


# LZU 计算物理 I 2024 期中

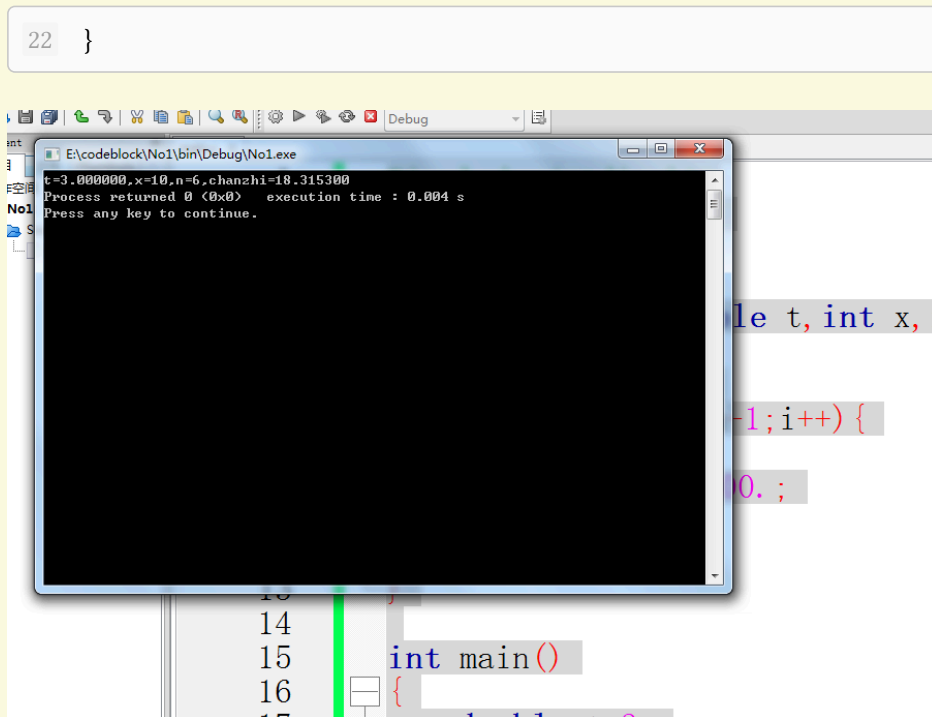
本文档由“REKPARC-SPST-LZU-UHT” 项目组编写，如你也有相关笔记或资料，欢迎参与贡献！（点击上方 [github](#) 图标可跳转链接） 或者可以邮件传递你的资料：“[lzuanonymous@qq.com](mailto:lzuanonymous@qq.com)”

