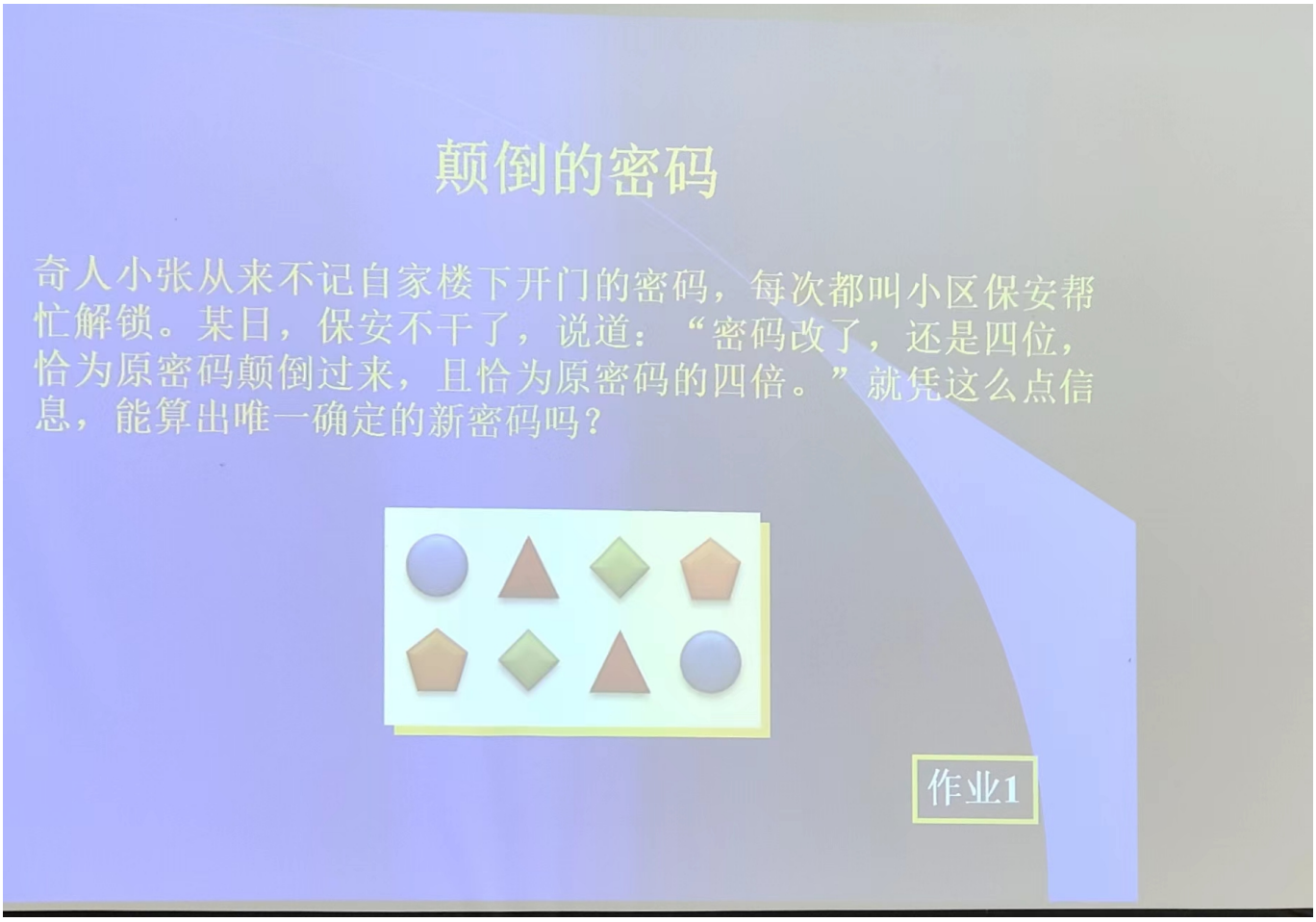
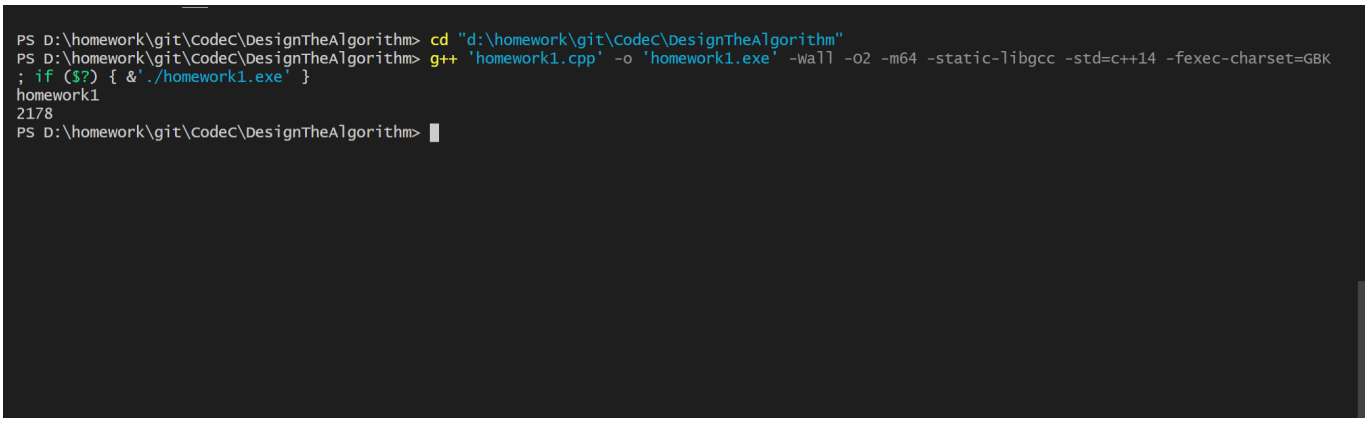
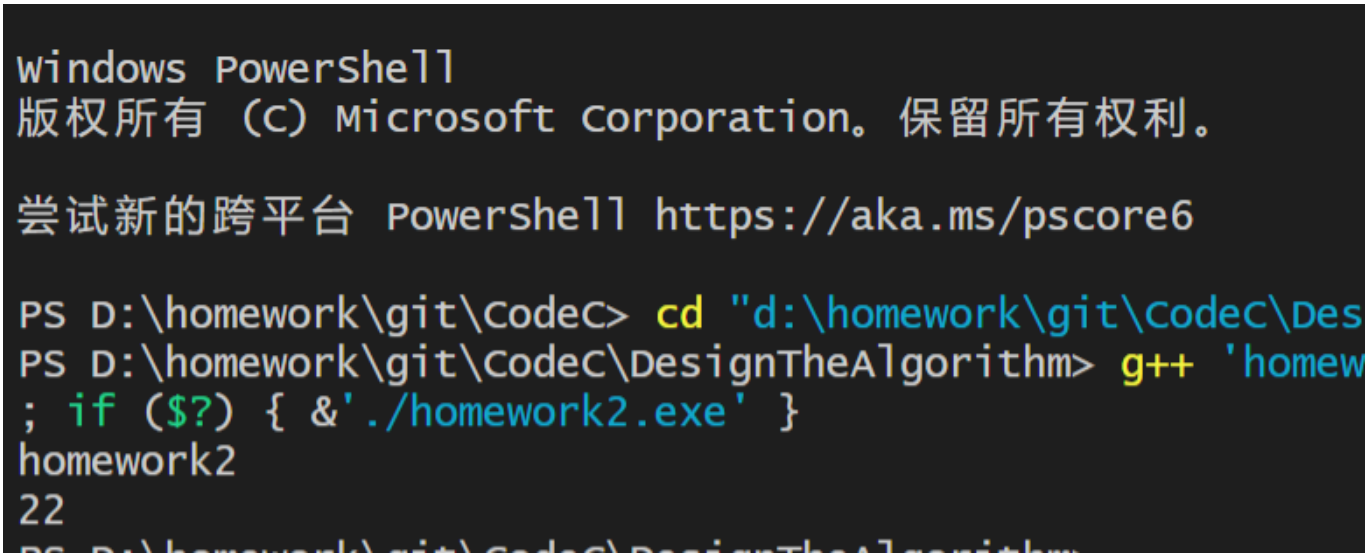



```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
  int i = 0, j = 0, k = 0, q = 0;
  int prior = 1000 * i + 100 * j + 10 * k + q;
  for (i = 1; i < 10; ++i)
    for (j = 1; j < 10; ++j)
      for (k = 1; k < 10; ++k)
        for (q = 1; q < 10; ++q)
        {
          int prior = 1000 * i + 100 * j + 10 * k + q;
          int parent = 1000 * q + 100 * k + 10 * j + i;
          if (4 * prior == parent)
          {
            printf("%d", prior);
            return 0;
          }
        }
}
```


PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework1.cpp' -o 'homework1.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework1.exe' }
homework1
2178
PS D:\homework\git\CodeC\DesignTheAlgorithm>


```c
#include <stdio.h>
int main()
{
  int cnt = 0;
```

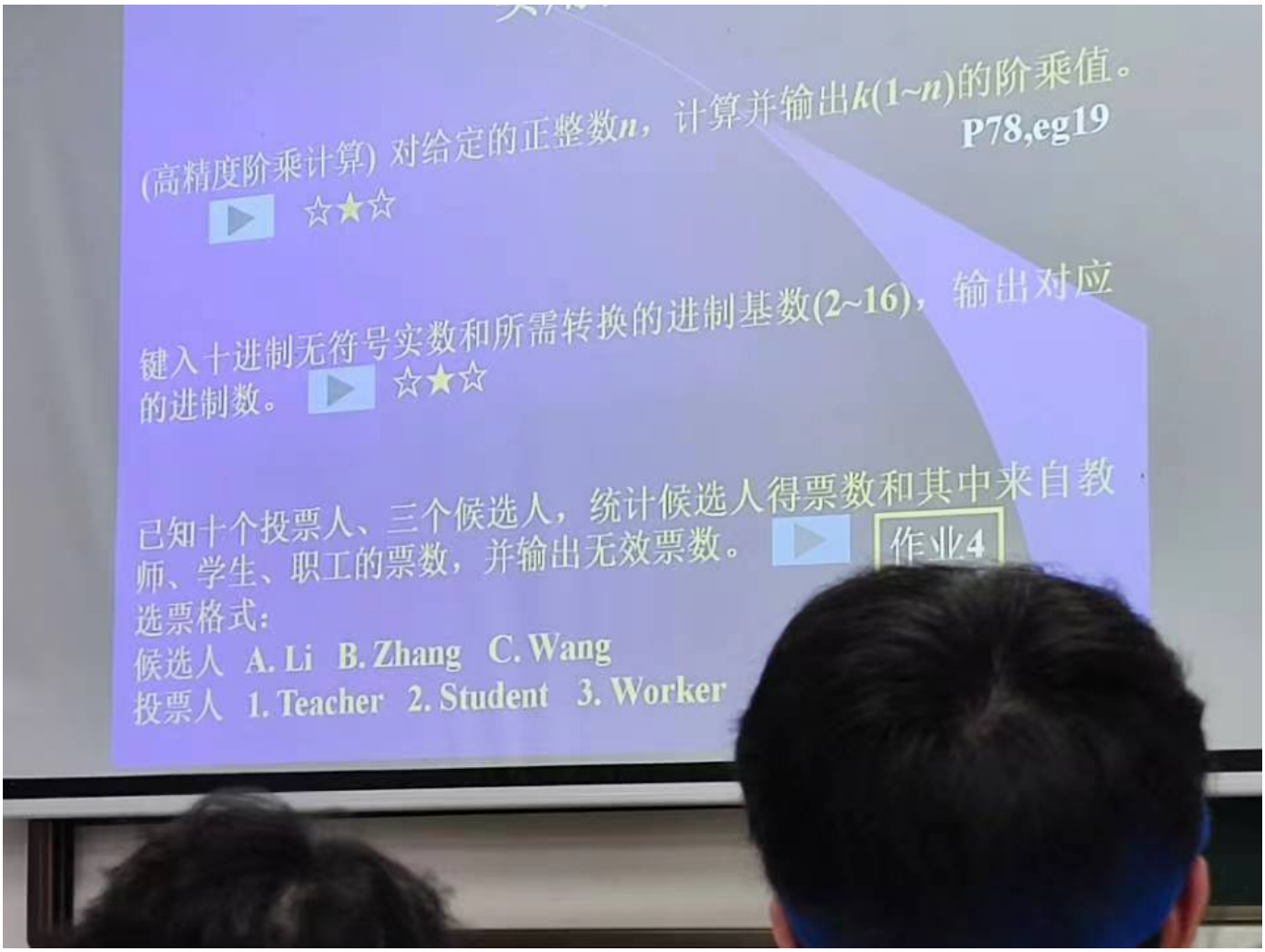
```
    int sum = 0;
    int start = 1;
    while (start < 100)
    {
      sum += start;
      start++;
      if (sum % 9 == 0)
      {
        cnt++;
      }
    }
    printf("%d", cnt);
  }
```




```c
#define N 30000
#include <stdio.h>
int main()
{
```

```
    int i, j;
