

CSC 840

Matthew Wishoff

3/26/16

Project 4

# Table of Contents

1.0	Abstract .....	3
2.0	Introduction .....	3
3.0	Method and software used.....	4
4.0	Graphs and tables .....	4
5.0	Analysis .....	6
6.0	Conclusions .....	7
7.0	Appendix.....	8
a.)	BM1BMProgram4 .....	8
b.)	BM1BMProgram6 .....	16
c.)	BM1BMProgram8.....	26
d.)	BM1BMProgram12 .....	40
e.)	Matrix Multiplication Preprocessor .....	61
f.)	Matrix Multiplication Standard.....	65

## 1.0 Abstract

Taking a look cyclomatic complexity which is related to PLOC and LLOC studies, we are going to see if there is indeed a relationship. Over the years there have been many papers written over which formula is right, and which formula is wrong. For the purposes of this paper we will be using the formula of  $V(g) = C + 1$ . Or in greater length the formula is also equivalent to  $V(g) = e - n + p + 1$ . I hope to find a relationship between LLOC and cyclomatic complexity, so that maybe cyclomatic complexity can be approximated using LLOC or PLOC.

## 2.0 Introduction

As stated in the abstract we will be using the formula (1)  $V(g) = C + 1$ , or (2)  $V(g) = e - n + p + 1$  in this paper. Why these formulas and not the other ones? Well with formula two it accounts for all paths, so that you have a complete graph which is the purpose for the +1 at the end of the equation. It also accounts for all functions which is important as many programs are broken up into modules.

Now what is cyclomatic complexity? Well it is closely related to cognitive complexity. What is cognitive complexity? Cognitive complexity is how hard it is to understand something. So to make it simple if a program has a very high cyclomatic complexity it should be very hard to understand. Humans have been said to only have 5 registers in their head to keep track of things. This has been tested using salinity in water, and having people try to gauge how salty it is. People were only able to gauge up to about 5 differences accurately. This has been extended to say you can only keep track of 5 things in your head at one time. This is important because if you make your program so cyclomaticly complex nobody but you would be able to understand it.

What to look for in calculating the cyclomatic complexity is one of the easier tasks after having written an LLOC and PLOC program. What needs to be added is the ability to increment the cyclomatic complexity for: while, if for, &&, ||, and : (inside switch cases). With this in mind we just add some extra if statements to our already existing PLOC/LLOC program. And now our cyclomatic complexity program is all set and ready to go!

### 3.0 Method and software used

To develop the program used to compute cyclomatic complexity I used Dev-C++. To Analyze the data in graphs I used Matlab, and Excel to format the data in a presentable way. Since I am working on a new laptop (the previous one I broke the hard drive), I don't have past homework assignments. Therefore, for this project I am going to be using the bench maker programs to run my cyclomatic complexity program on. This has pros and cons affiliated with it though. For example it is a computer generated program not written by a human. However, one of the benefits is that the LLOC can be strictly controlled so that you can increase LLOC, and PLOC and see if it effects the complexity in a controlled manner. As a bonus I am also going to run the cyclomatic complexity program on my preprocessor heavy matrix multiplication program, as well as the .i file for the same program to see how the expanded macros change the cyclomatic complexity of it.

