## 题目：冒泡排序

## 源代码：

#include <stdio.h>

int main()

{

int rank[100], number, i, process, time;

//输入需要排序的数组：

printf("\n请输入有多少个数需要排序： ");

scanf("%d", &number);

printf("输入需要排序的数字（中间用空格分隔）：");

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

{

scanf("%d", &rank[i]);

}

//排序数组并输出过程：

for (time = 1; time <number; time++)

{

for (i = 0; i < number - time; i++)

{

if (rank[i] > rank[i + 1])

{

process = rank[i];

rank[i] = rank[i + 1];

rank[i + 1] = process;

}

}

printf("\n第%d遍的排序结果：", time);

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

{

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

}

}

//输出最终结果：

printf("\n\n最终的排序结果：");

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

{

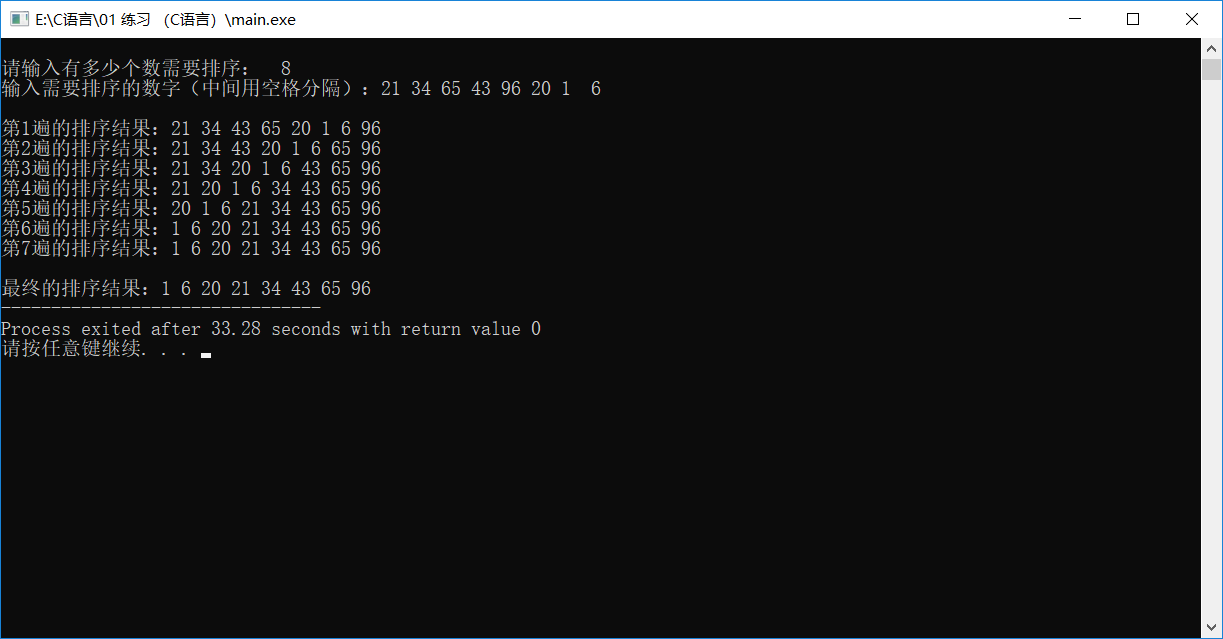
printf("%d ", rank[i]);

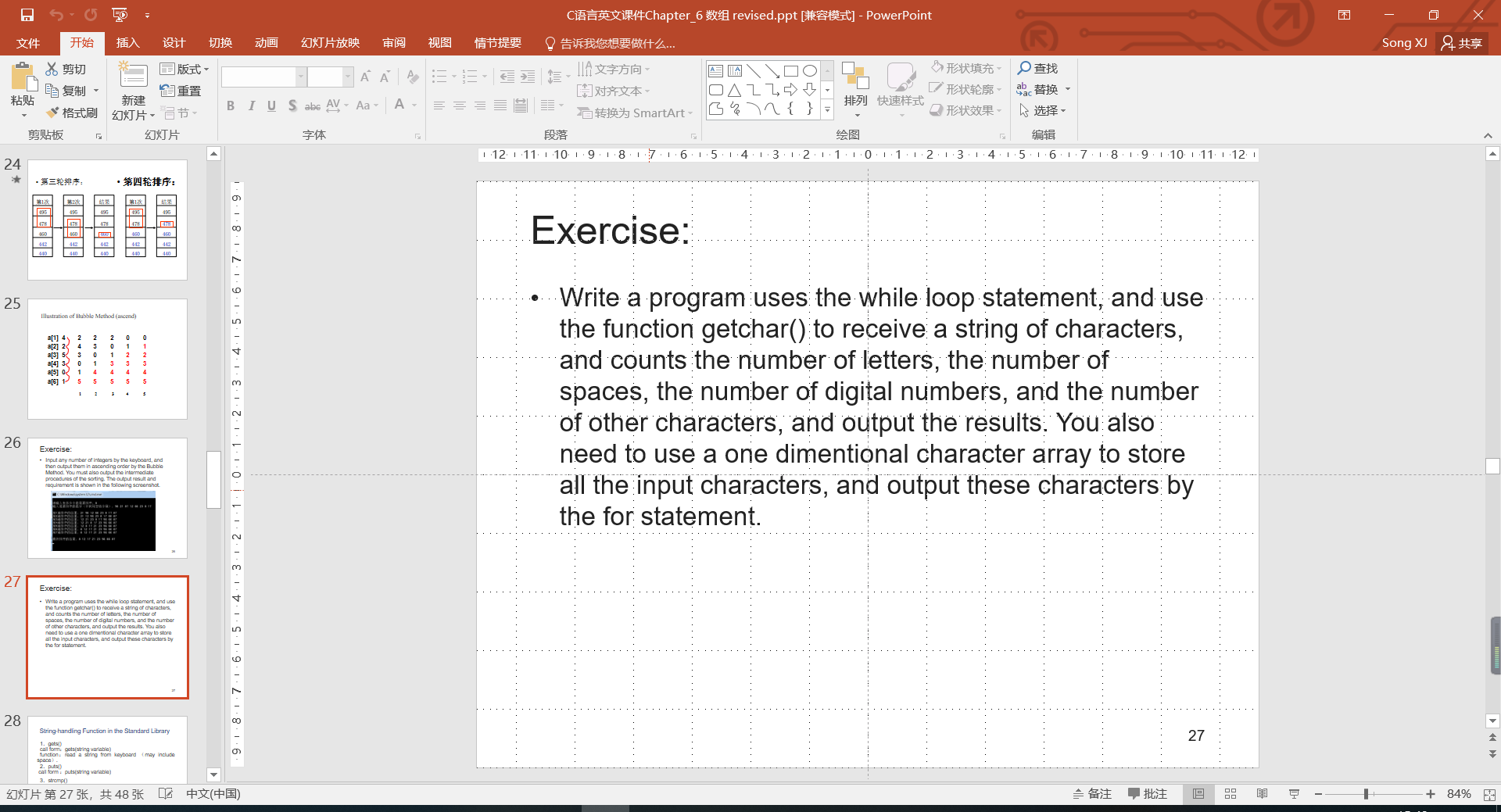
}

return 0;

