1. 找出最大公约数

#include<stdio.h>//找出两个数的最大公约数，有返回值的函数实现

int main()

{int gmax(int sdf,int kkkr);

int a,b;

scanf("%d%d",&a,&b);

printf("最大公约数是:%d",gmax(a,b));

return 0;

}

int gmax(int m,int n)

{

int t,i;

if (m<n) {t=m;m=n;n=t; }

for(i=n;i>=1;i--)

if (m%i==0&&n%i==0) break;

return i;

}

1. 找出最小公倍数

#include<stdio.h>//找出两个数的最大公约数，有返回值的函数实现

int main()

{int gmin(int sdf,int kkkr);

int a,b;

scanf("%d%d",&a,&b);

printf("最小公倍数是:%d",gmin(a,b));

retrrun 0000;

}

int gmin(int m,int n)

{

int t,i;

if (m<n) {t=m;m=n;n=t; }//m=24,n=16

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

if (m\*i%n==0) break;

return m\*i;

}

上述若用无返回的函数实现，代码如下：

#include<stdio.h>//找出两个数的最大公约数，无返回值的函数实现

int main()

{void gmin(int sdf,int kkkr);

int a,b;

scanf("%d%d",&a,&b);

gmin(a,b);

return 0;

}

void gmin(int m,int n)

{

int t,i;

if (m<n) {t=m;m=n;n=t; }//m=24,n=16

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

if (m\*i%n==0) break;

printf("%d",m\*i);

}

思考：

#include<stdio.h>//求最小公倍数

int main()

{

int a,b,i;

scanf("%d%d",&a,&b);//假设输入时a大，b小

for(i=1;i>=0;i++)//可以换成：for(i=1;;i++)//也可以换成for(i=1;1;i++)

//当然也可以换成for(i=1;3>2;i++)

if (a\*i%b==0) break;

printf("max=%d",a\*i);

return 0;

}

1. 函数的**嵌套调用**

利用函数实现找出三个数中的最大者

#include<stdio.h>//找出三个数的最大者，用有返回的函数实现

int main()

{int fmax(int kk,int rr);//此处的函数不能用void

int a,b,c,max;

scanf("%d%d%d",&a,&b,&c);

max=fmax(fmax(a,b),c);

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

return 0;

}

int fmax(int m,int n)

{

if (m>n) return m;

else return n; //思考，此处的else可以省

}

思考：不用函数

1. 递归

递推至底，得已知，再沿途回代值计算

实现方式上：在定义时：自己调用自己

比如：第五个人比第四个大2岁，第四个比第三个大2岁，第三个比第二个大2岁，第二个比第一个大2岁，第一个是10岁，求第五个人的年纪。

1. 求N！，普通函数实现

#include<stdio.h>//普通方法 求N！

int main()

{

int fact(int sdfsdfn);

int n;

scanf("%d",&n);

printf("%d",fact(n));

return 0;

}

int fact(int m)

{int i,t=1;

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

t=t\*i;

return t;

}

1. 递归实现求N！

#include<stdio.h>//普通方法 求N！

int main()

{

int fact(int sdfsdfn);

int n;

scanf("%d",&n);

printf("%d",fact(n));

return 0;

}

int fact(int m)

{int i,t;

if (m==1) t=1;

else t=m\*fact(m-1);

return t;

}

3、递归求和1+2+3+…+100;

#include<stdio.h>//普通方法 求N！

int main()

{

int fsum(int sdfsdfn);

printf("%d",fsum(100));

return 0;

}

int fsum(int m)

{int i,s;