## REKPARC-SPST-LZU-UHT 项目组

### 第一题 .

(编程, 100.0 分) 1. a) 某工厂一月份产值为  $t$  万元，以后的每一个月都比上一个月增产  $x\%$ ，编写函数计算前  $n$  个月的总产值。b) 调用该函数计算当  $t = 3, x = 10, n = 6$  情况下的总产值。如果没有特别说明，计算机语言为 C 语言，将程序源代码（注意源代码上传，不是截图上传）及运行结果截图请粘贴在答案区。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  double chanzhi(double t,int x,int n)
6  {
7      double sum=0;
8      double temp=t;
9      for(int i=0;i<n-1;i++){
10         sum+=temp;
11         temp*=1+x/100.;
12     }
13     return sum;
14 }
15
16 int main()
17 {
18     double t=3;
19     int x=10,n=6;
20     printf("t=%lf,x=%d,n=%d,chanzhi=%lf",t,x,n,chanzhi(t,x,n));
21     return 0;
```



score:80 (扣分原因未知)

### 第二题 .

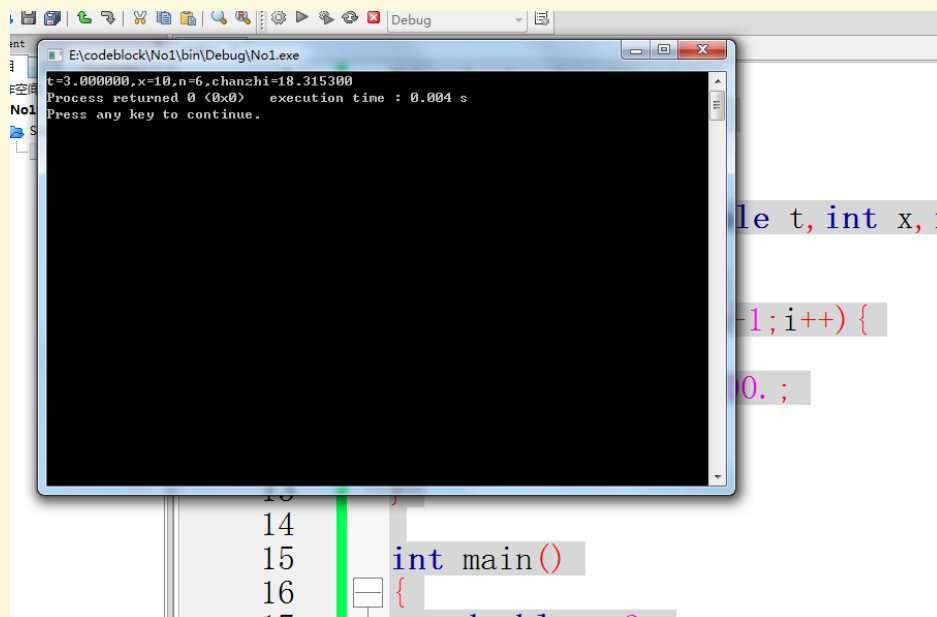
(编程, 100.0 分) 2. 已知面包价格  $x$  元/个，牛奶是  $y$  元/袋。小明的妈妈给了小明  $m$  元钱，编写函数计算并输出小明能买到的牛奶和面包的数量的所有可能的组合。这里假设  $x, y, m$  都是正整数。如果没有特别说明，计算机语言为 C 语言，将程序源代码（注意源代码上传，不是截图上传）及运行结果截图请粘贴在答案区。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  double chanzhi(double t,int x,int n)
6  {
```

```

7  double sum=0;
8  double temp=t;
9  for(int i=0;i<n-1;i++){
10     sum+=temp;
11     temp*=1+x/100.;
12 }
13 return sum;
14 }
15
16 int main()
17 {
18     double t=3;
19     int x=10,n=6;
20     printf("t=%lf,x=%d,n=%d,chanzhi=%lf",t,x,n,chanzhi(t,x,n));
21     return 0;
22 }