    int sum = 0;
    long int s[N];
    for (i = 2; i < N; i++)
    {
      for (j = 1, s[i] = 0; j < i; j++)
      {
        if (i % j == 0)
        {
          s[i] += j;
        }
      }
    }
    for (i = 2; i < N; i++)
    {
      j = s[i];
      if (j > N)
        continue;
      if (i == s[j] && j > i)
      {
        printf("(%d,%d)\n", i, j);
        sum++;
      }
    }
    printf("%d", sum);
  }
```

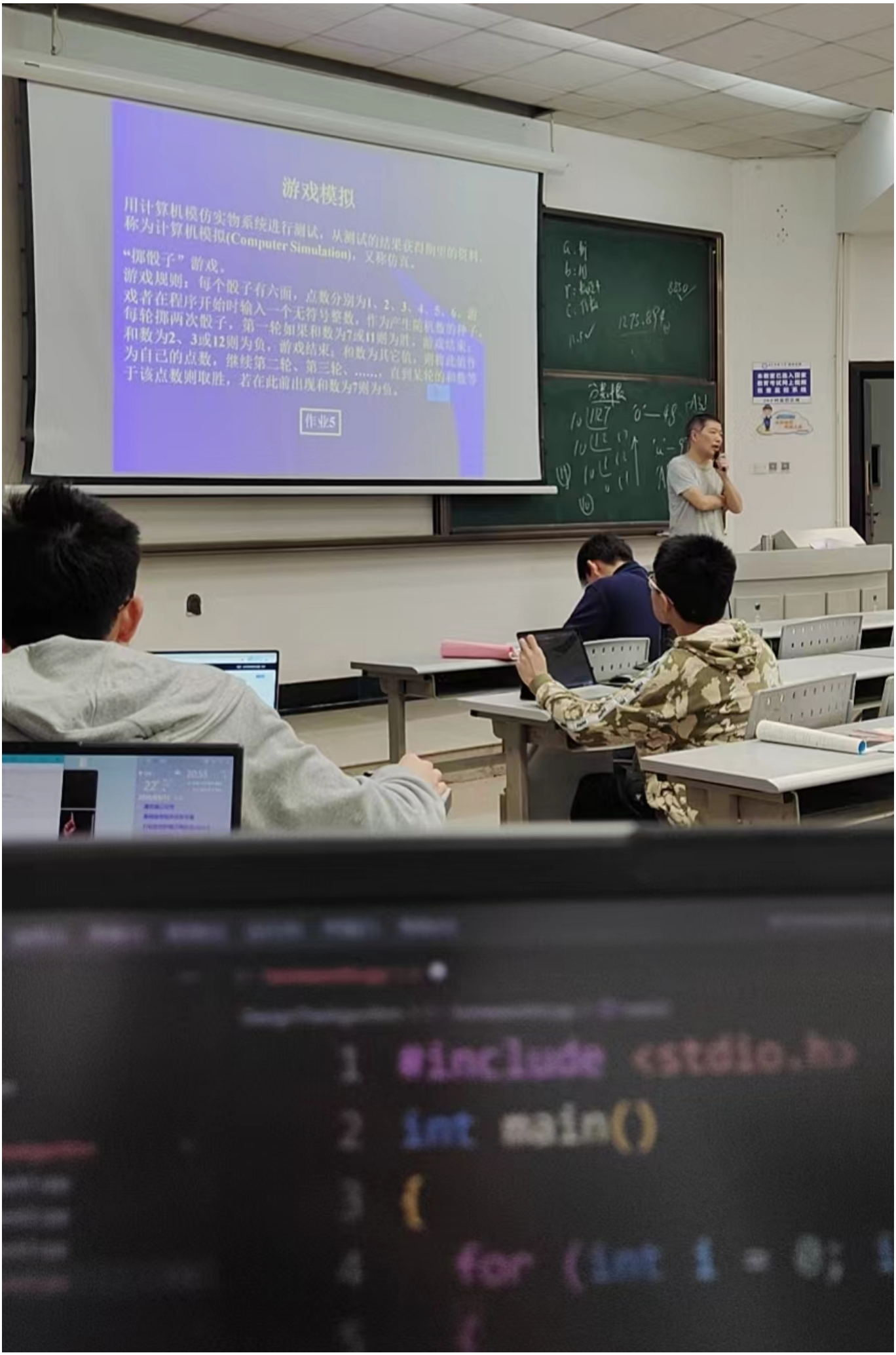
```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework3.cpp' -o 'homework3.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework3.exe' }
(220,284)
(1184,1210)
(2620,2924)
(5020,5564)
(6232,6368)
(10744,10856)
(12285,14595)
(17296,18416)
8
```

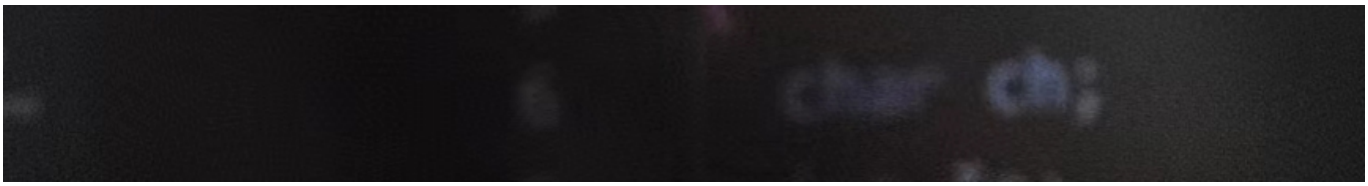



```c
#include <stdio.h>
void printResult(int ret[3][3], int nonsense)
{
  printf("A Teacher:%d Student:%d Worker:%d\n", ret[0][0], ret[0][1], ret[0][2]);
  printf("B Teacher:%d Student:%d Worker:%d\n", ret[1][0], ret[1][1], ret[1][2]);
  printf("C Teacher:%d Student:%d Worker:%d\n", ret[2][0], ret[2][1], ret[2][2]);
  printf("废票: %d\n", nonsense);
}
int main()
{
  int ret[3][3] = {0};
  char ch;
  int In = 0;
  int nonsense = 0;
  while (1)
  {
