在C#中，不同的数据类型混合在一个表达式中，会根据提升规则进行类型转换。

表达式提升规则：

IF一个操作数为decimal类型，THEN其他操作数提升为decimal类型（如果操作数为float或double，则产生错误）。

ELSE IF一个操作数为double类型，则第二个操作数提升为double类型

ELSE IF 一个操作数为float类型，则第二个操作数提升为float类型

ELSE IF 一个操作数为ulong类型，则第二个操作数提升为ulong类型（如果操作数为sbyte、short、int或long，则产生错误）

ELSE IF 一个操作数为long类型，则第二个操作数提升为long类型

ELSE IF 一个操作数为uint类型，而第二个操作数为sbyte、short或int（有符号整数），则两个 操作数都提升为long

ELSE IF 一个操作数为uint类型，则第二个操作数提升为uint

ELSE 两个操作数都提升为int类型。

对于最后一条规则，如果没有其他规则适用，则操作数都提升为int类型，表明所有算术运算的结果都不会小于int类型。

// Copyright 2016.刘珅珅

// author：刘珅珅

// 表达式类型转换

using *System*;

using *System*.*Collections*.*Generic*;

using *System*.*Linq*;

using *System*.*Text*;

using *System*.*Threading*.*Tasks*;

namespace type\_convert\_test1

{

class TypeConvertTest

{

static void Main(string[] args)

{

decimal d = 9.5m;

float f = 4.5f;

// error，decimal类型不能和float、double混用

// Console.WriteLine("d \* f = {0}", d \* f);

ulong ul = 245UL;

int i = 5;

// error，ulong不能和任何有符合整数类型混用

// Console.WriteLine("ul \* i = {0}", ul \* i);

// ulong可以和有符合的浮点型混用，结果为float或double

float f1 = ul \* f;

*Console*.*WriteLine*("ul \* f = {0}", ul \* f);

uint ui = 5;

// error，uint和sbyte，short，int混用时，结果为long

// int b = ui \* i;

long l = ui \* i;

ushort us = 23;

byte b = 2;

char ch = 'X';

// uint和ushort，byte，char混用时，结果为uint

uint ua = ui \* us;

uint ub = ui \* b;

uint uc = ui \* ch;

byte b1 = 3;

// byte和byte混用时，结果为int

int i1 = b1 \* b;

}

}

}