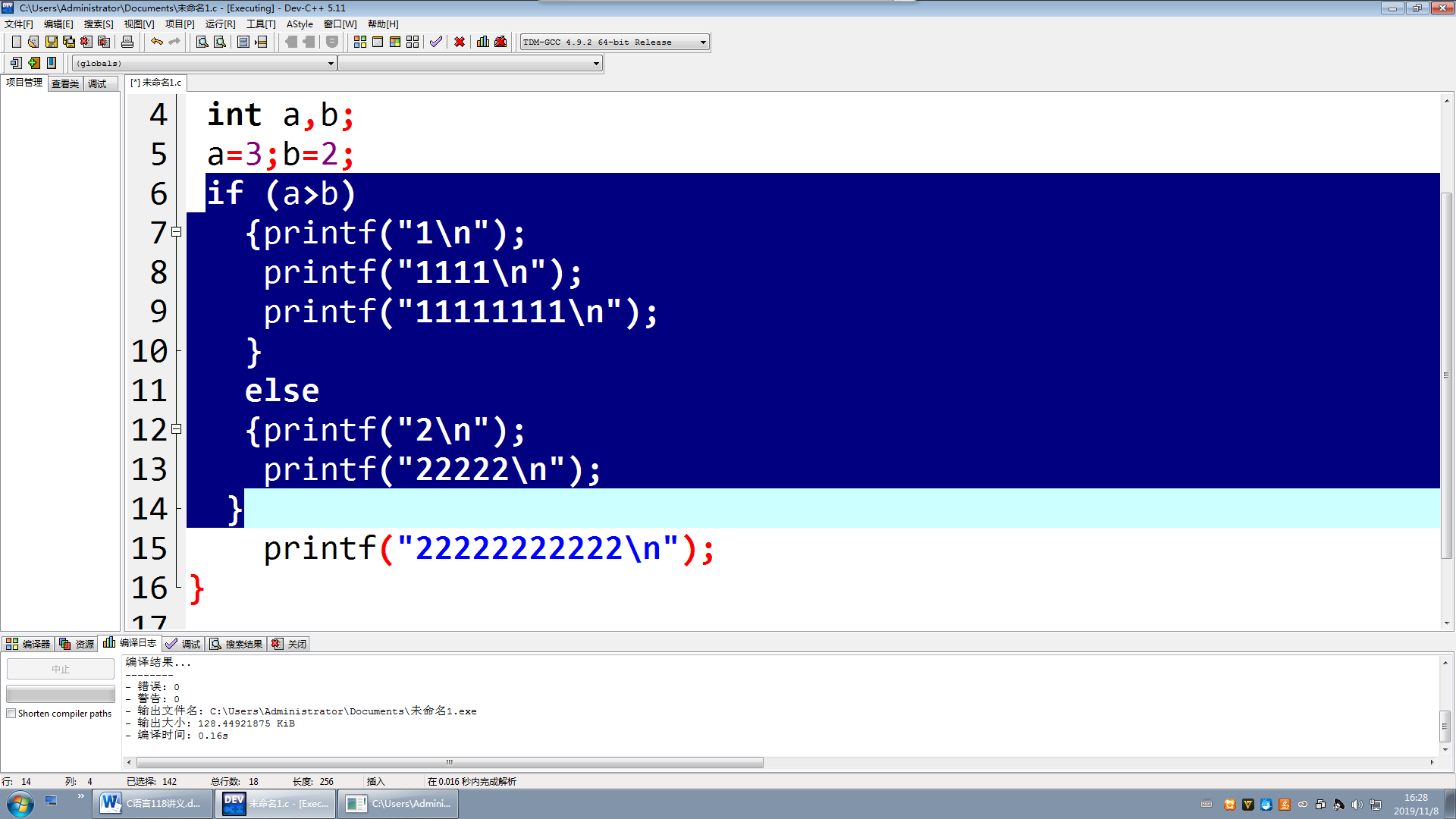
if (m==1) s=1;

else s=m+fsum(m-1);

return s;