```



score:95.0

### 第三题 .

编写函数计算两正整数的所有公因数的乘积并返回该值。

如果没有特别说明，计算机语言为 C 语言，将程序源代码（注意源代码上传，不是截图上传）及运行结果截图请粘贴在答案区。

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int min(int x,int y)
6  {
7      return x<y?x:y;
8  }
9
10 int mutifactor(int x,int y){
11     int muty=1;
12     for(int i=1;i<=min(x,y);i++){
13         if(x%i==0 && y%i==0){
14             muty=muty*i;
15         }
16     }
17     return muty;
18 }
19
20
21
22 int main()
23 {
24     int x,y;
25     scanf("%d%d",&x,&y);
26     printf("%d",mutifactor(x,y));
27     return 0;
28 }

```

score:100.0

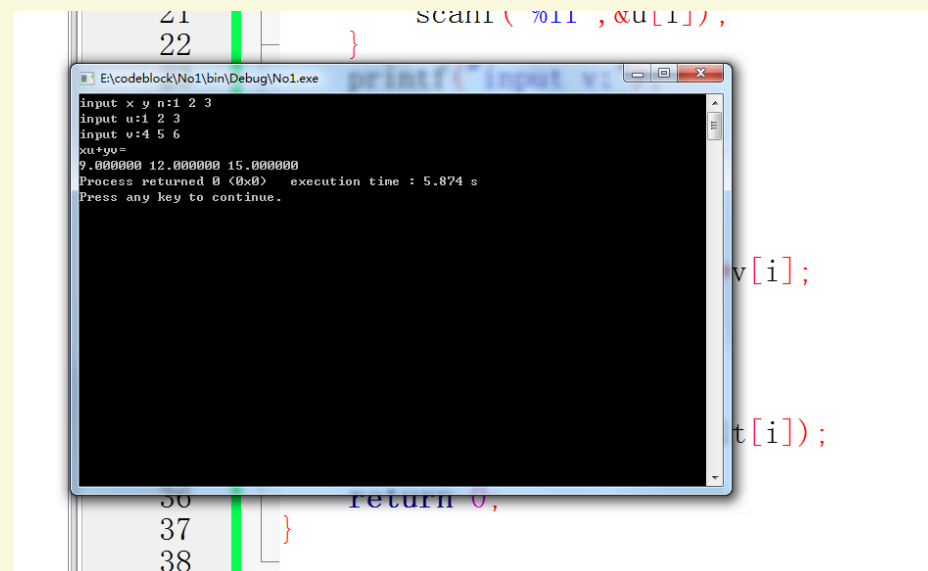
### 第四题 .

(编程, 100.0 分) 4. 已知中两向量  $u$ ,  $v$  和标量  $x, y$ , 计算  $W = xu + yv$  并输出到屏幕, 自行设定标量以及向量中元素的值。如果没有特别说明, 计算机语言为 C 语言, 将程序源代码 (注意源代码上传, 不是截图上传) 及运行结果截图请粘贴在答案区。

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int main()
6  {   int x,y;
7      int n;
8      printf("input x y n:");
9      scanf("%d %d %d",&x,&y,&n);
10     double u[10];
11     double v[10];
12     for(int i=0;i<n;i++){
13         u[i]=0;
14         v[i]=0;
15     }
16     printf("input u:");
17     for(int i=0;i<n;i++){
18         scanf("%lf",&u[i]);
19     }
20     printf("input v:");
21     for(int i=0;i<n;i++){
22         scanf("%lf",&v[i]);
23     }
24     double result[10];
25     for(int i=0;i<n;i++){
26         result[i]=x*u[i]+y*v[i];
27     }
28     printf("xu+yv=\n");
29
30     for(int i=0;i<n;i++){
31         printf("%lf ",result[i]);
32     }
33     return 0;
34 }

```



## 第五题 .

编写函数返回组成某一正整数的所有数字之和, 如: 组成 1056 的数字和为 12。如果没有特别说明, 计算机语言为 C 语言, 将程序源代码 (注意源代码上传, 不是截图上传) 及运行结果截图请粘贴在答案区。

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4
5  int sumeach(int x){
6      int sum=0;
7      int temp=x;
8      for(int i=0;x>0;i++){
9          temp=x%10;//当前个位数
10         sum+=temp;
11         x=x/10;
12     }
13     return sum;
14 }