}

## 输出结果：





## 源代码：

#include<stdio.h>

#include<string.h>

int main()

{

//定义

char c;

char str[100];

int num, i = 0, letter = 0, space = 0, digital = 0, other = 0;

//输入并计数

c = getchar();

str[0] = c;

while (c != 10)

{

i++;

if (c >= 48 && c <= 57) ++digital;

else if ((c >= 65 && c <= 90) || (c >= 97 && c <= 122)) ++letter;

else if (c == 32) ++space;

else ++other;

c = getchar();

str[i] = c;

}

//输出数组：

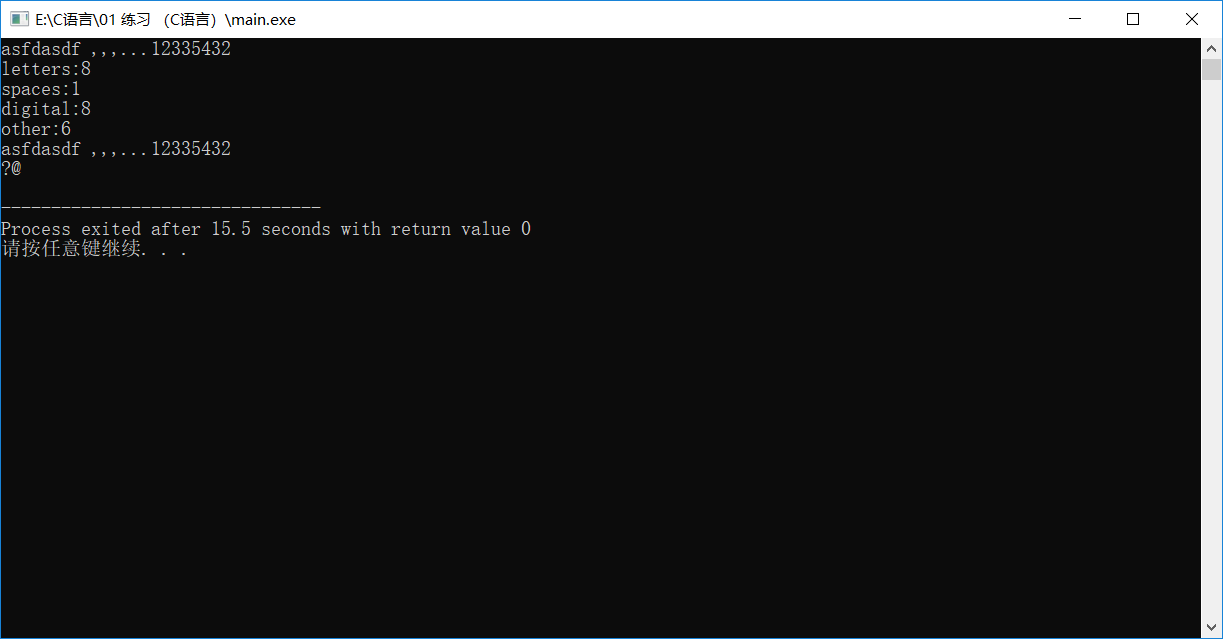
printf("letters:%d\nspaces:%d\ndigital:%d\nother:%d\n", letter, space, digital, other);

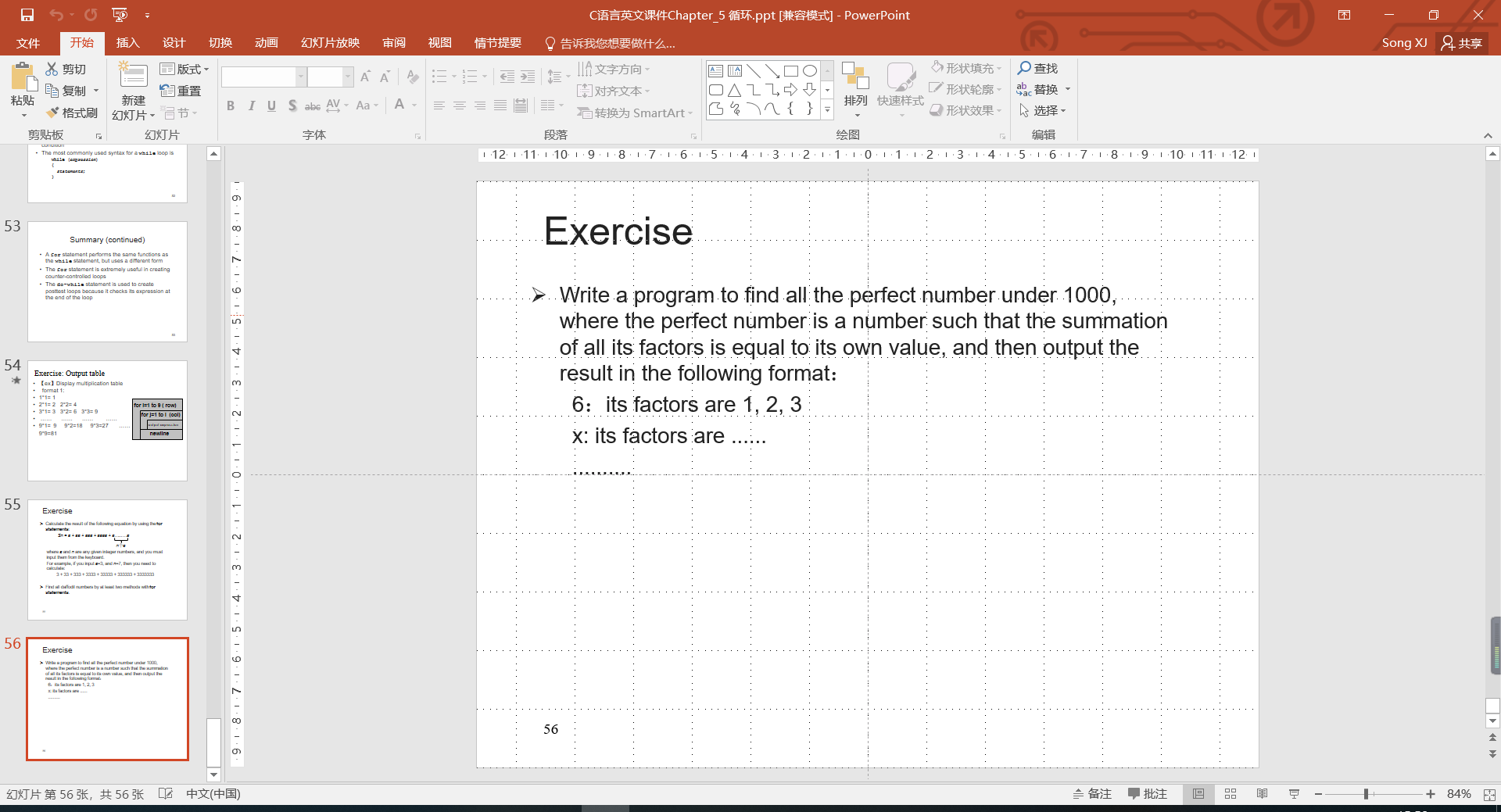
puts(str);

