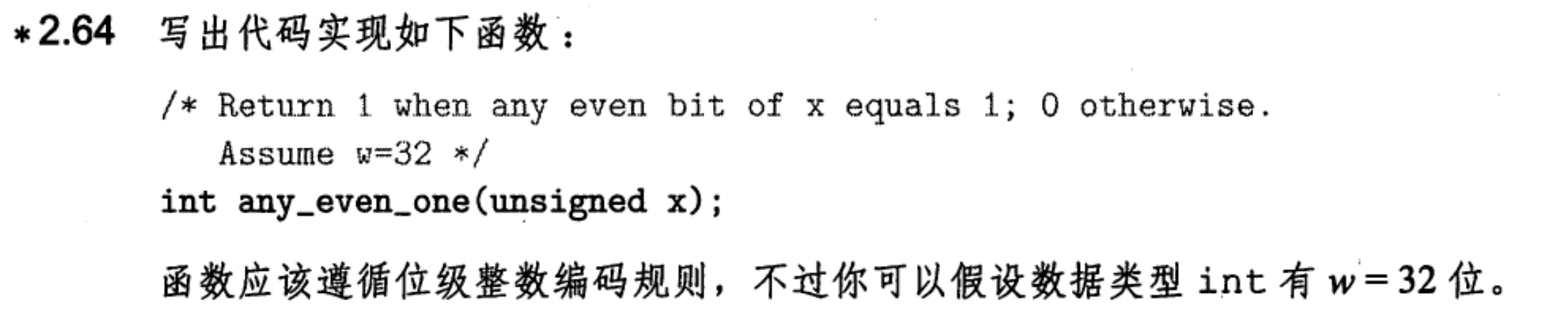
1. **Finish your homework independently**
2. **Convert this docx to pdf: “stuID\_name\_csapp1.pdf”**

**Example: ”2017010000\_zhangsan\_csapp1.pdf”**

1. **Submit this pdf: learn.tsinghua.edu.cn**

练习题：2.64；2.73；2.81；



#include <iostream>

int **any\_even\_one**(unsigned x);

int **main**(){

**std**::cout**<<any\_even\_one**(0x82);

**std**::cout**<<any\_even\_one**(0x51);

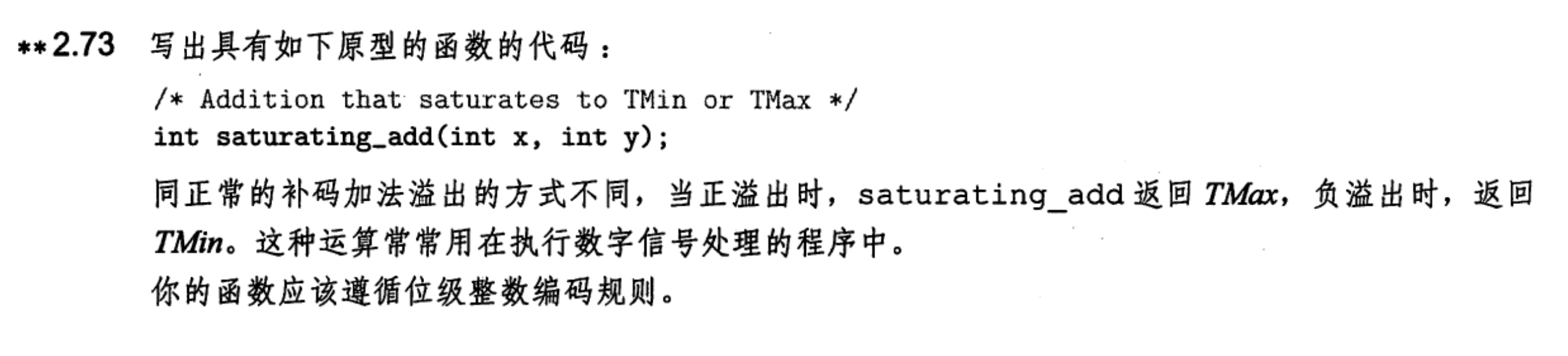
    return 0;

}

int **any\_even\_one**(unsigned x){

    return (x & 0x55555555) != 0;