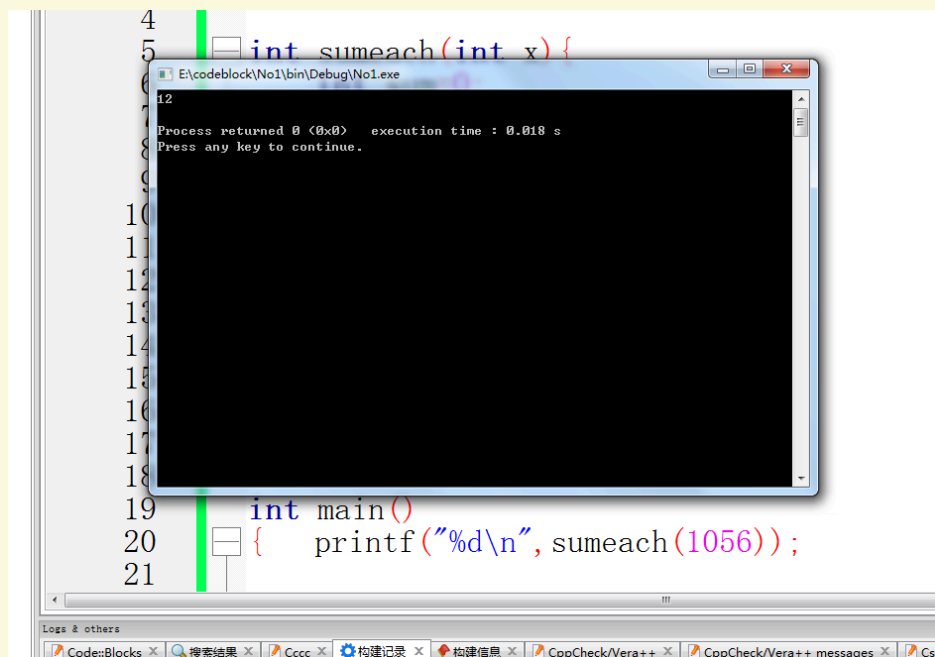
```

```

15
16 int main()
17 {
18     printf("%d\n",sumeach(1056));
19     return 0;
20 }

```

score:100.0



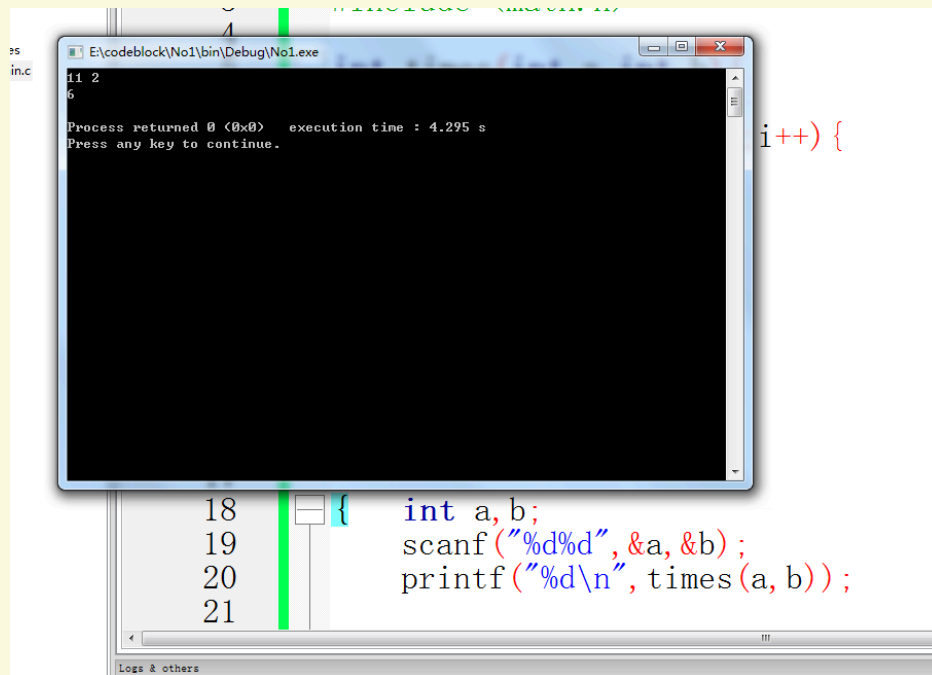
```

1  #include <stdio.h>
2
3  #include <stdlib.h>
4
5  #include <math.h>
6
7
8
9  int times(int a,int b){
10     int sum=0;
11     for(int i=0;i<a/b+2;i++){
12         if(b*i<a){
13             sum++;
14         }
15     }
16     return sum;
17 }
18 int main()
19 {
20     int a,b;
21     scanf("%d%d",&a,&b);
22     printf("%d\n",times(a,b));
23     return 0;
24 }
25

```

## 第六题 .

(编程, 100.0 分) 6. 一口水井深  $a$  米, 一只青蛙在井底, 一次向上跳  $b$  米, 编写函数计算青蛙至少需要跳多少次才能跳出水井。这里假设  $a, b$  都是正整数。如果没有特别说明, 计算机语言为 C 语言, 将程序源代码 (注意源代码上传, 不是截图上传) 及运行结果截图请粘贴在答案区。



```

11 2
6
Process returned 0 (0x0) execution time : 4.295 s
Press any key to continue.