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

int main()

{

int factor, number, sum = 0;

for (number = 1; number <= 1000; number++)

{

//找因数并求和

sum = 0;

for (factor = 1; factor <= number / 2; factor++)

{

if (number%factor == 0)

sum += factor;

}

//输出因数及结果

if (number == sum)

{

printf("\n%d:It's factors are: ", number);

for (factor = 1; factor <= number / 2; factor++)

{

if (number%factor == 0)

printf("%d,", factor);

}

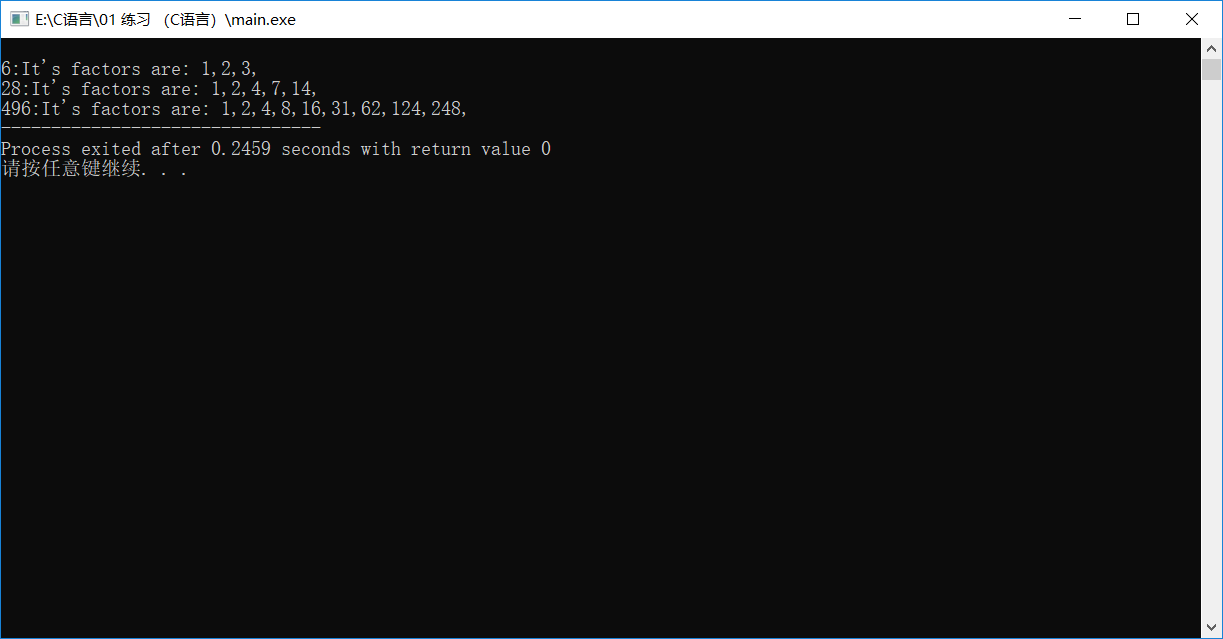
}

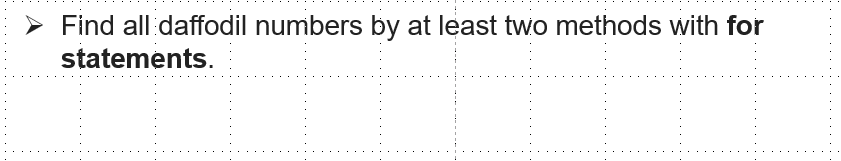
}

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

int main()

{

int Daffodil, Ge, Shi, Bai;

printf("Daffodil Number:\n");

for (Daffodil = 100; Daffodil < 1000; Daffodil++)

{

Bai = Daffodil / 100;

Shi = (Daffodil / 10) % 10;

Ge = Daffodil % 10;

Bai = Bai \* Bai \* Bai;

Shi = Shi \* Shi \* Shi;

Ge = Ge \* Ge \* Ge;

if (Daffodil == (Bai + Shi + Ge))

{

printf("%-7.d", Daffodil);