### 4.0 Graphs and tables

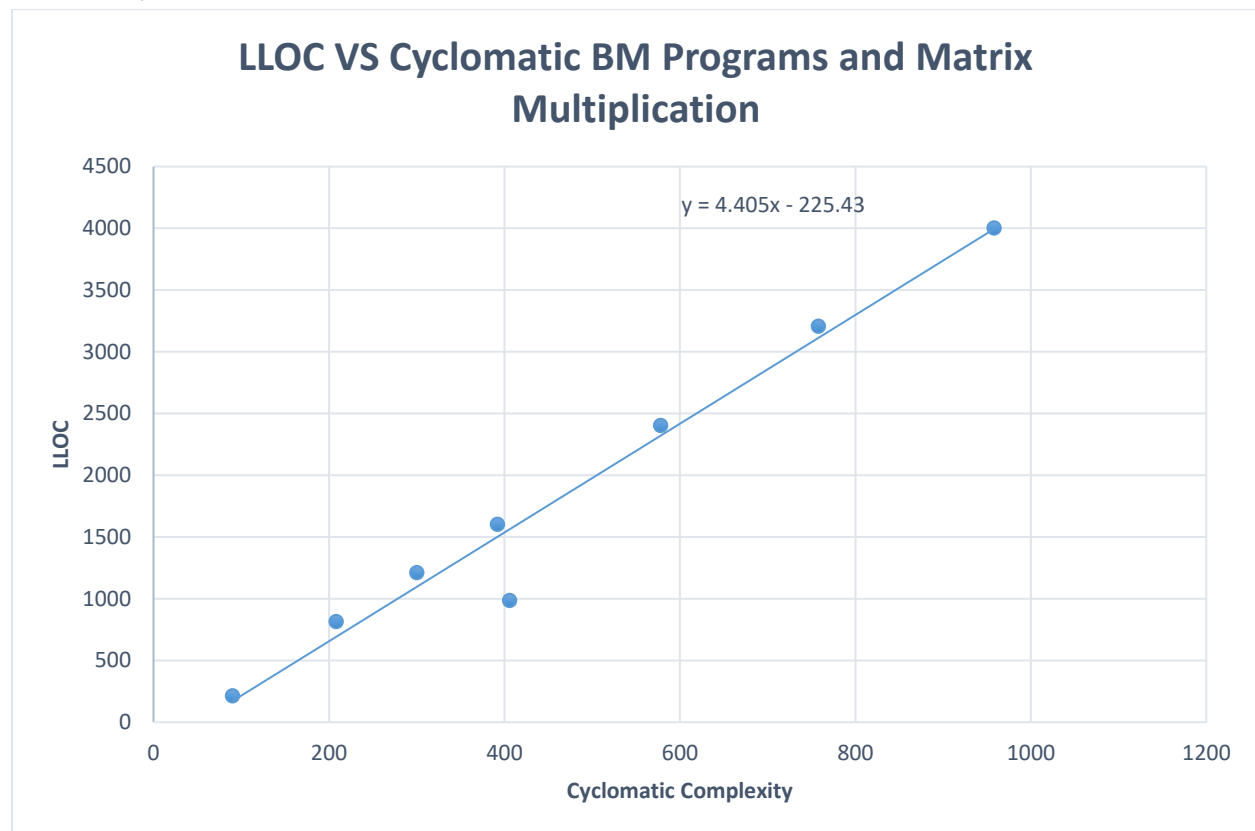


Figure 1

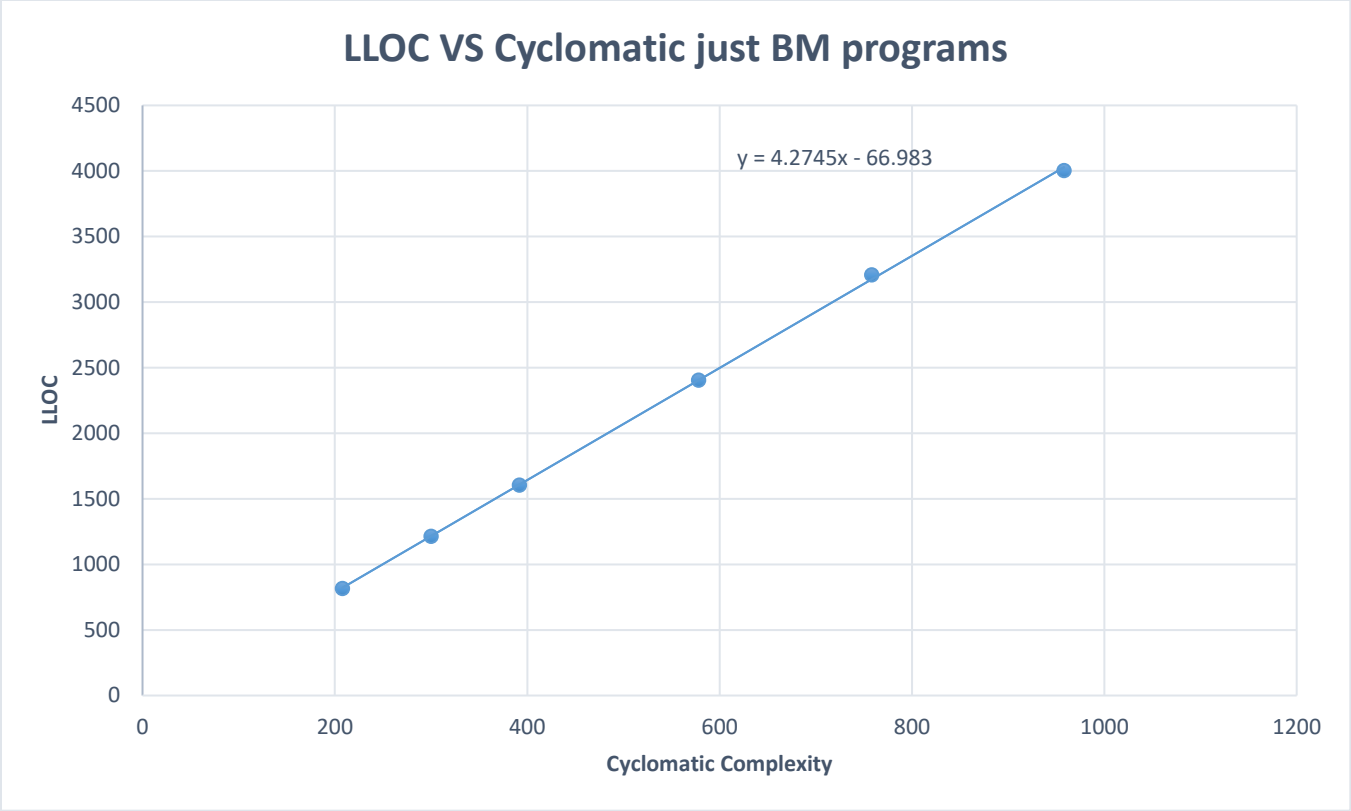


Figure 2

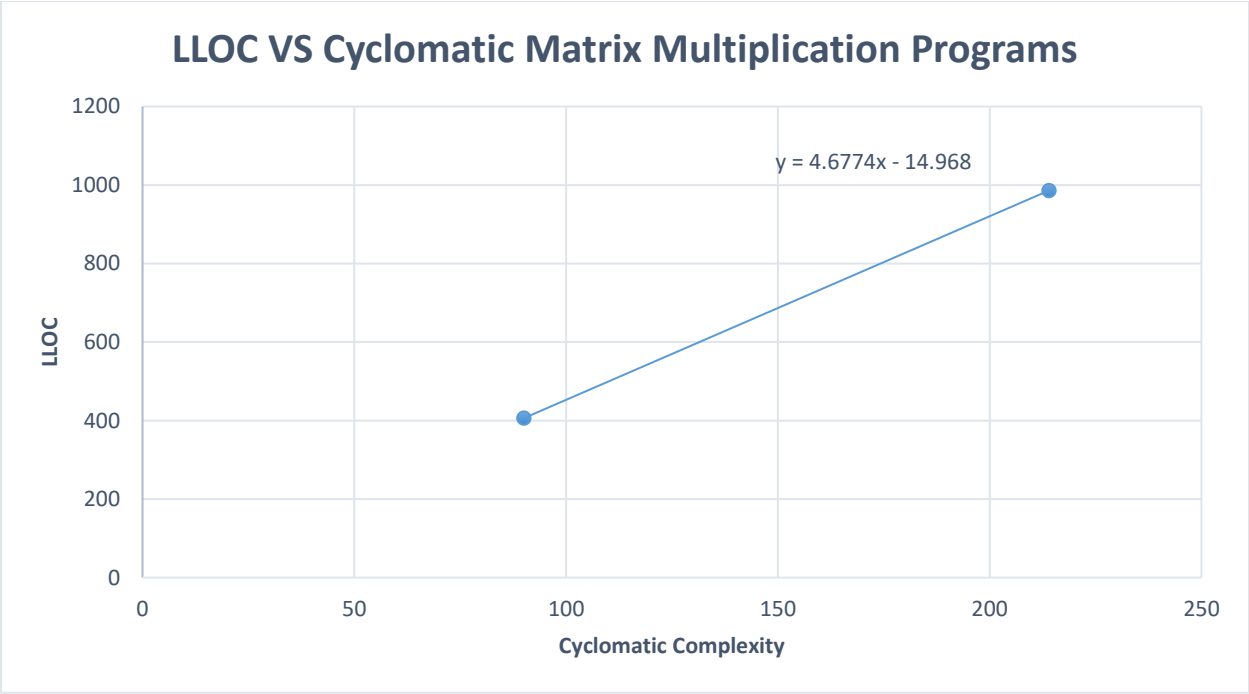


Figure 3

Program Name	Cyclomatic Complexity	LLOC Count
BM1BMProgram4.cpp	208	817
BM1BMProgram6.cpp	300	1213
BM1BMProgram8.cpp	392	1605
BM1BMProgram12.cpp	578	205
BM1BMProgram16.cpp	758	3208
BM1BMProgram20.cpp	958	4003
MatrixMultiplicationProject2.cpp	90	215
MatrixMultiplicationStandard.cpp	406	986

Figure 4

## 5.0 Analysis

Looking at figure one we can see that there is a very high correlation between LLOC and Cyclomatic complexity. Why might this be? Well for LLOC we count for, while, if, switch, and semi colon statements. Most of these are also counted for Cyclomatic complexity too! However, we do see one point that has fallen way short of the line but I think that can be expected from some programs. Some programs may not be as complex but still have a high number of LLOC due to possibly more arithmetic expressions terminated with semi-colons.

In figure two I took out the matrix multiplication programs, but kept the bench mark programs in. I think some conclusion that we can draw from this are when programs are written in a very systematic way we can expect a high correlation between LLOC and cyclomatic complexity. For example if we take 8 random programs written in 8 different styles, there could be no correlation between programs. This can happen based on what programs due. For example a program with no structures and just arithamtic expressions has a Cyclomatic complexity of 0, but can have a very high LLOC. Programs like these would throw off the data set. But if all programs have a balanced distribution of structures, and statements terminated with semi-colons we can see a very high correlation in figure 2.

On to figure 3 we can see that the program that doesn't use the preprocessor is far more complex than the program that does use the preprocessor. This is yet another reason to use the preprocessor over just brute forcing it with the code. The preprocessor way is also far better than having to recompile and run the program 8 separate times as well, since you will be letting the computer do all the work instead of you.

## 6.0 Conclusion

To answer the question: Is it possible to claim that the cyclomatic complexity depends only on LLOC? Yes and no. In the general case where there is a even distribution of control structures, and operations yes it is highly correlated. In the other case of when it is not evenly distributed such as having a high number of semi colons, and no control structures you would get a very skewed answer. Since LLOC depends also on semi-colons you would calculate a very high LLOC, but when calculating cyclomatic complexity you would get zero. Why is this? Because for cyclomatic complexity we do not care about the amount of semi-colons. So in this case cyclomatic if we graphed Cyclomatic complexity and LLOC we would get something that is an extreme outlier to the rest of the data.

So can this be modeled using LLOC every single time without fail? I would have to disagree, and say that the answer to that is no. However, can LLOC be used to have a correlation to cyclomatic complexity for well structured balanced programs yes! For example a company with a hard coding standard you could probably have a LLOC and Cyclomatic complexity that depends on it.