18 { int a, b;
19   scanf("%d%d", &a, &b);
20   printf("%d\n", times(a, b));
21 }

```

score:100.0

## 第七题 .

从键盘读入四个 0-9 之间的数字，输出这四个数字能组成的最大十进制数，比如：读入 0, 3, 1, 9，输出 9310。

如果没有特别说明，计算机语言为 C 语言，将程序源代码（注意源代码上传，不是截图上传）及运行结果截图请粘贴在答案区。

### ▲ 注意

### 注意 7.0.1

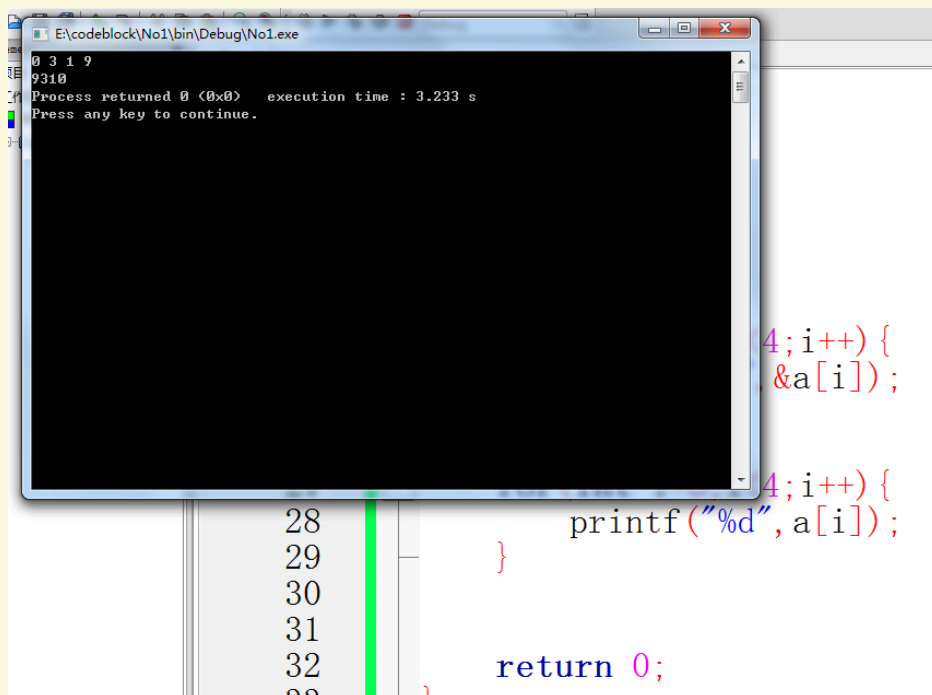
请注意，此题中的输出必须是以整型的形式输出，你不能逐个输出字符串，虽然这很奇怪，但是 zjz 就是这样要求的

```

2  #include <stdlib.h>
3  #include <math.h>
4  void sort(int a[],int n){
5      for(int i=0;i<n;i++){
6          for(int j=i;j<n;j++){
7              if(a[j]>a[i]){
8                  int temp=a[j];
9                  a[j]=a[i];
10                 a[i]=temp;
11             }
12         }
13     }
14 }
15 int main()
16 {
17     int a[4];
18     for(int i=0;i<4;i++){
19         scanf("%d",&a[i]);
20     }
21     sort(a,4);
22     for(int i=0;i<4;i++){
23         printf("%d",a[i]);
24     }
25     return 0;
26 }

```

```
1  #include <stdio.h>
```



```

E:\codeblock\No1\bin\Debug\No1.exe
0 3 1 9
9310
Process returned 0 (0x0) execution time : 3.233 s
Press any key to continue.

```

score:70.0

## 第八题 .

已知矩阵 B 是 4 X 4 的矩阵，写程序利用初等行变换将矩阵 B 化为阶梯型矩阵，矩阵 B 中元素的值用 `rand()%4` 产生，并检查你的计算结果是否正确。

如果没有特别说明，计算机语言为 C 语言，将程序源代码（注意源代码上传，不是截图上传）及运行结果截图请粘贴在答案区。

```

1  #include <math.h>
2  #include <stdio.h>
3  #include <stdlib.h>
4  int min(int a, int b) { return a < b ? a : b; }
5  //交换
6  void swap(double* a, double* b) {

```