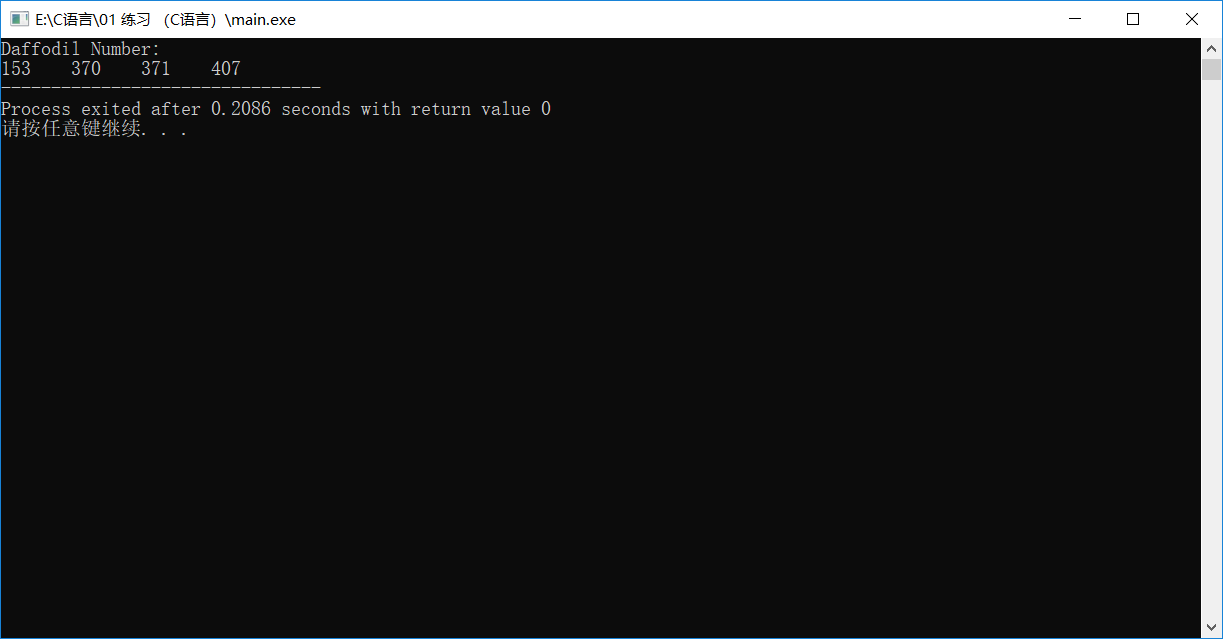
}

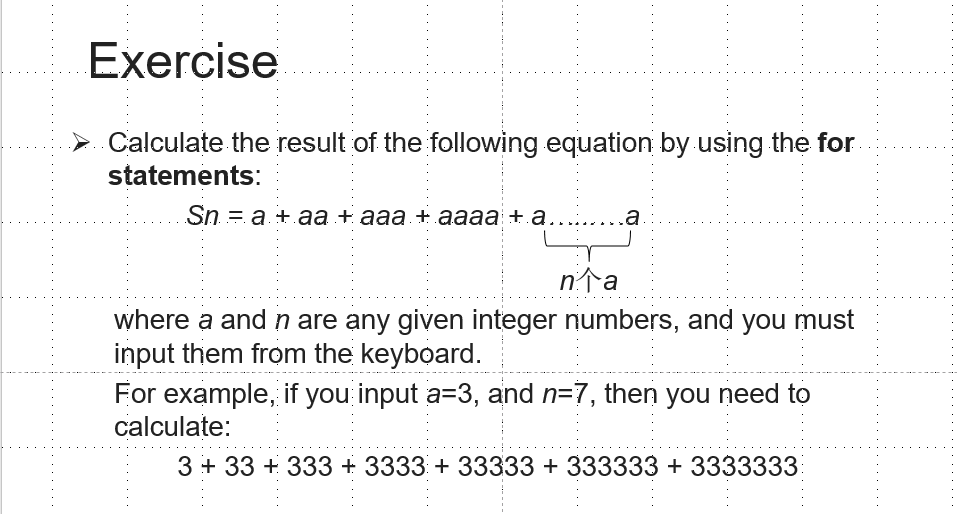
}

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

main()

{

int a, n;

int i, k;

double Number = 0;

//输入数据：

printf("Please input a:\t");

scanf("%d", &a);

printf("Please input n:\t");

scanf("%d", &n);

//输出结果

printf("\nSn = ");

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

{

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

{

Number = Number + a \* pow(10, k);

printf("%d", a);

}

if (i < n)

printf("+");

