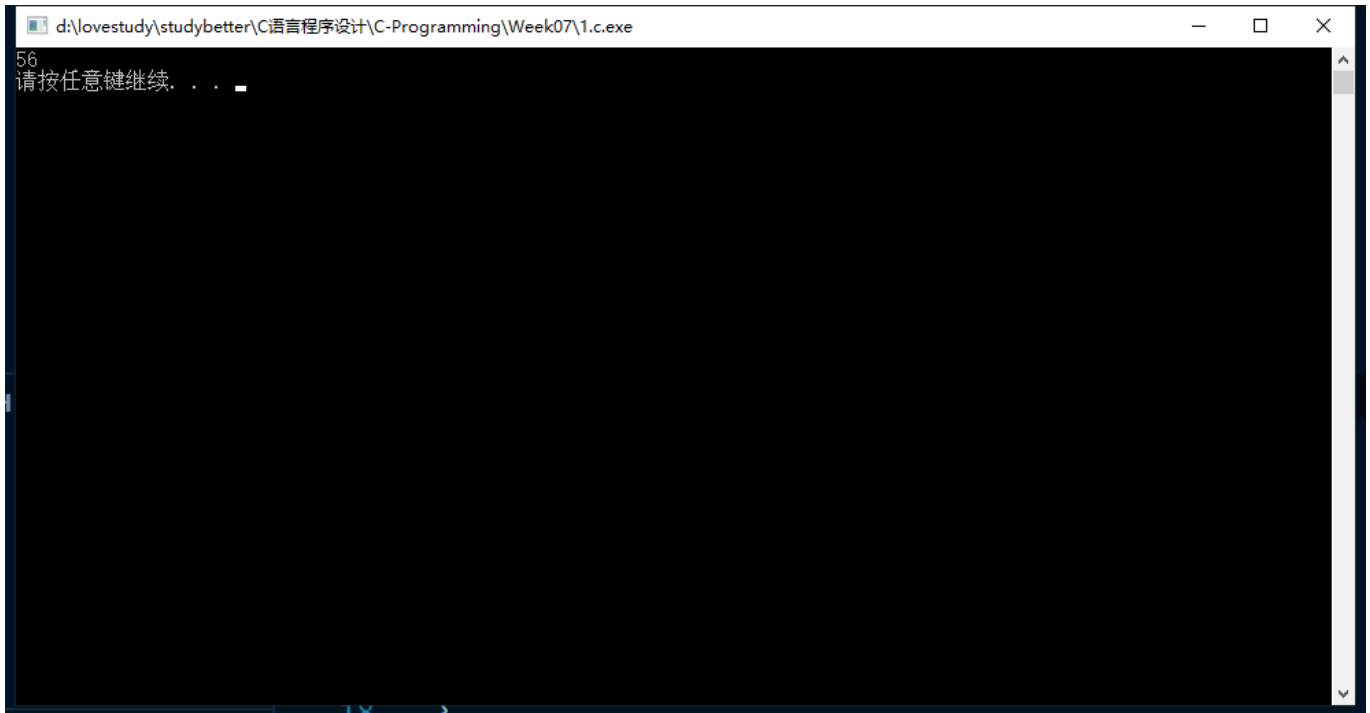
int factorial(int num);
int combination(int m, int n);

int main()
{
    int m = 8, n = 3;
    printf("%d\n", combination(m, n));

    system("pause");
    return 0;
}

int factorial(int num)
{
    if (num < 0)
    {
        printf("Input error!\n");
        return -1;
    }
    else if (num == 0)
        return 1;
    else
        return num * factorial(num - 1);
}

