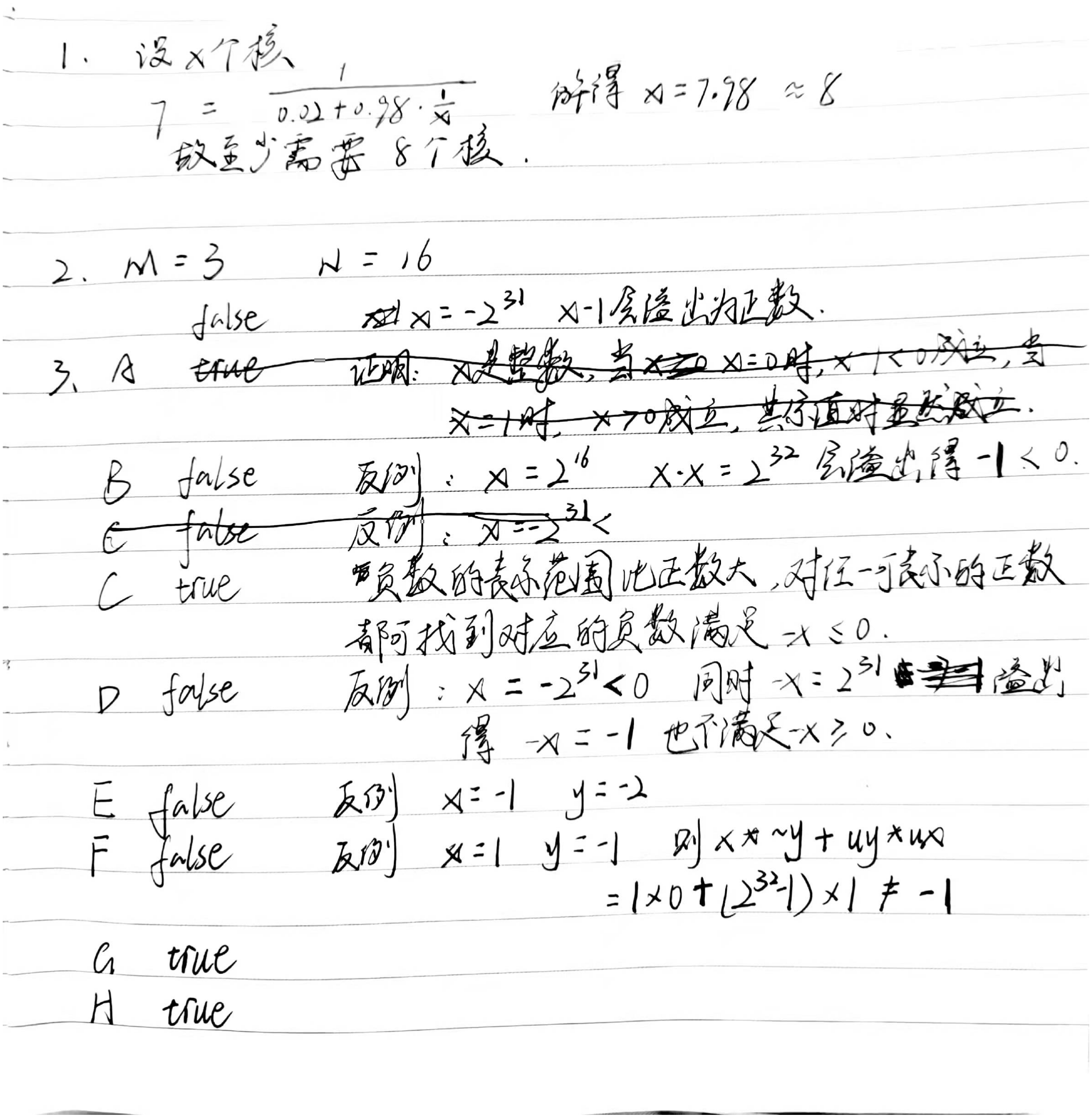
作业1

一、思考题



1. 设 x 个核

$7 = \frac{1}{0.02 + 0.98 \cdot \frac{1}{x}}$ 解得 $x = 7.98 \approx 8$

故至少需要 8 个核.

2. $M = 3$ $N = 16$

3. A false ~~x~~ $x = -2^{31}$ $x-1$ 会溢出为正数.

~~true 证明: x 是整数, 当 $x \geq 0$ $x=0$ 时, $x-1<0$ 成立, 当 $x=1$ 时, $x>0$ 成立, 其余值时显然成立.~~

B false 反例: $x = 2^{16}$ $x \cdot x = 2^{32}$ 会溢出, 得 $-1 < 0$.

~~C false 反例: $x = -2^{31} <$~~

C true 负数的表示范围比正数大, 对任一可表示的正数都可找到对应的负数满足 $-x \leq 0$.

D false 反例: $x = -2^{31} < 0$ 同时 $-x = 2^{31}$ 溢出得 $-x = -1$ 也不满足 $-x \geq 0$.