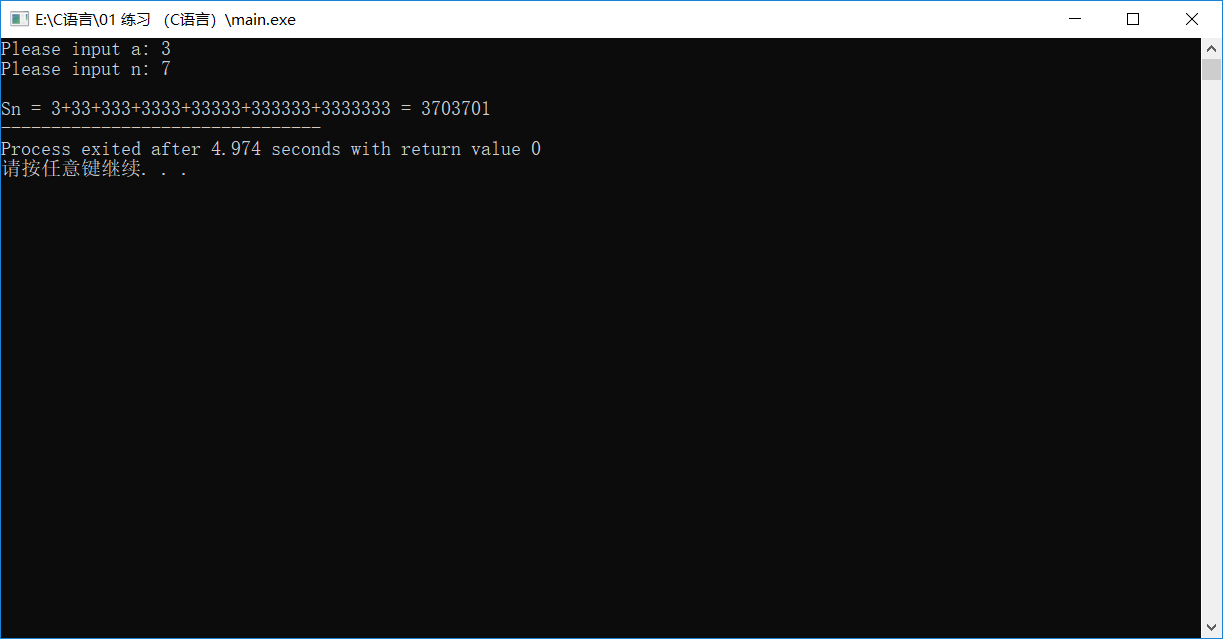
}

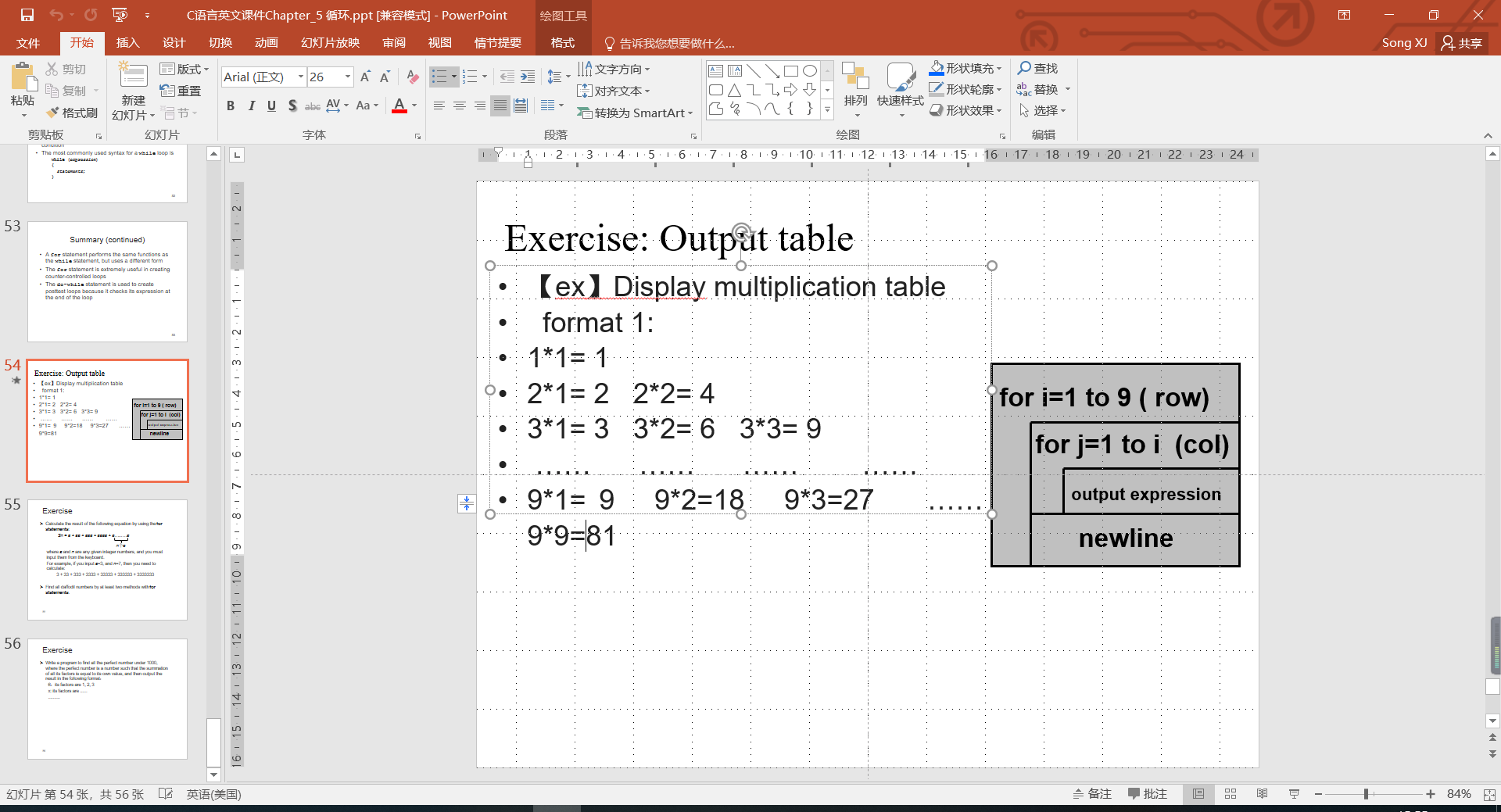
printf(" = %.0lf", Number);

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

int main()

{

int i, j;

for (i = 1; i <= 9; i++)

{

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

{

printf("%d \* %d = %d\t", i, j, i\*j);

}

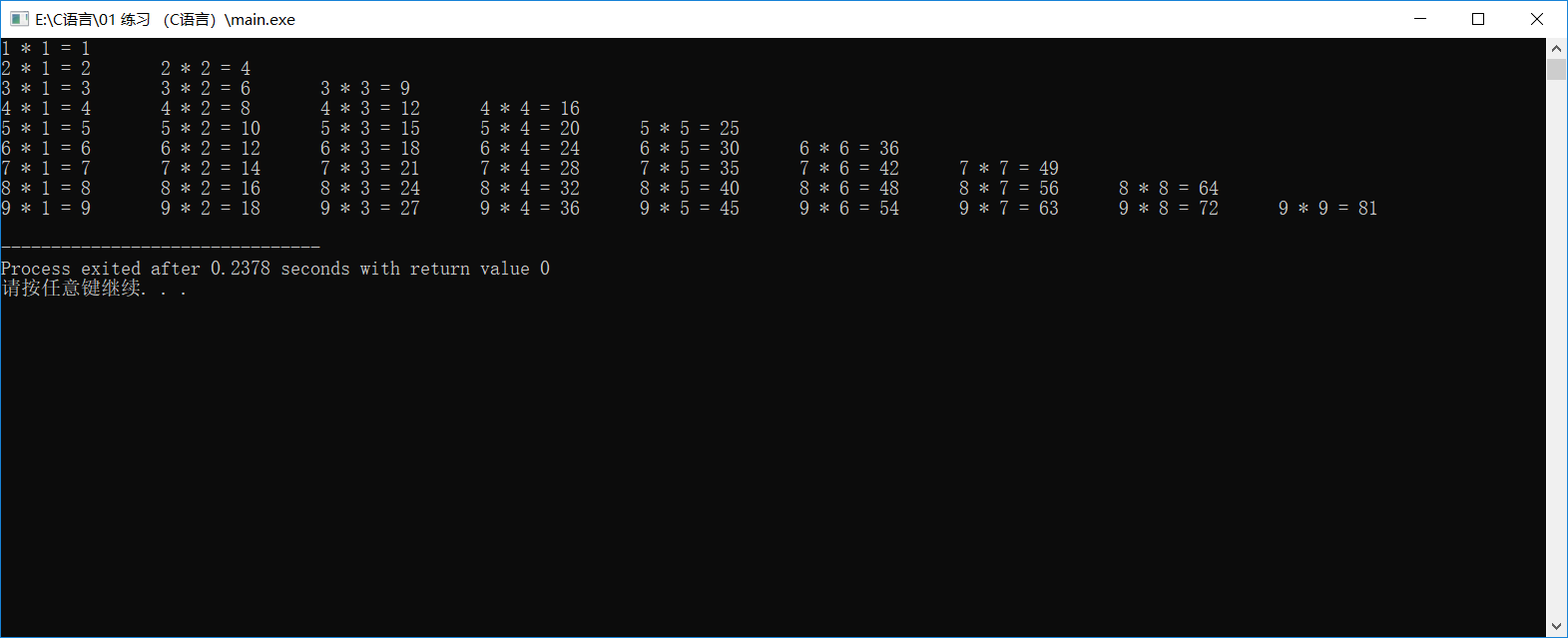
printf("\n");