int combination(int m, int n)
{
    if (m >= 0 && n >= 0)
        if (m < n)
        {
            printf("Input error!\n");
            return -1;
        }
        else
            return factorial(m) / (factorial(n) * factorial(m - n));
    else
    {
        printf("Input error!\n");
        return -1;
    }
}
```



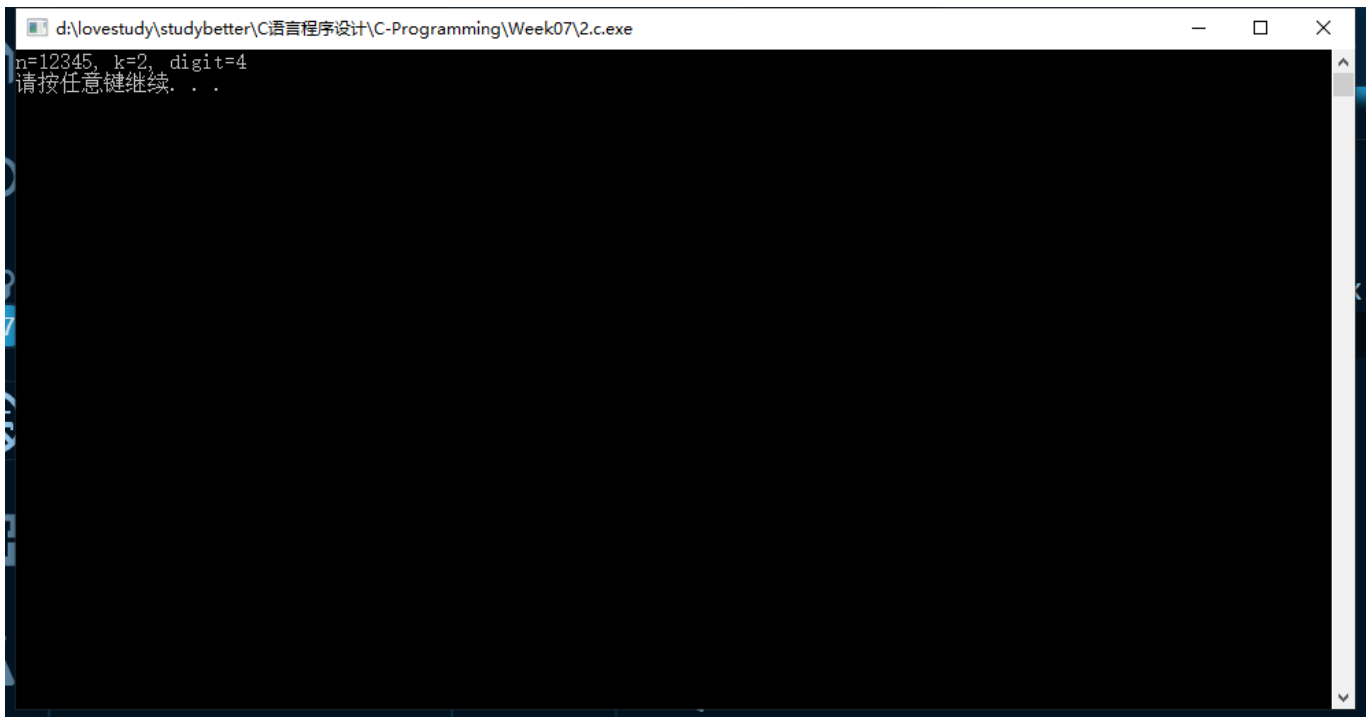
2. 实现一个函数num\_digits(n,k)，使得函数返回正整数n中的第k位数字（从右边算起）（如果k大于n所含有的数字的个数，则返回-1）并在主函数中进行测试；

```
//num_digits(n,k)
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXDIGITS 100

int num_digits(int num, int k);

int main()
{
    int n = 12345, k = 2;
    printf("n=%d, k=%d, digit=%c\n", n, k, num_digits(n, k));
    system("pause");
    return 0;
}

int num_digits(int n, int k)
{
    char arr[MAXDIGITS];
    if (k <= n)
    {
        itoa(n, arr, 10);
        int len = strlen(arr);
        return arr[len - k];
    }
    else
        return -1;
}
```



3. 用C实现一个能够计算高精度的Pi值的函数并测试（找一个自己理解并喜欢的公式来计算）；

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define epsilon 1e-6 //absloute error

int main()
{
    int i = 0;
    double sum = 0, pi = 0;
    double u;
    time_t start, end;

    //start = time(NULL);

    do
    {
        u = pow(-1.0, i) / (2 * i + 1);
        sum += u;
        printf("u=%lf, sum=%lf\n", u, 4.0 * sum);
        i++;
    } while (fabs(u) > epsilon);
    pi = 4.0 * sum;

    //end = time(NULL);

    printf("pi = %.10f\n", pi);
    printf("time = %d\n", difftime(end, start));
}
```

```

    system("pause");
    return 0;
}

```

```

d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\3.c.exe
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
u=-0.000001, sum=3.141591
u=0.000001, sum=3.141595
pi = 3.1415946536
time = 0
请按任意键继续. . .