## 7.0 Appendix

### a.) BM1BMProgram4

```
#include <iostream>
using namespace std;

#include <time.h>

int
IFcnt[19],IFEcnt[25],SWcnt[12],WHILEcnt[25],DOcnt[25],FORcnt[38];

int F1(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[3]%5 ; )
    {
        if( ++IFEcnt[2]%2 )
        {
            m += (f+n*j-a*k)%100;
            a -= (l-c*e+l+a*b+c-c-i-i)%100;
            m -= (e-m)%100;
            j -= (a-h-n+n*c+b+j-g)%100;
            i += (n*b+e-f+f*m-e+j-l+m+l*n*f-
d)%100;
            f -= (n-e*f)%100;
        }
        else
        {
            while( ++WHILEcnt[1]%5 )
            {
                do
                {
                    if( ++IFcnt[0]%10 )
                    {
                        l -= (n-l+m*j+g*b-d)%100;
                        a += (e*b+l*j-j-k+a+n-f+a-
b+l)%100;
                        m += (j*j-g-l*i+h+f*d-j+n-
j)%100;
                    }
                    f -= (f+m*l-a-e*i+b-l+i-n*n-
l)%100;
                    for( ; ++FORcnt[0]%5 ; )
                    {
                        n -= (f+h-j*l+l-a-j+e-e*h)%100;
                        d -= (e*f*b+n+i)%100;
                        b -= (h-j)%100;
                        i -= (e-b)%100;
                        m -= (i*a-j)%100;
                    }

                    switch( ++SWcnt[0]%3 )
                    {
                        case 1:
                        {
                            l += (j-a+b*m+g-j*a*e)%100;
                            f += (j-l-c+i+j-i-e+i-n)%100;

                            m = (c+a+l-b*j-b-h*b+g-
n)%100;
                            l += (l-b-f+e-i*g-n-c-c-
i+i+m+i+e)%100;
                        }
                        break;

                        case 2:
                        {
                            a += (b+f-d*g-a-d*a+c-k-e-
j)%100;
                            d += (a+c+f+j-e)%100;
                            d += (h-f)%100;
                            c += (j-h*c*k-f*a+n-a*n-
e)%100;
                            f -= (k+d+f+a+l+g+d-a*i)%100;
                            k -= (d-e*d*h-n-j-b*i*b)%100;
                        }
                        break;

                        default:
                        {
                            m -= (k+b*a-h*m+k+k-j+i*g+b-
m)%100;
                            j -= (f+l*d)%100;
                            j += (k-c-f+c)%100;
                            h += (f+d+j+h-e+a+h+g+m+n-
h+n)%100;
                            b -= (h-f+d*k*n*n-l+i-g-
b)%100;
                        }
                    }

                    if( ++IFEcnt[0]%2 )
                    {
                        l += (i+m+g+c*a*k)%100;
                        b += (m+g-n+k*e+b-a-
b+j+i*m)%100;
                    }
                    else
                    {
                        d -= (e-f-m-a)%100;
                    }
                } while( ++DOcnt[0]%5 );
                while( ++WHILEcnt[0]%5 )
                {
                    do
                    {
                        n -= (l+g)%100;
                        c -= (l*h)%100;
                        m -= (e+h-a-l-d+m-a+j+f-j*n-f-
a+m)%100;
                        i -= (i*m*n*a-h*m)%100;
                    } while( ++DOcnt[1]%5 );
                    for( ; ++FORcnt[1]%5 ; )
                    {
                        f += (f+l)%100;

                        l += (b-m-d-b-m*n+c*c-d*a-
a*n-e+c)%100;
                        g = (a*f+g)%100;
                        k += (b*m+i+b*l-d*k*n-
c)%100;
                        c -= (c-l+c+f-c*h+e-f)%100;
                        g += (g+g+b+b+m*c-n+k-i-
b+k)%100;
                    }
                    if( ++IFcnt[1]%10 )
                    {
                        h -= (d+d+g*e+g+d-a-n-
a+g)%100;
                        c += (e+l+j+j+n+h)%100;
                        a += (j*m)%100;
                        n -= (f-m-i*i)%100;
                        d -= (h*h-d+g*l)%100;
                    }
                    for( ; ++FORcnt[2]%5 ; )
                    {
                        n += (n+f-i-b+k+i-l+d-d)%100;
                        k -= (k+c-l-l)%100;
                        f += (h+j+e+d+g*d-f+h+c-
c*i+n+i)%100;
                    }
                    if( ++IFEcnt[1]%2 )
                    {
                        i = (l+f-d*a*e*m+f*a-
k*g)%100;
                        a -= (a*n+n*g-m+a)%100;
                        a = (k*j+b+d+l+f+l)%100;
                        b += (g-n*m+l-f-c*k+m-c-d+l-j-
i)%100;
                    }
                    else
                    {
                        j += (h*l+g+i+c+j-
g+a+l+c+c)%100;
                        f -= (h*l-g-l)%100;
                        m -= (g+i+a+d)%100;
                        i -= (g-k+j-h-l-i+n-d-e+l-
k+k*g)%100;
                        f += (g-j+g)%100;
                    }
                }
                g += (j+b*k+h-m*j-a+j*i-m*k+f-
m)%100;
                g = (b+g-n-n-f+i-l-g)%100;
                l -= (n+d*j*b+i-i)%100;
                d += (c+n+i)%100;
            }
            a -= (g-d)%100;
            g -= (n-f+m+n+d*d+e*l-b-
c*b)%100;
            j -= (g*k+i+f-b-m-n-c-g-j-f-d)%100;
        }
        d += (k+g-b*d+m)%100;
        d -= (i+n*c-j+c-e-f)%100;
    }
}
```



```

    d += (f*m+b-a+m-j-f+e)%100;
    j += (l-m*l+n+g*n-c)%100;
}
k -= (g-g*e)%100;
m -= (b-i*i*m-c+f*c-n-e-g*c)%100;
i += (m-e-e*e+i*a+d-k-g-a)%100;
i = (g-m-i)%100;
f += (c+j+f+a*k+a-g)%100;
b += (f+g-d*n-k+b-c*m-e-n)%100;
g -= (l+d)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F2(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[4]%5 )
    {
        do
        {
            g -= (h+e-m*m+c+l-m-c*h*i+j+f-
k*c)%100;
            for( ; ++FORcnt[6]%5 ; )
            {

                switch( ++SWcnt[1]%3 )
                {

                    case 1:
                    {
                        if( ++IFcnt[2]%10 )
                        {
                            i += (h+a+d-j*h-g+k-b-
l+h+c*a*k)%100;
                            k += (f+k-g+a+c-m+i-e-c-
e+m)%100;
                            f -= (j*m-n+i+a-b+i-g-d+l+a*e-
f)%100;
                            f += (m-h)%100;
                            n -= (e+j*c)%100;
                        }
                        if( ++IFcnt[3]%2 )
                        {
                            d =
(n+b*l+h+a+n+c*e*e+d+n*i*i+g)%100;
                            b += (m+e*l*b*h*k+a-
d+b+d)%100;
                            m += (h+n-i*k-d-b-l-
h+b+f)%100;
                            i += (m+i)%100;
                            a = (h*n*l-f-d)%100;
                        }
                        else
                        {
                            a += (f+m+h-d)%100;
                            b = (h+k+n*h)%100;
                            k -= (g*d+b+l+m+l-e*h-k-
c)%100;
                        }
                        while( ++WHILEcnt[2]%5 )
                        {
                            l += (f+a+h+c*g+m+i+k-a-l-
g)%100;