```

7  double temp = *a;
8  *a = *b;
9  *b = temp;
10 }
11 //给出第 i 列绝对值最大的位置
12 int placemax(double a[], int n) {
13     double b = a[0];
14     int q = 0;
15     for (int i = 0; i < n; i++) {
16         if (fabs(a[i]) > fabs(b)) {
17             b = a[i];
18             q = i;
19         }
20     }
21     return q;
22 }
23
24 int main() {
25     int m, n;
26     scanf("%d%d", &m, &n);
27     double del = pow(10, -6);
28     double **matrix = malloc(m * sizeof(double*));
29     for (int i = 0; i < m; i++) {
30         matrix[i] = malloc(n * sizeof(double));
31     }
32     for (int i = 0; i < m; i++) {
33         for (int j = 0; j < n; j++) {
34             matrix[i][j] = rand()%4;
35         }
36     }
37     for(int i=0;i<m;i++,puts("")){
38         for(int j=0;j<n;j++){
39             printf("%lf",matrix[i][j]);
40         }
41     }
42     printf("\n");
43     //将矩阵化为阶梯型矩阵
44     for (int i = 0; i < min(m, n); i++) {
45         int skip = 0;
46         double* coltemp = malloc(m * sizeof(double));
47         for (int j = 0; j < i; j++)
48             coltemp[j] = 0;

```

```

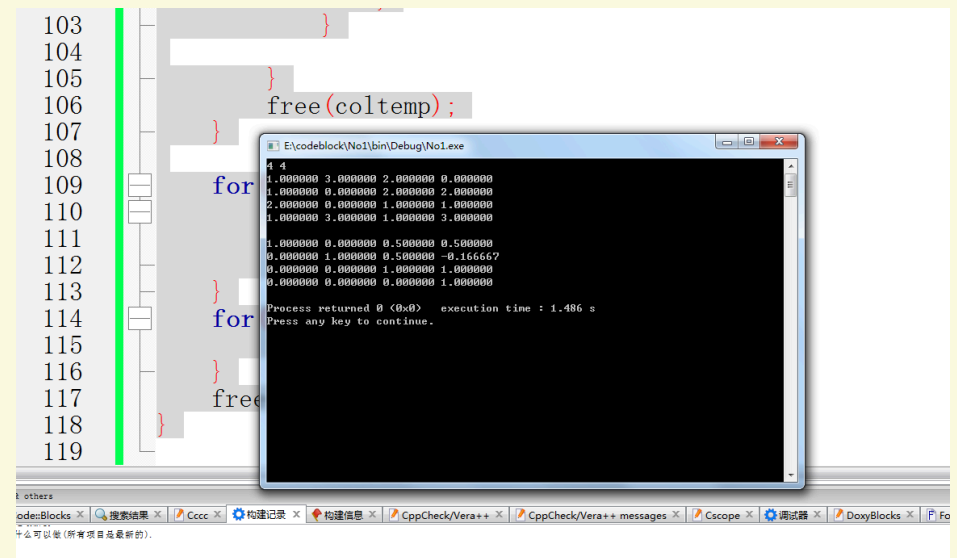
49 //第 i 列的主元列以下
50 for (int j = i; j < m; j++) {
51     coltemp[j] = matrix[j][i];
52 }
53 skip=1;
54 for (int j=i;j<m;j++) {
55     if(fabs(coltemp[j])>del){
56         skip=0;
57         break;
58     }
59 }
60 if(skip==1){
61     break;
62 }
63 //将第 i 列中绝对值最大的行交换到 i,i 位置上
64 int placetemp = placemax(coltemp, m);
65 for (int j = 0; j < n; j++) {
66     swap(&matrix[placetemp][j], &matrix[i][j]);
67 }
68 //第 i 行全除以主元列
69 for (int j = n - 1; j >= 0; j--) {
70     if (fabs(matrix[i][i]) > del) {
71         matrix[i][j] = matrix[i][j] / (matrix[i][i]);
72     }
73     else {
74         break;
75     }
76 }
77 /*
78 ai,i ..... ai,i+k ... ai,n
79 ...
80 ai+j,i ....ai+j,i+k ... ai+j,n
81 ...
82 am,i
83 */
84 if (skip == 0) {
85     for (int j = m - i - 1; j >= 1; j--) {
86         for (int k = n - i - 1; k >= 0; k--) {
87             matrix[i+j][i+k] -= matrix[i][i+k] * matrix[i+j][i];
88         }
89     }
90 }

```

```

91     free(coltemp);
92 }
93 for(int i=0;i<m;i++,puts("")){
94     for(int j=0;j<n;j++){
95         printf("%lf ",matrix[i][j]);
96     }
97 }
98 for(int i=0;i<m;i++){
99     free(matrix[i]);
100 }
101 free(matrix);
102 }

```



score:70.0 (rand 的种子需要设置)