```

#### 4. P248, 6;

```

//P248 6
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void HILO();
void Retry();

int onGame = 1;

int main()
{
    do
    {
        HILO();
    } while (onGame);

    system("pause");
    return 0;
}

void HILO()
{
    int a, guess, flag = 0;

```

```
srand(time(NULL));
a = rand() % 100 + 1; //Initialize a random integer between 1 and 100
printf("a = %d\n", a);

for (int i = 7; i >= 1; i--)
{
    printf("You have %d chances\n", i);
    scanf("%d", &guess);
    if (guess == a)
    {
        flag = 1;
        break;
    }
    if (guess > a)
    {
        printf("Wrong Number, Try Again, %d Chances Left :)\n", i - 1);
        printf("Note:Greater\n");
    }
    if (guess < a)
    {
        printf("Wrong Number, Try Again, %d Chances Left :)\n", i - 1);
        printf("Note:Smaller\n");
    }
}
if (flag)
    printf("Hooray, you have won!\n");
else
    printf("Sorry, you lose. The correct answer is %d\n", a);

Retry();
}

void Retry()
{
    char retry;

    printf("Would you like to play again (y/n)?\n");
    scanf("%c", &retry);

    if (retry == 'y')
        HILO();
    else if (retry == 'n')
    {
        printf("Exited.\n");
        onGame = 0;
    }
    else
    {
        printf("Read what I say! :(\n");
        Retry();
    }
}
```

```

d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\4.c.exe
a = 15
You have 7 chances
60
Wrong Number, Try Again, 6 Chances Left :)
Note:Greater
You have 6 chances
30
Wrong Number, Try Again, 5 Chances Left :)
Note:Greater
You have 5 chances
20
Wrong Number, Try Again, 4 Chances Left :)
Note:Greater
You have 4 chances
15
Hooray, you have won!
Would you like to play again (y/n)?
Read what I say! :(
Would you like to play again (y/n)?
n
Exited.
请按任意键继续. . .

```

5. P249, 12（了解一下当前的文本自动生成技术，以及在自动问答等方面的应用）；

```

//P249 12
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <time.h>

int inArray(char, char);

char kw[6][2] = {{ 'a', 't' }, { 'i', 's' }, { 'h', 'e' }, { 'w', 'e' }, { 'u', 'p' }, { 'o', 'n' }};

int main()
{
    int i = 0, r = 0, temp = 0;
    srand(time(NULL));

    do
    {
        temp = r;
        r = rand() % 26 + 1;
        printf("%c", r + 96);
        i++;
        //Sleep(200);
    } while (inArray((char)(temp + 96), (char)(r + 96)) == 0);

    printf("\n");
    printf("total:%d\n", i);
}