    scanf("%c", &ch);
    scanf("%d", &In);
    if (In > 3 || In <= 0)
    {
      nonsense++;
    }
    else if (In <= 3)
    {
      switch (ch)
```

```
        {
        case 'A':
          ret[0][In - 1]++;
          break;
        case 'B':
          ret[1][In - 1]++;
          break;
        case 'C':
          ret[2][In - 1]++;
          break;
        default:
          nonsense++;
          break;
        }
      }
      printResult(ret, nonsense);
      getchar();
    }
  }
```

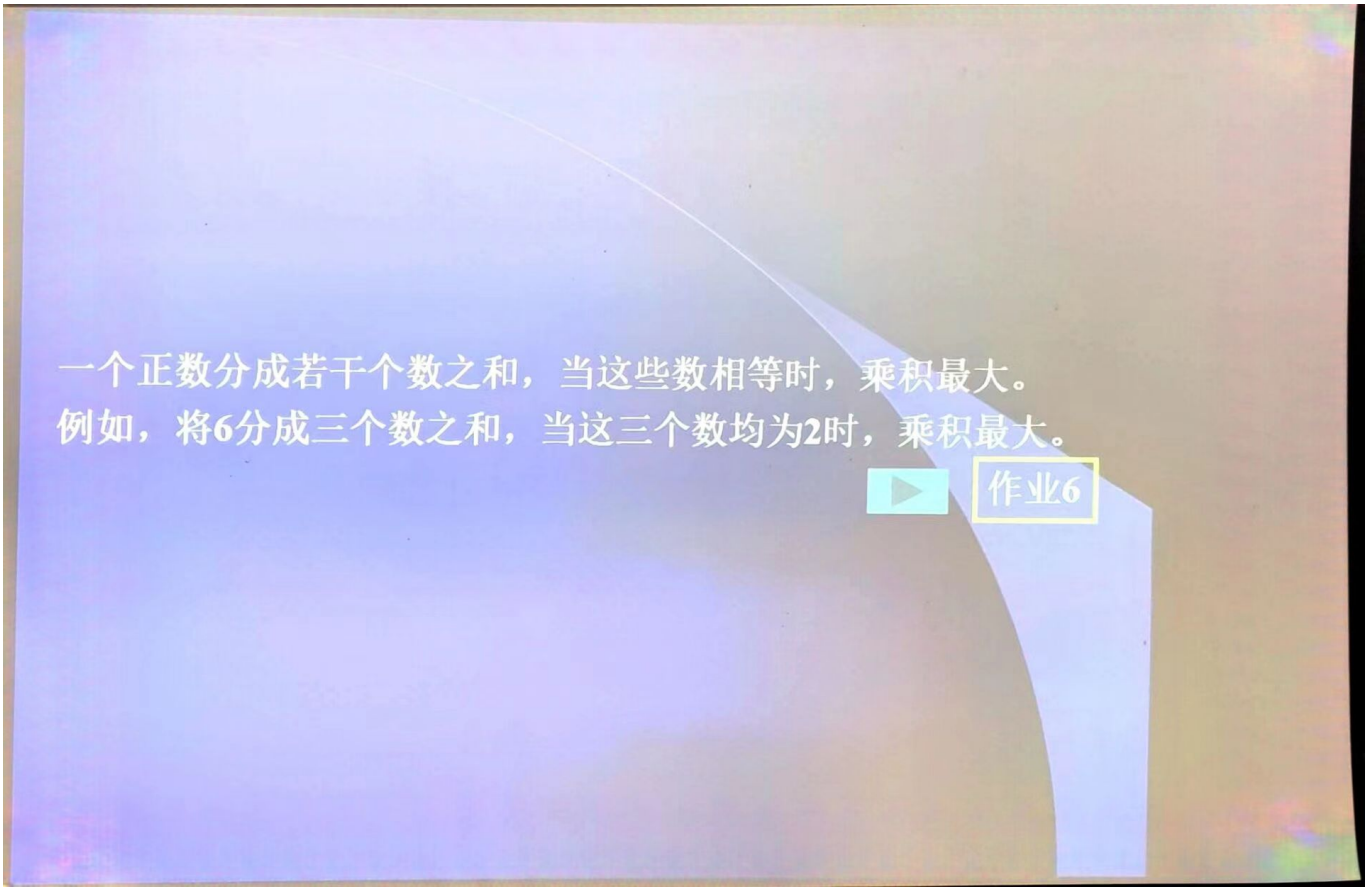
```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework4.cpp' -o 'homework4.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework4.exe' }
A2
A Teacher:0 Student:1 Worker:0
B Teacher:0 Student:0 Worker:0
C Teacher:0 Student:0 Worker:0
废票: 0
A3
A Teacher:0 Student:1 Worker:1
B Teacher:0 Student:0 Worker:0
C Teacher:0 Student:0 Worker:0
废票: 0
B3
A Teacher:0 Student:1 Worker:1
B Teacher:0 Student:0 Worker:1
C Teacher:0 Student:0 Worker:0
废票: 0
D55
A Teacher:0 Student:1 Worker:1
B Teacher:0 Student:0 Worker:1
C Teacher:0 Student:0 Worker:0
废票: 1
```


游戏模拟
用计算机模仿实物系统进行测试，从测试的结果获得所要的资料，称为计算机模拟(Computer Simulation)，又称仿真。
“掷骰子”游戏。
