```
上机作业
1.
源代码
#include<stdio.h>
# define PI 3.14159265
void main() {
    double r, h, s, ssum, v;
    printf("Please input the radius and the height of the cylinder:");
    scanf ("%lf%lf", &r, &h);
    s=2*PI*r*h;
    ssum=s+2*PI*r*r;
    v=PI*r*r*h:
    printf("The lateral area of the cylinder is %.4f\n", s);
    printf("The sum of the area of the cylinder is %.4f\n", ssum);
    printf("The volume of the cylinder is %.4f\n", v);
}
```

```
#include(stdio.h)
# define PI 3.14159265
∃void main() {
    double r, h, s, ssum, v;
    printf("Please input the radius and the height of the cylinder:");
    scanf ("%lf%lf", &r, &h);
    s=2*PI*r*h;
    ssum=s+2*PI*r*r;
    v=PI*r*r*h;
    printf("The lateral area of the cylinder is %.4f\n",s);
    printf("The sum of the area of the cylinder is %.4f\n", ssum);
    printf("The volume of the cylinder is \%. 4f\n", v);
         C:\WINDOWS\system32\cmd. X
        Please input the radius and the height of the cylinder:3.5 4.6
        The lateral area of the cylinder is 101.1593
        The sum of the area of the cylinder is 178.1283
        The volume of the cylinder is 177.0287
        请按任意键继续.
```

```
2.
  (1) #include<stdio.h>
#include<math.h>
#define F(x, y, z) sqrt((y)*(y)-4*(a)*(c))
void main() {
```

```
float a, b, c, x1, x2;

a=1;

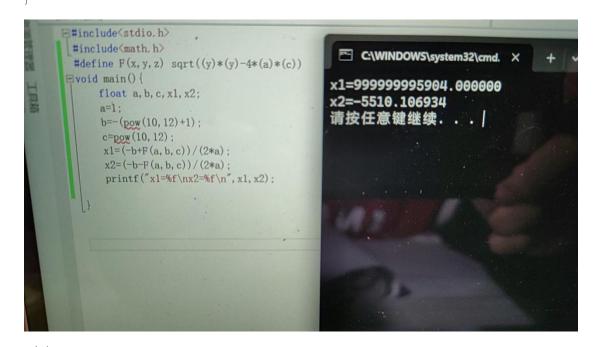
b=-(pow(10, 12)+1);

c=pow(10, 12);

x1=(-b+F(a, b, c))/(2*a);

x2=(-b-F(a, b, c))/(2*a);

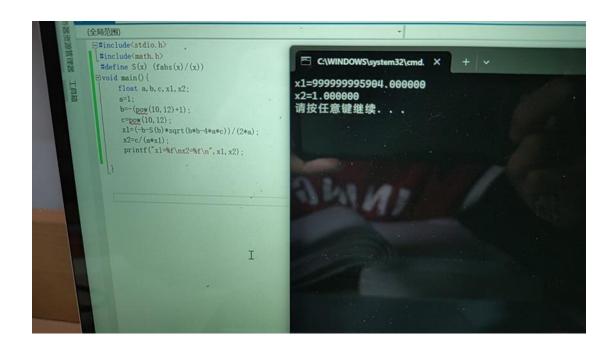
printf("x1=%f\nx2=%f\n", x1, x2);
```



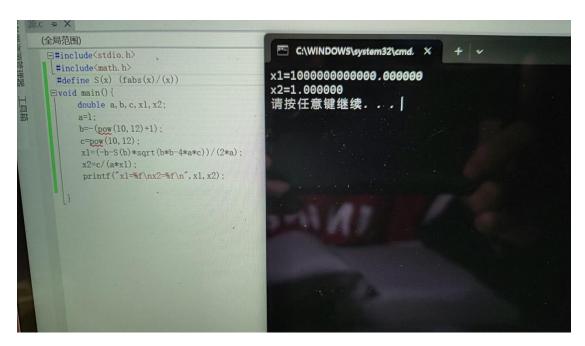
(2) #include<stdio.h> #include<math.h> #define F(x, y, z) sqrt((y)*(y)-4*(a)*(c)) void main() { double a, b, c, x1, x2; a=1; b=-(pow(10, 12)+1); c=pow(10, 12); x1=(-b+F(a, b, c))/(2*a); x2=(-b-F(a, b, c))/(2*a); printf("x1=%f\nx2=%f\n", x1, x2); }

```
(全局范围)
                                                                       - @ main()
 E#include(stdio.h)
  #include (math. h)
                                             C:\WINDOWS\system32\cmd. X
   #define F(x, y, z) sqrt((y)*(y)-4*(a)*(c))
 Evoid main() {
                                           x1=1000000000000.000000
      double a, b, c, x1, x2;
                                           x2=1.000000
       a=1:
       b=-(pow(10, 12)+1);
                                           请按任意键继续...
       c=pow(10, 12);
       x1=(-b+F(a,b,c))/(2*a);
        x2=(-b-F(a,b,c))/(2*a);
        printf("x1=%f\nx2=%f\n", x1, x2);
```

```
(3) #include<stdio.h>
#include<math.h>
#define S(x) (fabs(x)/(x))
void main() {
    float a, b, c, x1, x2;
    a=1;
    b=-(pow(10, 12)+1);
    c=pow(10, 12);
    x1=(-b-S(b)*sqrt(b*b-4*a*c))/(2*a);
    x2=c/(a*x1);
    printf("x1=%f\nx2=%f\n", x1, x2);
}
```



```
(4) #include<stdio.h>
#include<math.h>
#define S(x) (fabs(x)/(x))
void main() {
    double a, b, c, x1, x2;
    a=1;
    b=-(pow(10, 12)+1);
    c=pow(10, 12);
    x1=(-b-S(b)*sqrt(b*b-4*a*c))/(2*a);
    x2=c/(a*x1);
    printf("x1=%f\nx2=%f\n", x1, x2);
}
```



原因 double 型 64 位存储,精度高,存储 b, c 无误差 float 32 位存储, b, c 绝对值较大, 用 float 存储会有误差