```

```

                                a -= (d-g+k+d-g-j+g-
e*h+g)%100;
                                a -= (l-k*c+e-b+f-k-b+a+i-
h)%100;
                                a = (n+c)%100;
                            }
                        do
                        {
                            b += (e-e-b*l*g+f+j+m)%100;
                            g += (e+b+c-d)%100;
                            j += (f*d*m+c*f-b)%100;
                            g -= (j*i)%100;
                            n += (m-e+h+e)%100;
                            g += (m-b+l-l*d-j-l)%100;
                        } while( ++DOcnt[3]%5 );
                        for( ; ++FORcnt[4]%5 ; )
                        {
                            a += (j-i+b-k+j-f-f-n+e+m-
e)%100;
                            i -= (j-k+a-a-n-k-h+j+h-
j+e)%100;
                            j += (b-h+e-i)%100;
                            n -= (i+k-e-a-c-g+b+n)%100;
                            b -= (f-g-i+a-g*e-h-n-
j+m+a)%100;
                        }
                        for( ; ++FORcnt[5]%5 ; )
                        {
                            b += (h+m+h)%100;
                        }
                    }
                    break;
                case 2:
                {
                    if( ++IFcnt[4]%2 )
                    {
                        k -= (a-c*n*n+m)%100;
                        n -= (n-a+l)%100;
                        e += (h*i+a+i-a+g)%100;
                        h += (h-j-l*b-l-i-c-n-b-h)%100;
                    }
                    else
                    {
                        h += (c+a)%100;
                        k -= (d*a)%100;
                        a -= (j+j+m-k-i+h*k-
h+h+m+g)%100;
                        l = (i-i*i*g-b-l-k-e+c+a-n+h-
d+d)%100;
                        d += (h-f+k)%100;
                    }
                    while( ++WHILEcnt[3]%5 )
                    {
                        d += (h+e-f+n+f-n+k-k+g+k-
l)%100;
                        b -= (n+n*k*h*m-g-
g+i+j+c*i)%100;
                        k -= (j+m-i-d-e-m)%100;
                        b -= (j+d-h*d+m*h-b+l-n-
a)%100;
                        m -= (l*a+a-h+c+m-n)%100;
                        g += (b+k-b-g-j*a+e-g*j*e-
b)%100;
                    }
                }
            }