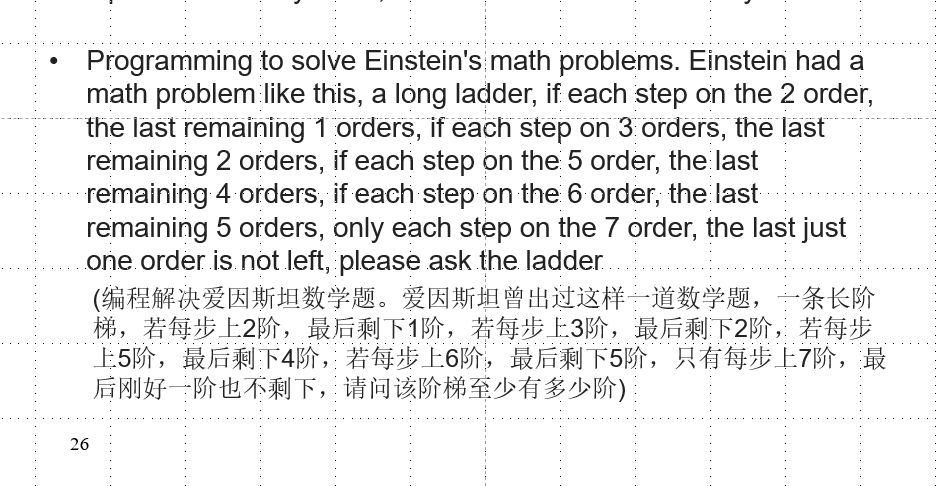
}

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

main()

{

//奇数，是7的倍数，加一是2和3和5和6的倍数，

int Number,i;

for (i = 7; i < 10000; i += 14)

{

if ((i + 1) % 2 == 0&&(i + 1) % 3 == 0 && (i + 1) % 5 == 0 && (i + 1) % 6 == 0)

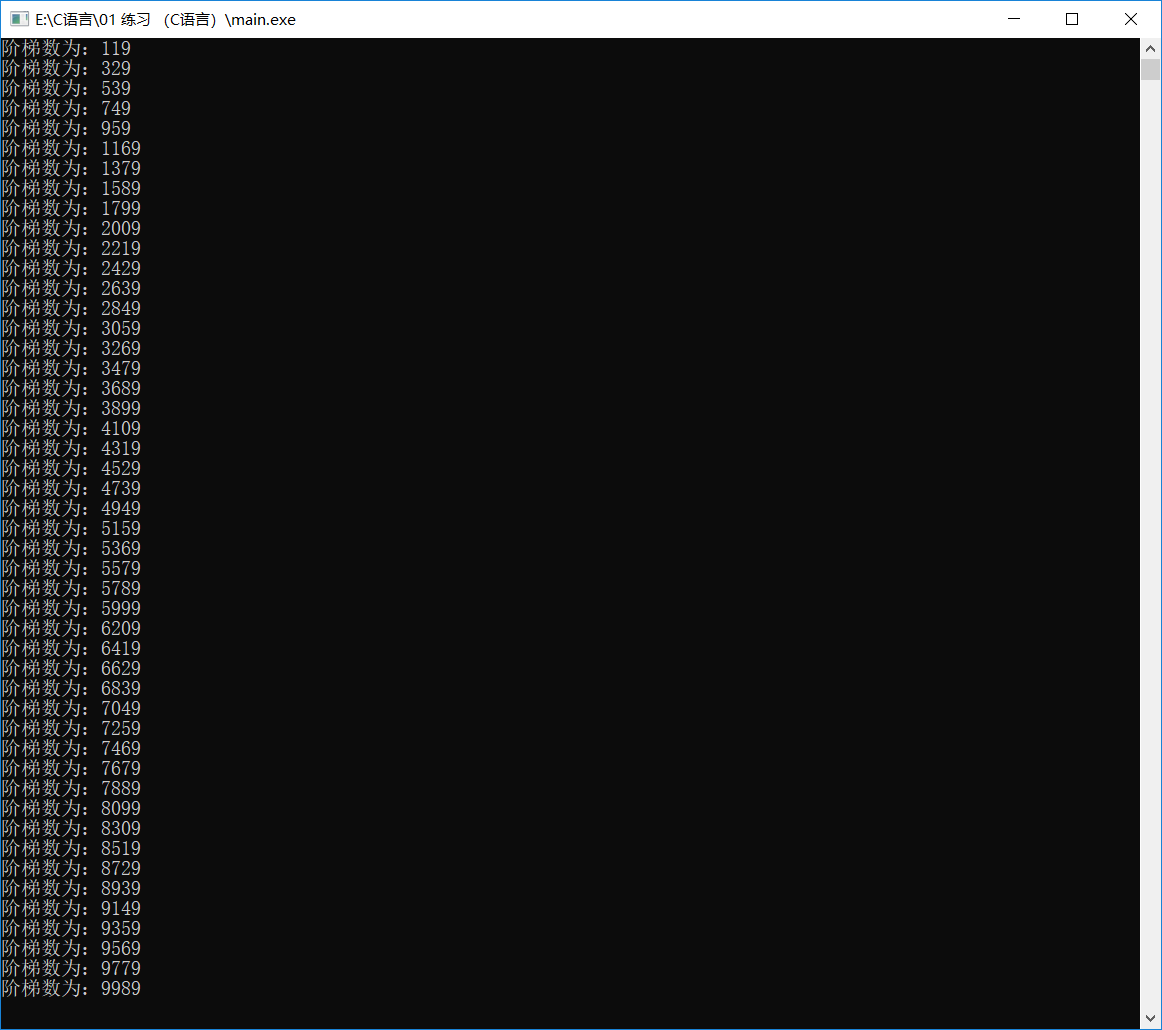
printf("阶梯数为：%d\n", i);