游戏规则：每个骰子有六面，点数分别为1、2、3、4、5、6。游戏者在程序开始时输入一个无符号整数，作为产生随机数的种子。每轮掷两次骰子，第一轮如果和数为7或11则为胜，游戏结束；和数为2、3或12则为负，游戏结束；和数为其它值，则将此值作为自己的点数，继续第二轮、第三轮、……，直到某轮的和数等于该点数则取胜，若在此前出现和数为7则为负。
作业5



```c
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char *argv[])
{
  int winC = 0;
  int loseC = 0;
  int aid = 0;
  srand((unsigned int)time(0));
  int first = rand() % 6 + 1;
  int second = rand() % 6 + 1;
  // printf("(%d,%d)", first, second);
  aid = first + second;
  if (aid == 7 || aid == 11)
  {
    printf("you win it\n");
    return 0;
  }
  else
  {
    if (aid == 2 || aid == 3 || aid == 12)
    {
      printf("you lose it\n");
      return 0;
    }
    else
    {
      while (1)
      {
        first = rand() % 6 + 1;
        second = rand() % 6 + 1;
        if (aid == first + second)
        {
          printf("you win it\n");
          break;
        }
        else if (first + second == 7)
        {
          printf("you lose it\n");
          break;
        }
      }
    }
  }
  return 0;
}
```

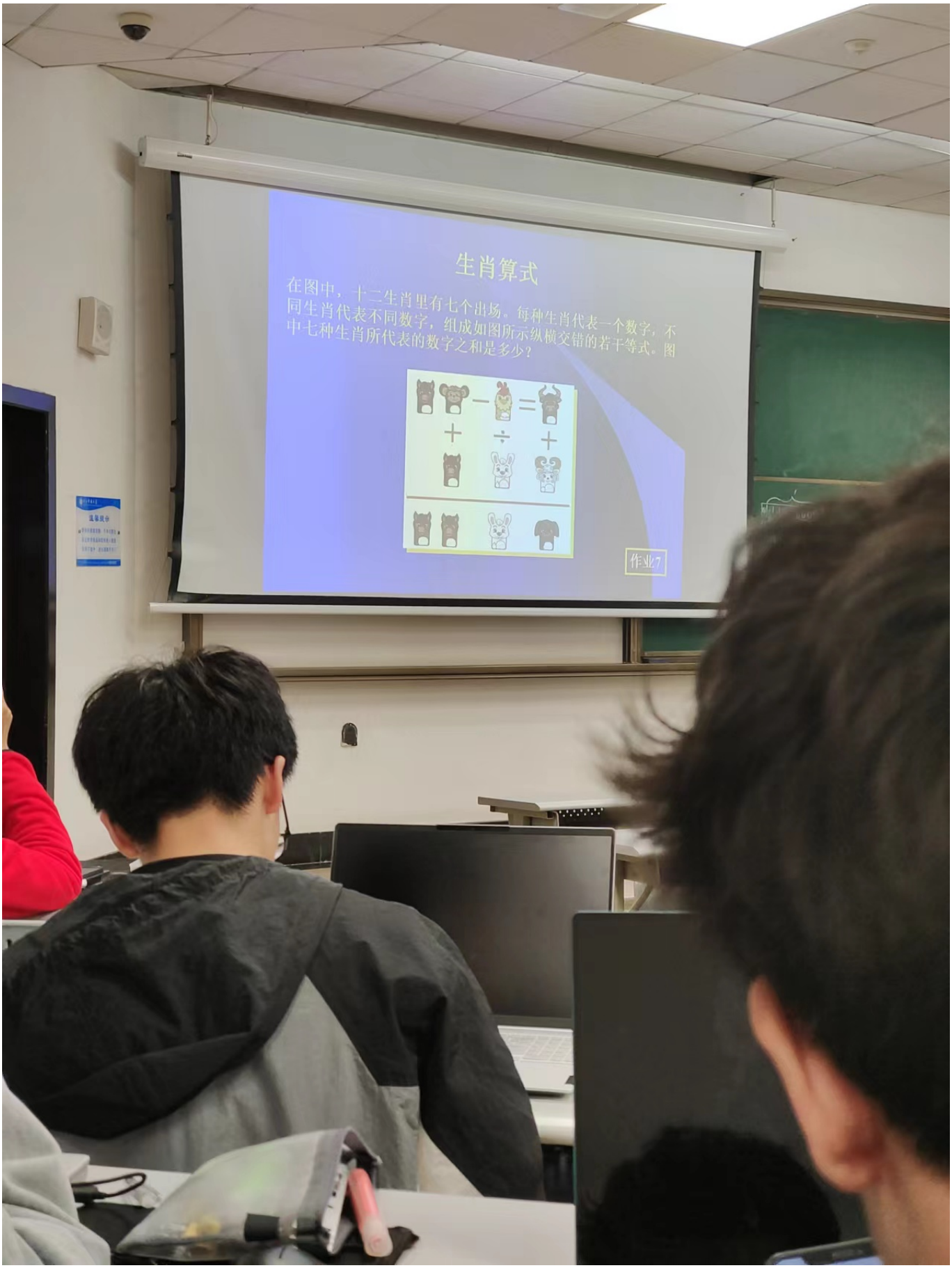
```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework5.cpp' -o 'homework5.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework5.exe' }
homework5.cpp: In function 'int main(int, char**)':