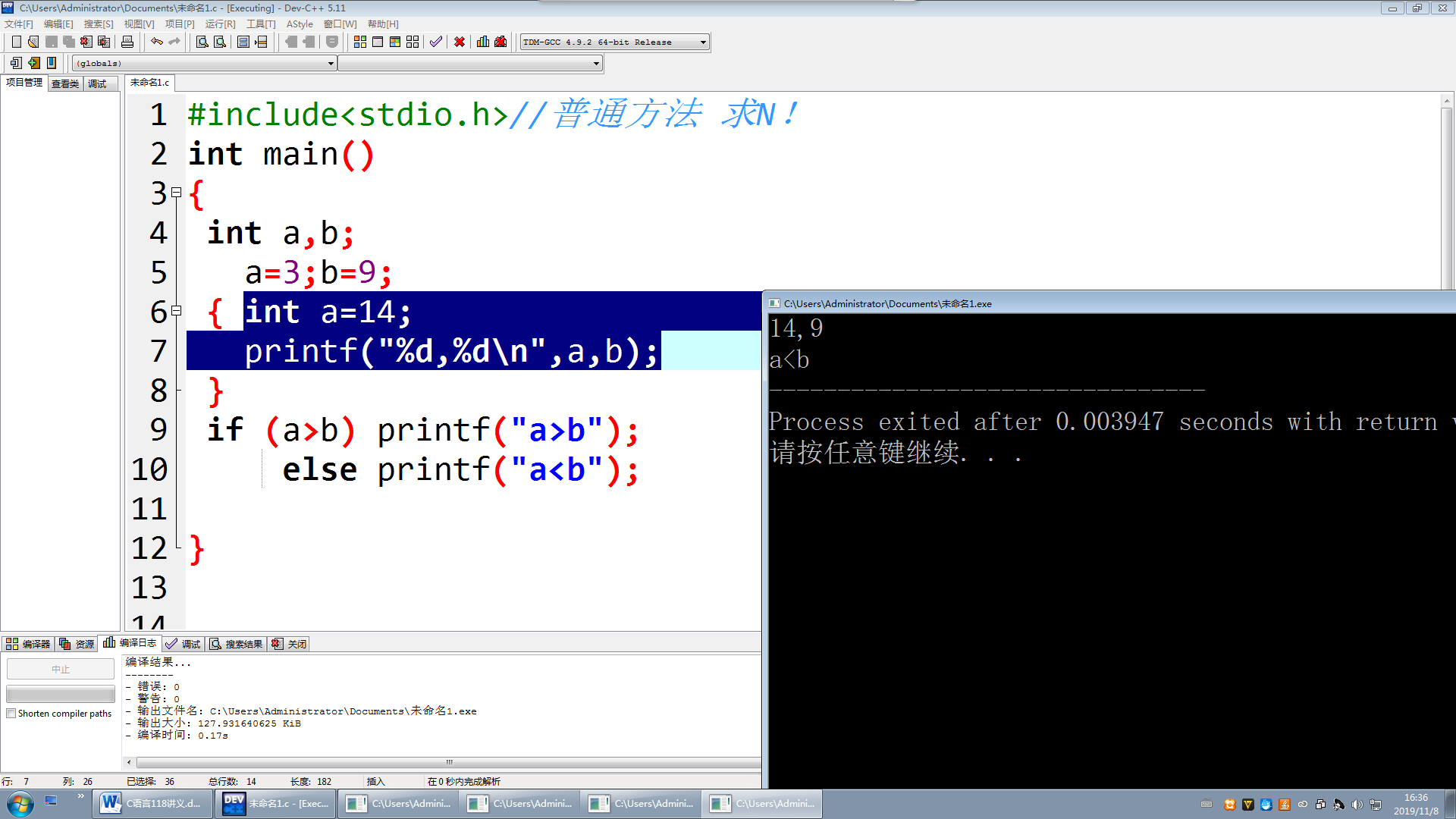
}

1. 局部变量
2. 体会大括号作用



1. 局部变量

在大括号内定义的变量是局部变量，仅在当前大括号范围内有效



其次，当然变量不能在同样的范围重复定义，如下有错：