}



#include<iostream>

#include<climits>

int **saturating\_add**(int x, int y);

int **main**(){

**std**::cout**<<saturating\_add**(999,-10000)**<<std**::**endl**;

**std**::cout**<<saturating\_add**(**INT\_MAX**-200,55)**<<std**::**endl**;

**std**::cout**<<saturating\_add**(**INT\_MAX**,10000)**<<std**::**endl**;

**std**::cout**<<saturating\_add**(**INT\_MIN**,10000)**<<std**::**endl**;

**std**::cout**<<saturating\_add**(**INT\_MIN**,-100)**<<std**::**endl**;

**std**::cout**<<saturating\_add**(**INT\_MAX**-100,45)**<<std**::**endl**;

}

int **saturating\_add**(int x, int y){

    int PosOverflow = (x > 0 && y > 0 && x + y <= 0);

    int NegOverflow =  (x < 0 && y < 0 && x + y > 0);

    int Overflow = PosOverflow | NegOverflow;

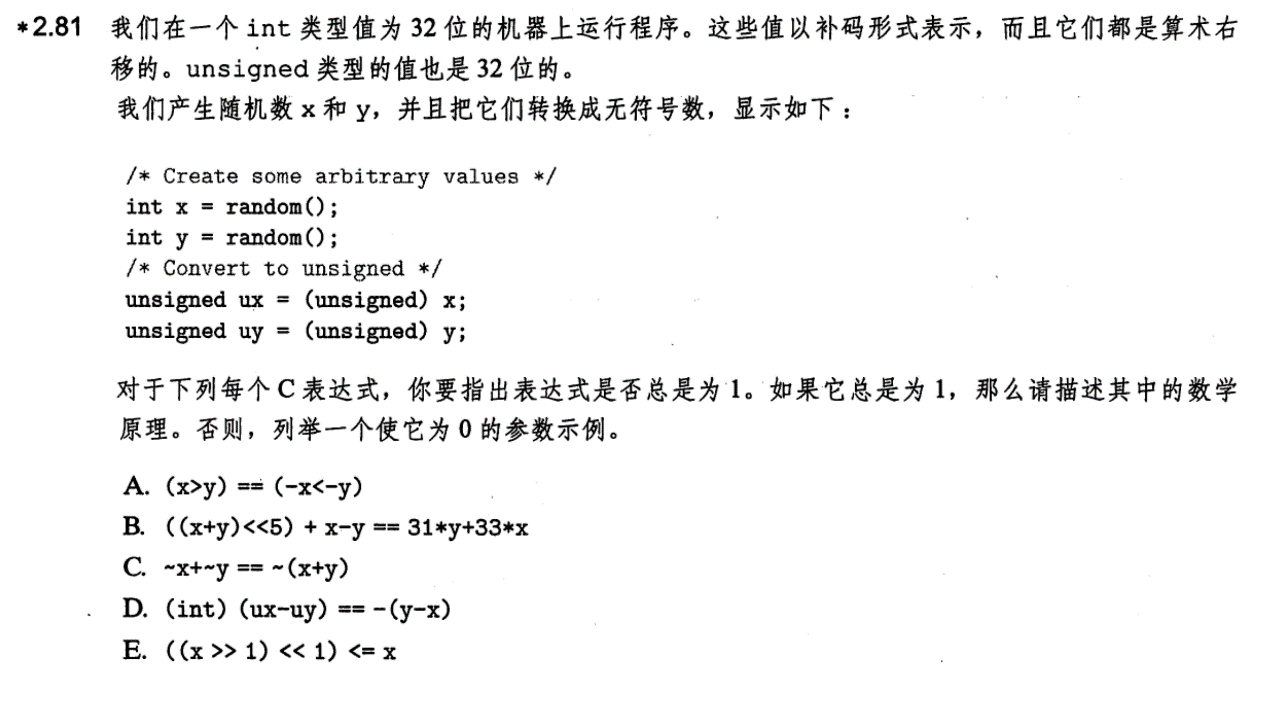
    int sum = PosOverflow & **INT\_MAX** |

            NegOverflow & **INT\_MIN** |

            ~Overflow & (x + y);

    return sum;

}



A：不总为1。如果，。由于，此时逻辑为0。

B：总为1。担心如果x极小，y极大，使得(x+y)<<5正溢出，而31\*y+33\*x不发生正溢出，则逻辑为0。x=1，y=7，验证为真，实际上该式符合补码运算规律。

C：不总为1。如x=1，y=0。

D：总为1。位级表示相同，而且最后都转换成int类型。

E：总为1。先右移一位再（算数）左移一位，可能会导致结果比原来小1或者相等。