```

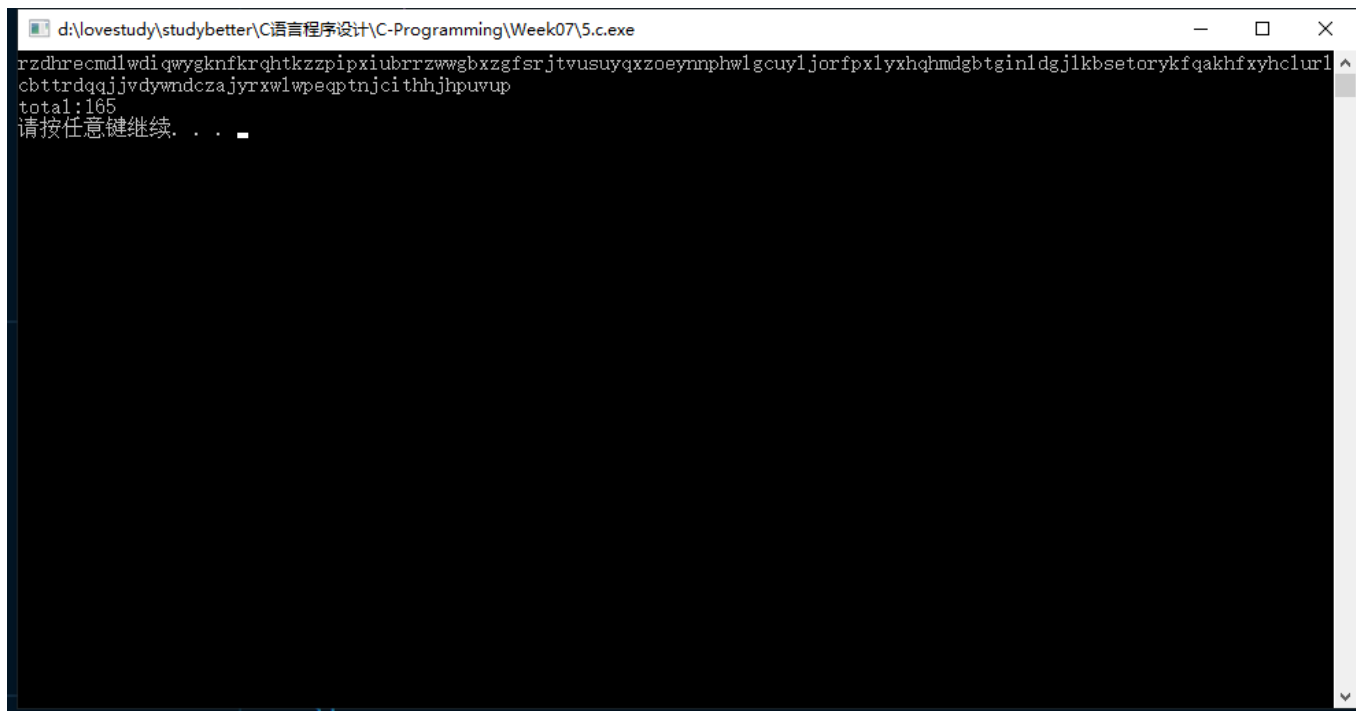
```

    system("pause");
    return 0;
}

int inArray(char c1, char c2)
{
    for (int j = 0; j < 6; j++)
    {
        if (c1 == kw[j][0] && c2 == kw[j][1])
            return 1;
        else
            continue;
    }

    return 0;
}

```



```

d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\5.c.exe
rzdhrccmdlwdi qwygknfkrqhtkzzpipxiubrzzwgbxzfgrjtvusuyqxxoeynnphwlgcuyljorfxlyxhqhmdgbtginldgjlkbsetrykfqakhfxyhclur1
cbttrdqqjjvdywndcza jyrxwlpqptnjci thhjhpuvup
total:165
请按任意键继续. . .

```

## 6. P249,13（注意边界条件的设置）；

```

//P249 13
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define NightNum 500.0

int main()
{
    int count = 0;
    srand(time(NULL));

    for (int i = 1; i <= NightNum; i++)