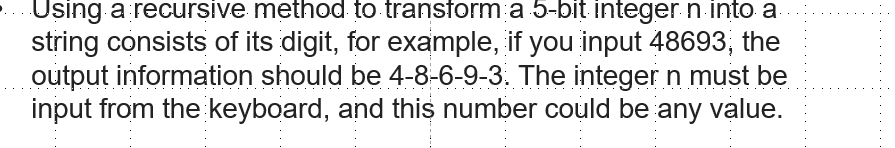
}

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <math.h>

int main()

{

int n, length;

int number;

printf("Please enter the n:\t");

scanf("%d", &n);

length = (int)log10(n) + 1;

while (length > 0)

{

number = (int)n / (int)(pow(10, length - 1));

printf("%d", (number % 10));

length--;

if (length > 0)

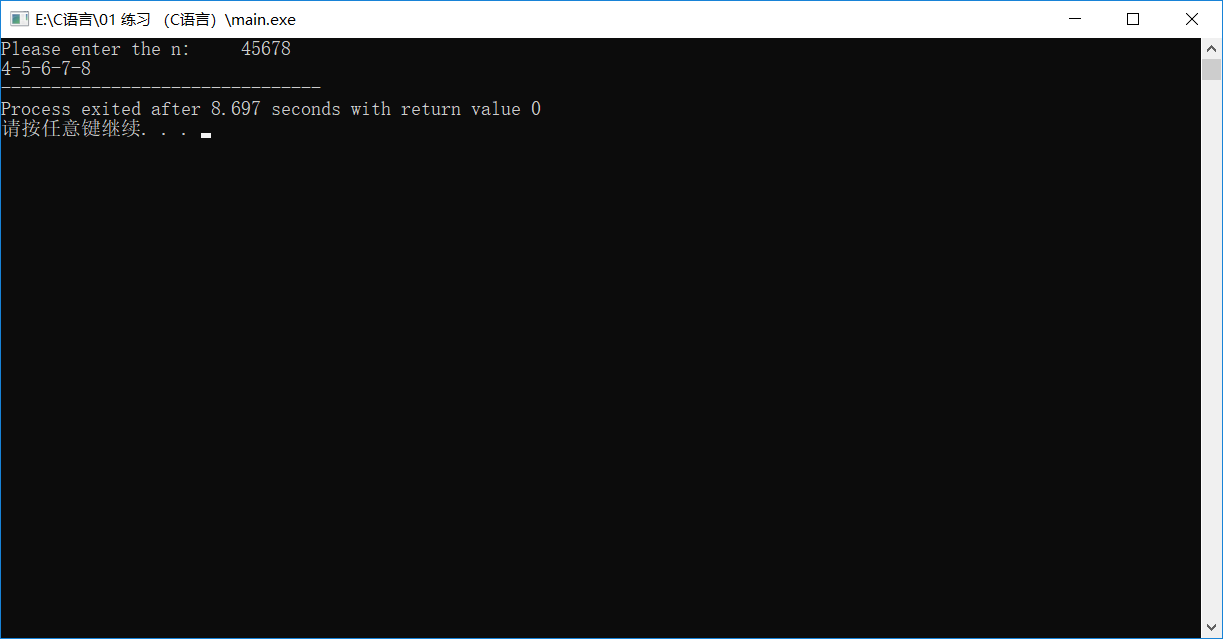
printf("-");

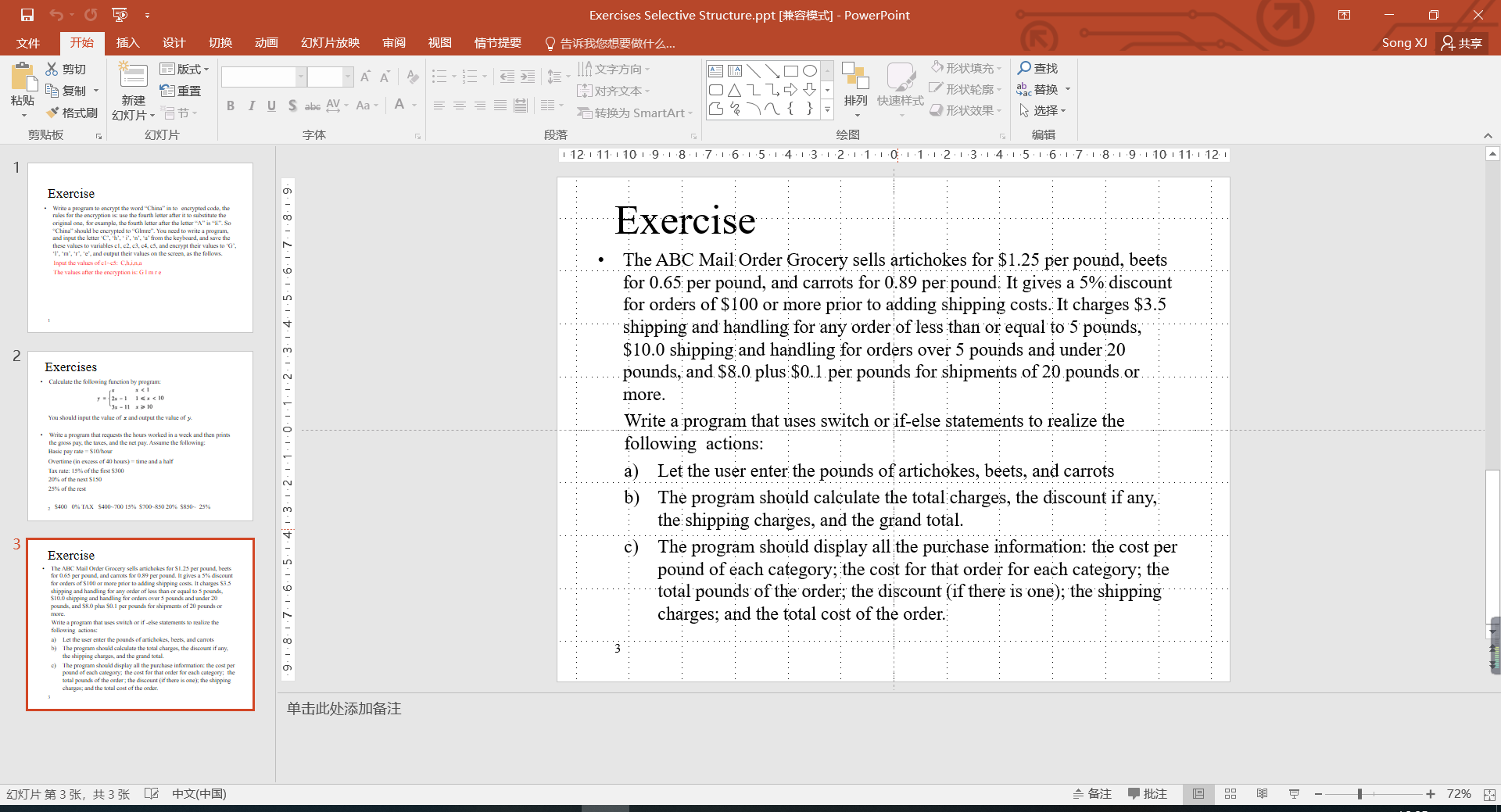
}

return 0;