homework5.cpp:6:7: warning: unused variable 'winC' [-Wunused-variable]
   int winC = 0;
       ^~~~
homework5.cpp:7:7: warning: unused variable 'loseC' [-Wunused-variable]
   int loseC = 0;
       ^~~~~
you win it
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework5.cpp' -o 'homework5.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework5.exe' }
homework5.cpp: In function 'int main(int, char**)':
homework5.cpp:6:7: warning: unused variable 'winC' [-Wunused-variable]
   int winC = 0;
       ^~~~
homework5.cpp:7:7: warning: unused variable 'loseC' [-Wunused-variable]
   int loseC = 0;
       ^~~~~
you lose it
PS D:\homework\git\CodeC\DesignTheAlgorithm>
```




```c
#define MAX_NAME 4
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h>
int main()
{
  srand((unsigned int)(time(0)));
  for (int i = 0; i < 10; ++i)
  {
    double a = (double)rand();
    for (int i = 1; i <= MAX_NAME; ++i)
    {
      printf("%.f", a / MAX_NAME);
```

```
        if (i != MAX_NAME)
        {
          printf("*");
        }
        if (i == MAX_NAME)
        {
          printf("=");
          printf("%.f", pow(a / MAX_NAME, MAX_NAME));
        }
      }
      printf("\n");
      Sleep(100);
    }
  }
```

```