```

```

{
    int step = 0, r = 0;
    do
    {
        r = rand() % 100 + 1;

        if (r < 50)
            step += 2;
        else
            step -= 1;

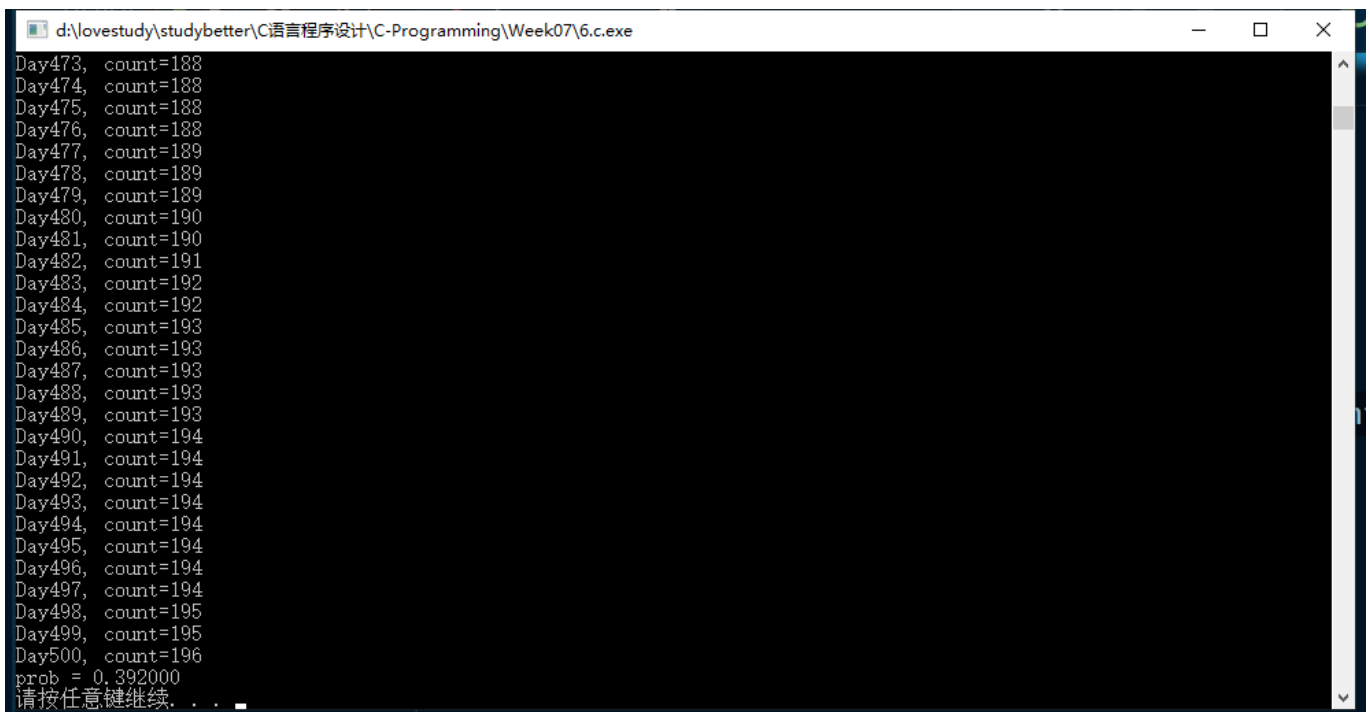
        if (step == 0)
            break;
    } while (step < 10);

    if (step)
        count += 1;

    printf("Day%d, count=%d\n", i, count);
}

printf("prob = %lf\n", count / NightNum);
system("pause");
return 0;
}

```



```

d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\6.c.exe
Day473, count=188
Day474, count=188
Day475, count=188
Day476, count=188
Day477, count=189
Day478, count=189
Day479, count=189
Day480, count=190
Day481, count=190
Day482, count=191
Day483, count=192
Day484, count=192
Day485, count=193
Day486, count=193
Day487, count=193
Day488, count=193
Day489, count=193
Day490, count=194
Day491, count=194
Day492, count=194
Day493, count=194
Day494, count=194
Day495, count=194
Day496, count=194
Day497, count=194
Day498, count=195
Day499, count=195
Day500, count=196
prob = 0.392000
请按任意键继续. . .

```

7. 用C实现一个能够产生高斯分布随机数的函数并测试;

```

//Gaussian Distribution
#include <stdio.h>
#include <stdlib.h>