E false 反例 $x = -1$ $y = -2$

F false 反例 $x = 1$ $y = -1$ 则 $x * \sim y + uy * ux$

$= 1 \times 0 + (2^{32}-1) \times 1 \neq -1$

G true

H true

4. (1) N　　反例 $x=-2^{31}, y=-1$，$-x=-2^{31}, -y=1$

$(x<y)=1$　$(-x>-y)=0$

(2) Y

(3) Y

(4) ~~N~~　~~反例 $x=-2^{32}$ $x=-1, y=0$，$ux=2^{32}-1$ $uy=0$~~

~~$ux-uy=2^{32}-1 \neq -$~~

Y

(5) Y

(6) Y

(7) N　　$x=2^{31}-1$ 转float时，会有部分位数被舍去.

(8) N　　$x+y$ 溢出，$x=2^{31}-1$ $y=1$

(9) Y

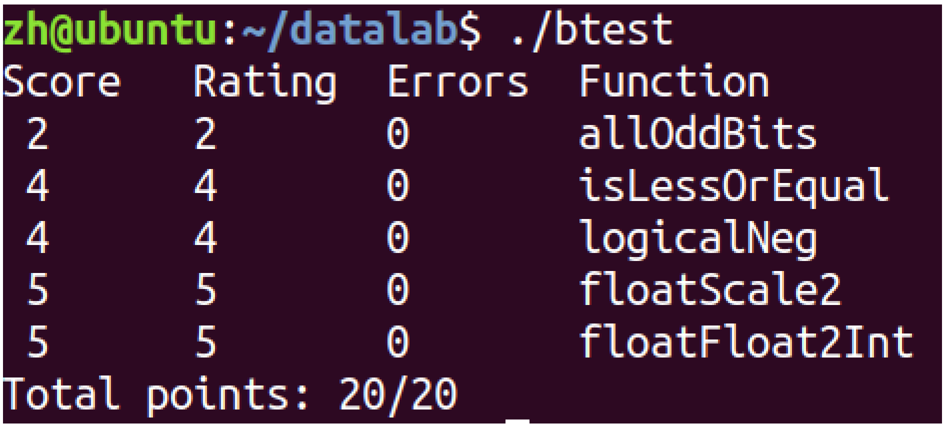
(10) ~~N~~　~~反例：尾数最多有31位，两个double相乘最多产生~~ ~~多~~ ~~62位尾数，而double最多只会有~~ ~~52位~~ ~~53位，会发生舍入，因此改变运算顺序会改变结果.~~

~~可能.~~

(10) Y

二、实践题

result：



```
zh@ubuntu:~/datalab$ ./btest
Score   Rating  Errors  Function
 2       2       0       allOddBits
 4       4       0       isLessOrEqual
 4       4       0       logicalNeg
 5       5       0       floatScale2
 5       5       0       floatFloat2Int
Total points: 20/20
```

codes:

- allOddBits：

```c
int allOddBits(int x) {
    int s = 0x00000055;
    int y = s | (s << 8) | (s << 16) | (s << 24);
    int r = y | x;
    return !(r ^ (~0));
}
```

- isLessOrEqual：

```c
int isLessOrEqual(int x, int y) {
//your codes here
        int x1=(x>>31)&0x1;
        int y1=(y>>31)&0x1;
        int nepo=x1&(~y1);
        int pone=(~x1)&y1;
        int diff=(~x+1)+y;
        int d=(diff>>31)&0x1;
        return nepo|(!pone&!d);
}
```

- logicalNeg：

```c
int logicalNeg(int x) {
//your codes here
return ((x | (~x +1)) >> 31) + 1;
}
```

- floatScale2：

```c
unsigned floatScale2(unsigned uf) {
    //your codes here
    unsigned sign = (uf >> 31) & 0x1;
    unsigned exp = (uf >> 23) & 0x000000FF;
    unsigned frac = ((0x0000007F << 16) | (0x000000FF << 8) | (0x000000FF)) & uf;
    if (exp == 0xFF)return uf;
    if (exp == 0)return (sign << 31) | (exp << 23) | (frac << 1);
    return (sign << 31) | ((++exp) << 23) | frac;
}
```

- floatFloat2Int:

```c
int floatFloat2Int(unsigned uf) {
    //your codes here
    unsigned sign = (uf >> 31) & 0x1;
    unsigned exp = (uf >> 23) & 0x000000FF;
    unsigned frac = ((0x0000007F << 16) | (0x000000FF << 8) | (0x000000FF)) & uf;
    int E = exp - 127;
    if (E < 0)return 0;
    else if (E >= 31)return 0x80000000;
    else {
        frac = frac | (1 << 23);
        if (E > 23)frac <<= (E - 23);
        else frac >>= (23 - E);
        if (sign)return -frac;
        else return frac;
    }
}
```