PS D:\homework\git\Codec\DesignTheAlgorithm> cd "d:\homework\git\Codec\DesignTheAlgorithm"
PS D:\homework\git\Codec\DesignTheAlgorithm> g++ 'homework6.cpp' -o 'homework6.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework6.exe' }
1254*1254*1254*1254=2472806570256
6504*6504*6504*6504=1789460557664256
2180*2180*2180*2180=22606033353690
6836*6836*6836*6836=2183457206081419
1364*1364*1364*1364=3458908347073
4790*4790*4790*4790=526431724810000
99*99*99*99=97033582
3340*3340*3340*3340=124447411360000
4220*4220*4220*4220=317289440170710
6775*6775*6775*6775=2106867375390625
PS D:\homework\git\Codec\DesignTheAlgorithm>
```

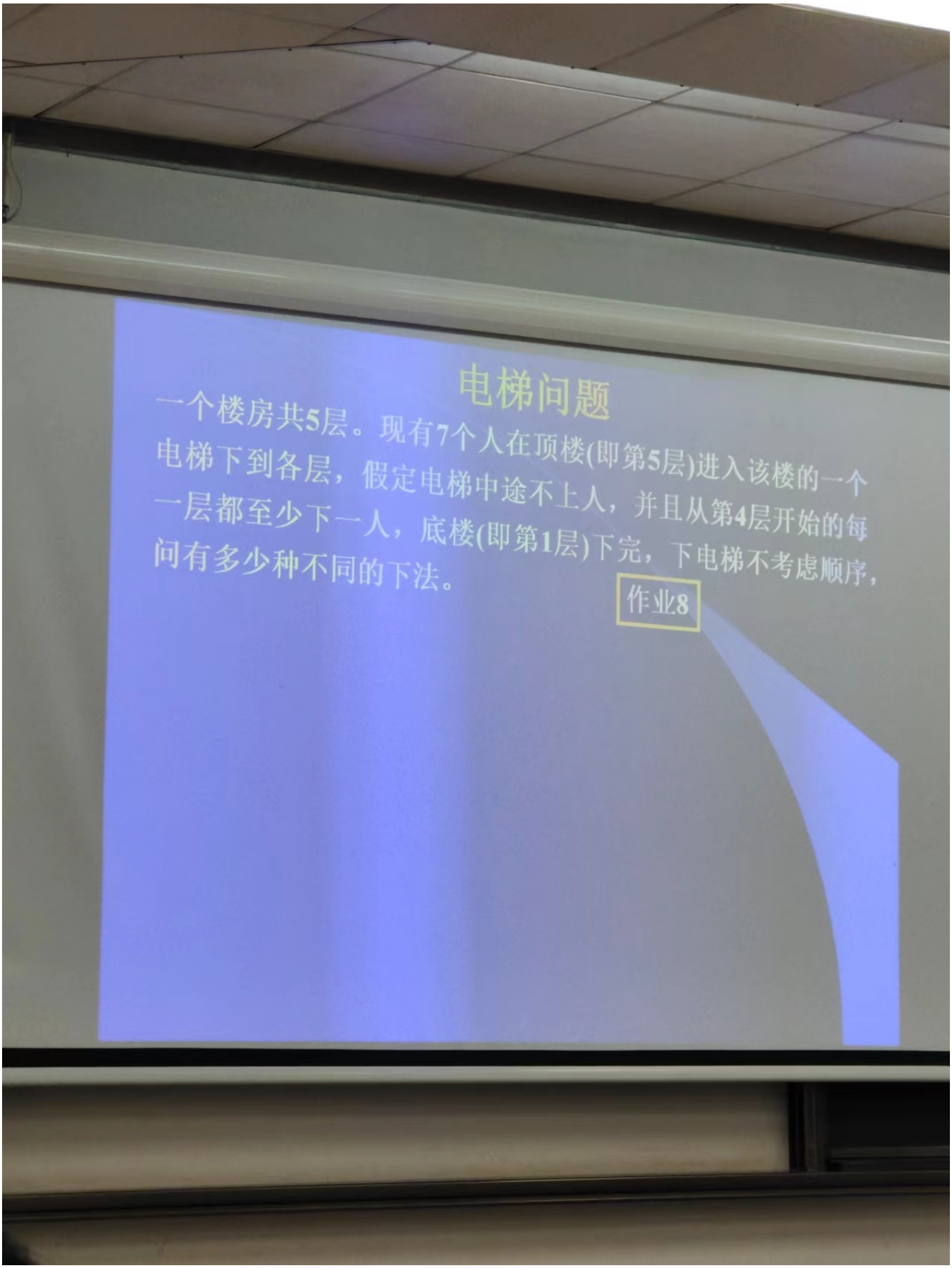



```
#include <stdio.h>
#include <stdlib.h>
int main()
{
```

```
  int niu;
  int hou;
  int ji;
  int tu;
  int yang;
  int gou;
  int zhu;
  for (niu = 0; niu < 10; ++niu)
    for (hou = 0; niu < 10; ++niu)
      for (ji = 0; ji < 10; ++ji)
        for (tu = 1; tu < 10; ++tu)
          for (yang = 0; yang < 10; ++yang)
            for (gou = 0; gou < 10; ++gou)
              for (zhu = 1; zhu < 10; ++zhu)
              {
                if ((10 * zhu + hou + zhu == zhu * 11) && (10 * zhu + hou - ji ==
niu) && (ji / tu == tu) && (niu + yang == gou))