```



```
#include <math.h>
#include <time.h>
#define MAXNUM 100
double genGaussianRand();

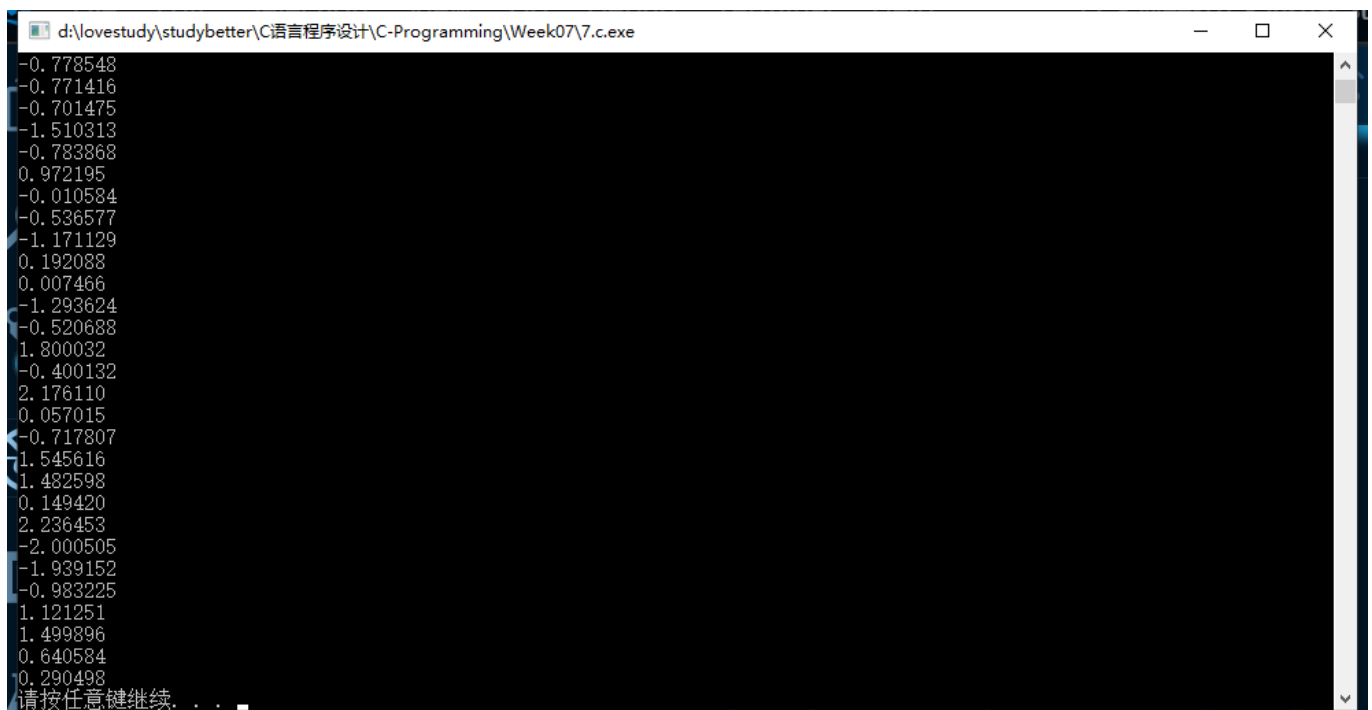
int main()
{
    srand(time(NULL));
    for (int i = 0; i < MAXNUM; i++)
    {
        printf("%lf\n", genGaussianRand());
    }
    system("pause");
    return 0;
}

double genGaussianRand()
{
    static double U1, U2;
    static int phase = 0;
    double Z;

    U1 = rand() * (1.0 / RAND_MAX);
    U2 = rand() * (1.0 / RAND_MAX);

    if (phase == 0)
        Z = sqrt(-2.0 * log(U1)) * sin(2.0 * M_PI * U2);
    else
        Z = sqrt(-2.0 * log(U1)) * cos(2.0 * M_PI * U2);

    phase = 1 - phase;
    return Z;
}
```



The screenshot shows a Windows command prompt window titled "d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\7.c.exe". The window displays the output of the C program, which consists of 100 lines of Gaussian random numbers. The numbers are printed in a single column, with some values like -0.778548, -0.771416, -0.701475, -1.510313, -0.783868, 0.972195, -0.010584, -0.536577, -1.171129, 0.192088, 0.007466, -1.293624, -0.520688, 1.800032, -0.400132, 2.176110, 0.057015, -0.717807, 1.545616, 1.482598, 0.149420, 2.236453, -2.000505, -1.939152, -0.983225, 1.121251, 1.499896, 0.640584, and 0.290498. The window ends with the prompt "请按任意键继续. . .".