```

```

                                break;
                            default:
                            {
                                do
                                {
                                    b -= (d-k*j*b*n+m+j-f-j-
f+b+a*g-a)%100;
                                    j += (l-l+c)%100;
                                    b += (m-a-a-a-k*g+i-e-
n*f*n+f)%100;
                                } while( ++DOcnt[4]%5 );
                                b -= (c*b-c+d-e-i-i)%100;
                                g -= (b+k+g-h*i-a)%100;
                                d -= (m+i-j-n+b-h*g+d-i-n)%100;
                                k = (h+k+b*a+c+c-c-f*b-m*d-
j)%100;
                            }
                        }
                    i -= (d*c+m-k*b*c*f-
n+d+g+m*k)%100;
                    c += (h-b)%100;
                    l += (i-n+i-e-n)%100;
                }
                f += (e+a+d-n*l+c)%100;
                c = (c+a+b+c)%100;
                l -= (k+c)%100;
            } while( ++DOcnt[2]%5 );
            d += (h+g+e-n-n*c*h)%100;
            d += (h+k)%100;
            f += (h+c+e-m*l-b-f*f+n*e)%100;
            e += (i+i+b+n+g*f+m+d-h)%100;
        }
        l -= (m-j)%100;
        i += (l-k*d-f+k-n-c)%100;
        k += (k-e*l+e-m+d)%100;
        m -= (k-g+b*k+n-j*a-i+g+n-e+b)%100;
        n += (l+i)%100;
        c -= (n+k)%100;
        g += (i-l-n*h+g-f+d-a+f+n+n*l+d)%100;
        return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
    }

    int F3(void)
    {
        int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
        a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
        if( ++IFcnt[5]%10 )
        {
            m -= (k-c+a+k+j-j-g+l+g-e*e)%100;
            for( ; ++FORcnt[10]%5 ; )
            {

                switch( ++SWcnt[2]%3 )
                {

                    case 1:
                    {
                        if( ++IFcnt[6]%2 )
                        {
                            if( ++IFcnt[5]%2 )
                            {
                                b += (j-l+d+j-a+g+k-
d+h+c*j*d+b)%100;

```

```

j += (c-e-k+k+i-e)%100;
d -= (b-j*a+g-a)%100;
i = (f+d+f*h-j+m-k+b-e-
k+g)%100;
m += (m-i-m-a+h-b-m+n-
f+h)%100;
a += (b*a-c+d+k-a*j-l-
h+m+f)%100;
}
else
{
a = (l-c-c*k-c*g*n-e-f)%100;
m += (d+g*b-l+a+c)%100;
k -= (a*b*c+f*g+k*h-l-g*e-
n+m*j-l)%100;
f += (i*d)%100;
}
while( ++WHILEcnt[6]%5 )
{
k = (n-m-f)%100;
j -= (n-b)%100;
f += (d+l)%100;
e += (c-b-n-i+l+n-e+g-b)%100;
d += (n-i*j-f-j-n*m+a-
h+a+c+l*e)%100;
}
do
{
h += (j*g+d)%100;
} while( ++DOcnt[6]%5 );
e -= (e-a-c-a)%100;
for( ; ++FORcnt[9]%5 ; )
{
l -= (f-a+f*n+f-c-a-m-
g+g+c+j)%100;
m += (f-j*e-b+n+g-n+m+a*d-i-
f-e)%100;
}
}
else
{
while( ++WHILEcnt[5]%5 )
{
j += (k*h)%100;
n += (m-e+i+l+i+l-i-f-b)%100;
j -= (a*i-e)%100;
n += (d+k*h-m+a+l+f+c-
g)%100;
k += (i-g-e-n-a*n-d*m+i+g-h-
b+f)%100;
}
do
{
g -= (m-c-d-m-g*i*c*m-b+e-
j+f*d-i)%100;
a =
(n*h+e*l+m*m+e+k)%100;
h += (g+j+a*b)%100;
i += (f*j)%100;
} while( ++DOcnt[5]%5 );
for( ; ++FORcnt[7]%5 ; )
{
e = (c-b+l+l)%100;
d -= (f-a-i*b)%100;
f -= (n+e)%100;
l += (n+a)%100;

```

```

j += (d*a*j-f+d+n-i*a+k-j+i+i-
n+e)%100;
b += (l+d+l-a+c*b-g+h-j-
e)%100;
}
if( ++IFcnt[3]%10 )
{
m += (c+g)%100;
h += (k+h-f-h*h*g-n-
b*g)%100;
k += (b*n-j+d*n+i+i+d)%100;
b = (e+b-i)%100;
a -= (d*e+j)%100;
}
for( ; ++FORcnt[8]%5 ; )
{
k -= (e+m-g+g+f)%100;
g += (c+h+g+i)%100;
d += (c-g+n*c+c-f+b-i+i+n-i-
c)%100;
}
}
switch( ++SWcnt[3]%3 )
{
case 1:
{
if( ++IFcnt[4]%10 )
{
b += (b-m-j-j+j+c-e+a*j-h-
i*m)%100;
l += (a-c*i*b)%100;
a += (f+f*e+k*h-b+b-
c*f+a+i)%100;
k -= (i-l+j*h-b+l-k+a-m+i-
b)%100;
}
if( ++IFcnt[7]%2 )
{
d += (d-c-n+m+d-h+b*e*g*e-
g-i*i)%100;
e += (j+b*b)%100;
j += (l*a)%100;
g = (e*l-f+k)%100;
n -= (j-c+m*k+d-n-k-l+m-
a+e+m)%100;
}
else
{
g -= (l+d*l+e-h-h*f-f-f-h)%100;
i += (d+e+d*g-k-n*c-j)%100;
d -= (f-n+f)%100;
b -= (a*j+a+a*f-f-i-j-
l*j+h+e+k+f)%100;
f -= (b*j*b+m+c*j*d)%100;
l += (d-k-g-e-i+c+f-b+i)%100;
}
k += (m*c-k-h-l-f-g*k*l-j-
c*f+h)%100;
i -= (c*i-a-m-h*m-l-g+m-
h+l)%100;
a += (c*e*n*c+k-a+e*b-
g+g*k)%100;
}
break;