                {
                  printf("已找到\n");

printf("niu=%d\nhou=%d\nji=%d\ntu=%d\nyang=%d\ngou=%d\nzhu=%d\n", niu, hou, ji,
tu, yang, gou, zhu);
                  return 0;
                }
              }
  printf("没找到");
  return -1;
}
```

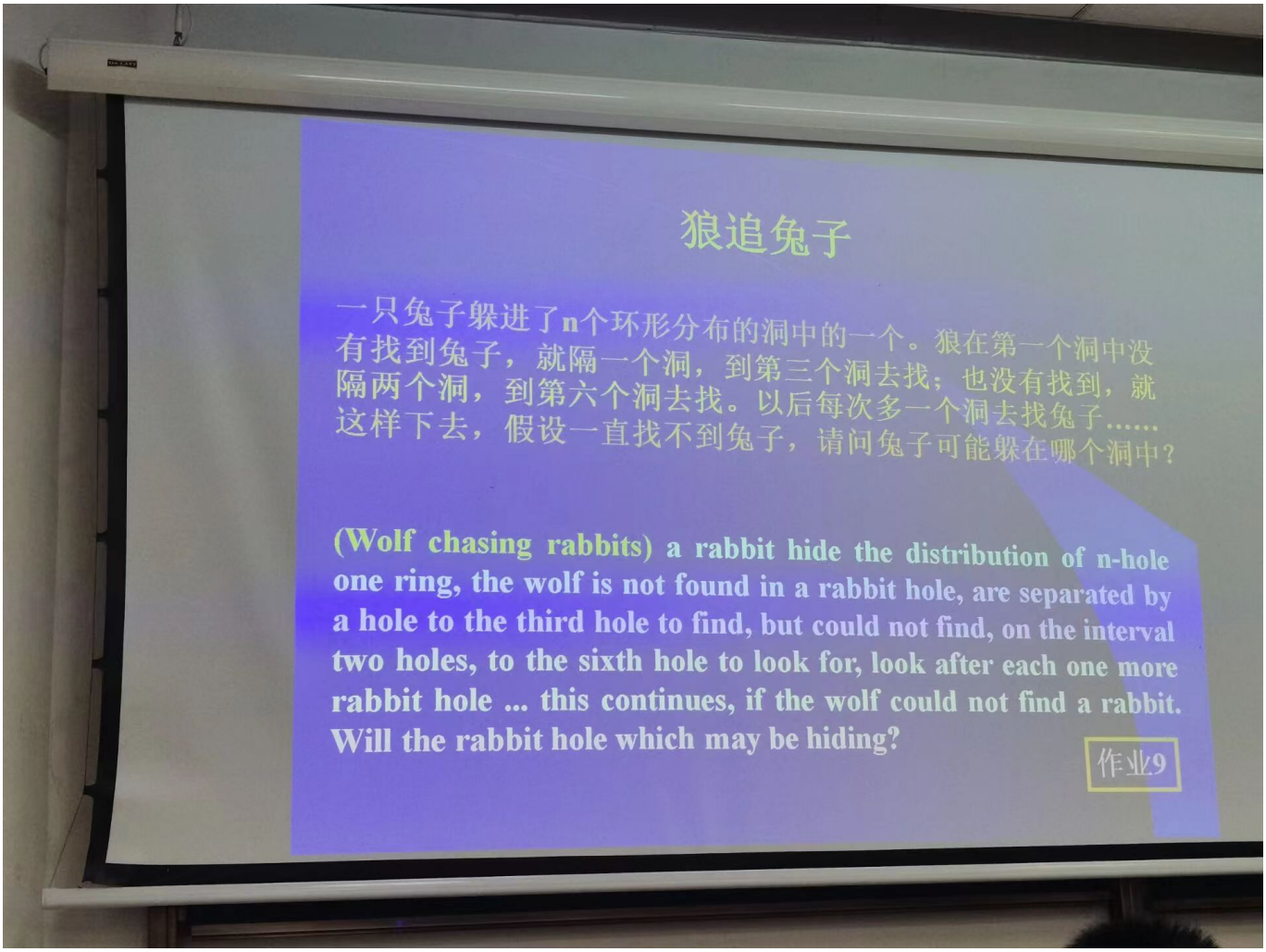
```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework7.cpp' -o 'homework7.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework7.exe' }
已找到
niu=1
hou=0
ji=9
tu=3
yang=0
gou=1
zhu=1
PS D:\homework\git\CodeC\DesignTheAlgorithm>
```




```
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
//其实就是问7个之间插三个板有多少可能，一共6个空。即C63*A77
```

```cpp
long int fact(int n)
{
  if (n == 1)
    return n;
  else if (n == 2)
    return n;
  else
    return n * fact(n - 1);
}
int main()
{
  int m = 6;
  int n = 3;
  int h = 7;
  std::cout << (fact(m) / (fact(m - n) * fact(n))) * fact(h);
}
```

```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework8.cpp' -o 'homework8.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework8.exe' }
100800
PS D:\homework\git\CodeC\DesignTheAlgorithm>
```




```cpp
#include <iostream>
using namespace std;
int full_ERROR(int n)
```

```
{
  if (n == 1)
    return 0;