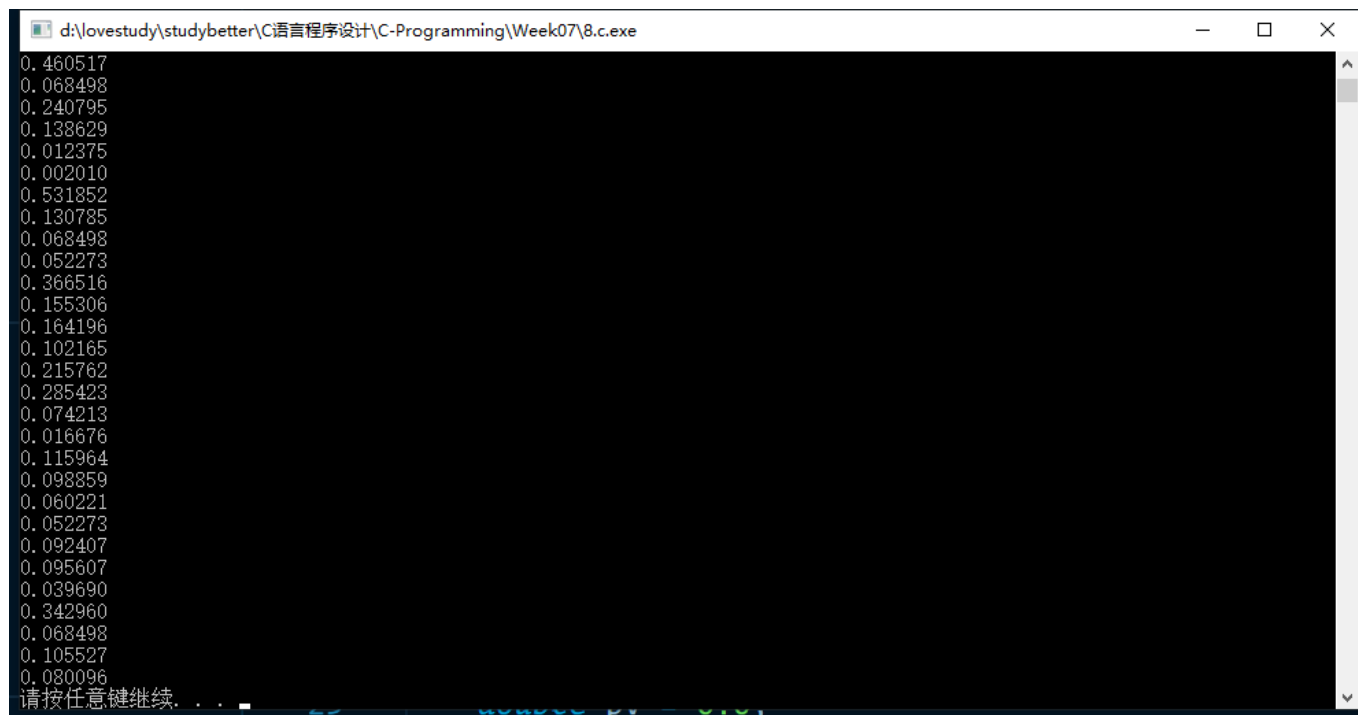
## 8. 用C实现一个能够产生指数分布随机数的函数并测试。

```
//Index Distribution
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#define MAXNUM 100
#define LAMBDA 5.0

double randomExponential(double);

int main()
{
    for (int i = 0; i < MAXNUM; i++)
    {
        printf("%lf\n", randomExponential(LAMBDA));
    }
    system("pause");
    return 0;
}

double randomExponential(double lambda)
{
    double pv = 0.0;
    do
    {
        pv = (double)(rand() % 100) / 100;
    } while (pv == 0);
    pv = (-1 / lambda) * log(1 - pv);
    return pv;
}
```



```
d:\lovestudy\studybetter\C语言程序设计\C-Programming\Week07\8.c.exe
0.460517
0.068498
0.240795
0.138629
0.012375
0.002010
0.531852
0.130785
0.068498
0.052273
0.366516
0.155306
0.164196
0.102165
0.215762
0.285423
0.074213
0.016676
0.115964
0.098859
0.060221
0.052273
0.092407
0.095607
0.039690
0.342960
0.068498
0.105527
0.080096
请按任意键继续. . .
```