```

```

case 2:
{
k = (n+j+i-c*d*h*f*d-e-
g+i)%100;
n += (e*j+e-b*j-b+m-b-d*j+a-
c)%100;
l += (b+j*j+l-a*e-m)%100;
}
break;

default:
{
m += (d+g+b*i*e+l-d*i+g-c+m-
g+c)%100;
m += (g-f*d-k)%100;
j = (m+h+i-h*h-m)%100;
h -= (m+d)%100;
h -= (m-l-n-k+d+k-k-i*c+n)%100;
}
}

f += (l-g-g-m+j+n-c+b-e*m-
d)%100;
e -= (l+c+m+h+l+g-c*g-b+m-
g*d)%100;
a = (k-a*k+d*n-g+b)%100;
h -= (k+n)%100;
}
break;

case 2:
{
i -= (c+d-n)%100;
m -= (n-j*n*i+f-k*c)%100;
c += (m+k*f-a-g+b)%100;
e -= (a*n+e-l*k+d+l-c)%100;
}
break;

default:
{
m += (c+h-b+m+c+g+k-m-
a*i)%100;
b += (l+h)%100;
f -= (d+k*g*a+n*h+m-j-n-f)%100;
n -= (a*f+h+c-c*e+a)%100;
h -= (j+d-k-h*a-k-b)%100;
m += (l-h+j+b+f*b-l)%100;
}
}

h -= (e-j*b)%100;
l = (g*l)%100;
a += (a*n*m-a-c+k)%100;
}
h -= (g*a+b-h+e*m+k*a+m-
b+l)%100;
m -= (n+d-h-f-c+d+n-k-g+j+l-c-
g)%100;
l = (m-l+d*j)%100;
}
k += (m-b)%100;
i -= (e*g-g-l+h-d)%100;
e -= (g-e+n+a-k+n+h+a+a*m+h-
g)%100;

```

```

j -= (d-n-i-l*d)%100;
k -= (a-g-f+d*i-l-a*b-n)%100;
e = (k*i+b+j)%100;
l += (h+j-k-e*f*i*f+e)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F4(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[8]%5 )
    {
        do
        {
            for( ; ++FORcnt[13]%5 ; )
            {
                for( ; ++FORcnt[12]%5 ; )
                {
                    if( ++IFCcnt[8]%2 )
                    {
                        b += (n*n)%100;
                        l -= (l+f*k-h)%100;
                        e += (c*i-d+f-m-a-l*n+a*a-
m*f-f)%100;
                    }
                    else
                    {
                        b -= (c+c-d-d)%100;
                        l = (n*d)%100;
                        m -= (e+h-e-b*l+m+k+c)%100;
                        n -= (d-l+a-a-i-a+b)%100;
                        d += (n+c-d-i+d-i*k*m*e-k-
d+f)%100;
                    }
                    while( ++WHILEcnt[7]%5 )
                    {
                        m -= (n+d+c+n-m)%100;
                        f -= (f+a-k*j*e*g-
k+d+g+h+l*c+d+m)%100;
                        e -= (c-f)%100;
                        n -= (d-j)%100;
                        c -= (e*f*c-j+g*g+a-n-h-d-b-
l)%100;
                    }
                    do
                    {
                        n += (m-j)%100;
                        n += (i-f+a)%100;
                        a -= (m-n+a)%100;
                        g += (l-l*e)%100;
                    } while( ++DOcnt[8]%5 );
                    if( ++IFcnt[6]%10 )
                    {
                        d -= (h-a+k-i+d)%100;
                        c += (k+h-j+c*l+b-
a+c*d+i+k+n)%100;
                        b = (m-k+j+b+n-k+l*j+g)%100;
                        m -= (f+i+m*m*n*f+e-m+c-
h*k)%100;
                        c -= (l-h-a*k*i*k-
d+e*b+i)%100;
                        m -= (e-j-c-c+l+n)%100;
                    }
                }
            }
        }
    }
}