  if (n == 2)
    return 1;
  return (n - 1) * (full_ERROR(n - 1) + full_ERROR(n - 2));
}
int main()
{
  int n, res;
  cin >> n;
  res = full_ERROR(n);
  cout << res;
  return 0;
}
```

```
PS D:\homework\git\CodeC\DesignTheAlgorithm> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework8.cpp' -o 'homework8.exe' -Wall -O2 -m64 -static-libgcc -std=c++14 -fexec-charset=GBK
; if ($?) { &'./homework8.exe' }
100800
PS D:\homework\git\CodeC\DesignTheAlgorithm>
```


```
#include <stdio.h>
#include <stdlib.h>
#define MAX 1000
#define N 10
```

```cpp
int main()
{
  int i;
  bool *pos = (bool *)malloc(sizeof(bool));
  for (int i = 0; i < N; ++i)
  {
    pos[i] = 0;
  }
  int a = 0;
  for (i = 0; i < MAX; i++)
  {
    pos[a] = true;
    a++;
    a += i;
    a = a % 10;
  }
  for (i = 0; i < N; i++)
    if (!pos[i])
      printf("兔子可能在第%d洞中\n", i + 1);
  return 0;
}
```

```
PS D:\homework\git\CodeC> cd "d:\homework\git\CodeC\DesignTheAlgorithm"
PS D:\homework\git\CodeC\DesignTheAlgorithm> g++ 'homework10.cpp' -o 'homework10.exe' -Wall -O2 -m64 -static-libgcc -st
d=c++14 -fexec-charset=GBK ; if ($?) { &'./homework10.exe' }
兔子可能在第3洞中
兔子可能在第5洞中
兔子可能在第8洞中
兔子可能在第10洞中
PS D:\homework\git\CodeC\DesignTheAlgorithm>
```