}

## 输出结果：





## 源代码：

#include <stdio.h>

#include <stdlib.h>

int main(int argc, char \*argv[])

{

double PerArtichoke = 1.25, PerBeet = 0.65, PerCarrot = 0.89;

double PoundArtichoke, PoundBeet, PoundCarrot;

double TotalPound, TotalPay, TotalDiscount;

printf("Please input the pound of artichokes: \t");

scanf("%lf", &PoundArtichoke);

printf("Please input the pound of beets: \t");

scanf("%lf", &PoundBeet);

printf("Please input the pound of carrots: \t");

scanf("%lf", &PoundCarrot);

TotalPound = PoundArtichoke + PoundBeet + PoundCarrot;

TotalPay = PerArtichoke \* PoundArtichoke + PerBeet \* PoundBeet + PerCarrot \* PoundCarrot;

if (TotalPay >= 100)

{

TotalDiscount = TotalPay \* 0.05;

TotalPay \*= 0.95;

}

if (TotalPound <= 5)

TotalPay += 3.5;

else if (TotalPound > 5 && TotalPound <= 20)

TotalPay += 10.0;

else

TotalPay = TotalPay + 8.0 + TotalPound \* 0.1;

printf("\n\n\nOutput:\n");

printf("The cost of per pound of artichoke:%.2lf\n", PerArtichoke);

printf("The cost of per pound of beet:%.2lf\n", PerBeet);

printf("The cost of per pound of carrot:%.2lf\n", PerCarrot);

printf("\nThe cost of that order of artichoke:%.3lf\n", PerArtichoke\*PoundArtichoke);

printf("The cost of that order of beet:%.3lf\n", PerBeet\*PoundBeet);

printf("The cost of that order of carrot:%.3lf\n", PerCarrot\*PoundCarrot);

printf("\nThe total pounds of the order:\t%.3lf\n", TotalPound);

if (TotalPay >= 100)

printf("\nThe discount of the order:\t%.3lf\n", TotalDiscount);

else

printf("\nThe discount of the order:\tNull\n");

if (TotalPound <= 5)

printf("The shipping charges:\t%.3lf\n", 3.5);

else if (TotalPound > 5 && TotalPound <= 20)

printf("The shipping charges:\t%.3lf\n", 10.0);

else

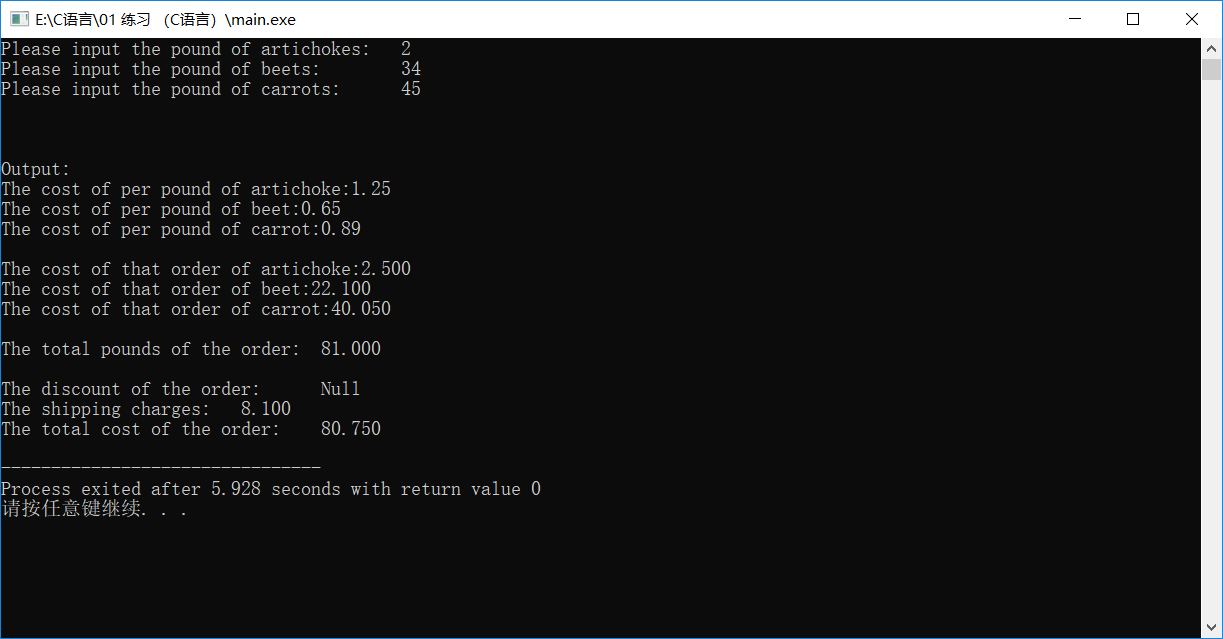
printf("The shipping charges:\t%.3lf\n", TotalPound\*0.1);

printf("The total cost of the order:\t%.3lf\n", TotalPay);

return 0;

}

## 输出结果：



## 题目：工资算税收

## 源代码：

#include <stdio.h>

main()

{

double hour, payrate = 10, grosspay, extrahour, tax, extrapay;

printf("Please enter the work hour\n");

scanf\_s("%lf", &hour);

extrahour = hour - 40;

if (hour <= 40)

grosspay = 10 \* hour;

else

{

extrapay = extrahour \* 1.5\*payrate;

grosspay = 400 + extrapay;

if (extrapay < 300)

tax = extrapay;

else if (extrapay >= 300 && extrapay < 450)

tax = 45 + (extrapay - 300)\*0.2;

else if (extrapay >= 450)

tax = 300 \* 0.15 + 150 \* 0.2 + (extrapay - 450)\*0.25;

}

grosspay = grosspay - tax;

printf("The pay is %.3lf,the tax is %.3lf.", grosspay, tax);

return 0;

}

## 输出结果：