```

```

b -= (c-
d+b+n+a+d*m+n+k+b+k*h+a*b)%100;
for( ; ++FORcnt[11]%5 ; )
{
    e -= (k*e+n-j-m-a+b+k*i-d-
e*b)%100;
    j -= (g+m+n*e+b)%100;
    n -= (f-f-c-e-i*m)%100;
    k -= (b+l-a+a-a+g+d*i+k-
n+k+i)%100;
    a += (k*f+m-f+i*a-g+c+n*m-
b)%100;
}

switch( ++SWcnt[4]%3 )
{
    case 1:
    {
        a -= (i-f+j+b*d*c+m+c+d)%100;
    }
    break;

    case 2:
    {
        k += (b+e*m-c+a*m*k-b-b-
n+l)%100;
        h += (g-m*m-a-c*e)%100;
    }
    break;

    default:
    {
        m += (l*c+j+a*a+h-i+k+f+k-c-
e*a)%100;
        g += (f+c+j-e*k+n-e+k-a-c)%100;
        k += (b-h-h+m-i*g+d-m-
b+d*b+k+l+j)%100;
        k += (d+n+a)%100;
        i -= (h*k)%100;
    }

    a += (g*h-e+c)%100;
    n -= (i-g*g-d-g-n+l+n-h-c-h+g-
j+i)%100;
}

i += (i-n+f-n+f-c-k)%100;
m += (k+m+c*c+m)%100;
c += (f+i-d-i+h-b+e*d*c+d-c+l+j-
h)%100;
e += (f-k)%100;
} while( ++DOcnt[7]%5 );
n -= (h-d-h-f-n-m+h+f-b-
a+g+k+e+k)%100;
k = (g+h+d-l+l*k-f*g-l+e+f)%100;
j -= (c+g+j+f+j)%100;
f -= (i-g*f*a+c+e+e+l-c+f-e)%100;
}

e -= (b-a+j+c-k+g-j)%100;
n -= (f-i-h-j*e-i-i)%100;
k -= (i-l)%100;
h = (l*g-i*g+b*a+c+k)%100;
c -= (n-n-i-n+c*n-a*n)%100;
m -= (i+j-e*e*n*c)%100;

```

```

c -= (m-l-f+n+l-l+e)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F5(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCcnt[11]%2 )
    {
        n -= (d-k-g*l-k+d+k-d+e-g-i)%100;
    }
    else
    {
        while( ++WHILEcnt[10]%5 )
        {
            do
            {
                for( ; ++FORcnt[15]%5 ; )
                {
                    if( ++IFcnt[7]%10 )
                    {
                        c -= (f*k-f*f*d-k)%100;
                        f += (j*h-f-k-b)%100;
                        l -= (n+g*l-d-k*k-n*b)%100;
                        k += (g*l+d+a+c-b+e-
n+k+g*c)%100;
                        d -= (h-d*n-h-l-l)%100;
                    }
                    for( ; ++FORcnt[14]%5 ; )
                    {
                        l += (a+f+j*d-f-f)%100;
                        j -= (i-k+d-b-j+d+h-d)%100;
                        h = (c+b+e-i*h-g*e*a*b+f-
d)%100;
                        b -= (c+h-b*j+h-
b+d+h+i*i*f*e*c+i-i)%100;
                    }
                    if( ++IFCcnt[9]%2 )
                    {
                        a += (e*j-f-d-i+f+g-f-k+i*b-d-
m+b)%100;
                        i += (b+d-n)%100;
                        n -= (l+c-e+j)%100;
                        d += (a+d)%100;
                        k -= (i+g-b-b-d-g+a-j-
h+a)%100;
                    }
                    else
                    {
                        m -= (m+e*f+l)%100;
                        j = (n+b*m-c-b-j-j+j+a-
h+k+e)%100;
                        l = (j*c-a+g+m*e-b*i*n)%100;
                        a += (l+n+f+j-d-b)%100;
                        m -= (b-l-k+d+j+b*j+k)%100;
                        d -= (m-j)%100;
                    }
                }
            } while( ++WHILEcnt[9]%5 )
            {
                do
                {
                    a -= (e*a-l*e*f)%100;
                    d -= (g-f+e+m+a+l-i)%100;
                }
            }
        }
    }
}

```

```

        d += (k+j+l+a-h-k-m*m)%100;
        m -= (f+g)%100;
    } while( ++DOcnt[10]%5 );
    l = (c-m-
f+g+j*i*k*k*l*a*f+k)%100;
    for( ; ++FORcnt[16]%5 ; )
    {
        h -= (i+n+l+h+a*f-l)%100;
        i += (i+i+d+b+a+i-n-
c+g*n)%100;
        n -= (g*h-f*h)%100;
        k += (j+f)%100;
        k -= (a-k-e*j-d+i+l+g)%100;
        b += (c+h-l+d-e+n)%100;
    }

    switch( ++SWcnt[5]%3 )
    {

        case 1:
        {
            c -= (a+m)%100;
            h += (l-f-k*e*c*h*f-
i+i+a*g+c)%100;
            l -= (d*b+g-k)%100;
            c += (e-l+m-e-
m+c+e*g+d+b+c*j+f)%100;
            g += (c-g-f+j-l*j+l)%100;
        }
        break;

        case 2:
        {
            n -= (l-g+i+j)%100;
            n -= (a*b+n-m*h)%100;
            k += (b+n-n)%100;
        }
        break;

        default:
        {
            e += (k+k-g)%100;
            j -= (e+f+e*k*j-d*m-n+h*m-c-
k)%100;
            e -= (h+l-m-k)%100;
            i -= (h+m-c+m+b+c+h-
i+m)%100;
            e -= (j+m+j-c+i+d)%100;
        }
    }

    if( ++IFcnt[8]%10 )
    {
        l += (n-c+n+l+k+h-l+i+d-
i+k+d)%100;
        i += (l-l*k+j-c+n+a)%100;
        e += (m*k*c-k*b-l+d)%100;
        i -= (b*i+a-j-l)%100;
    }
}
if( ++IFEcnt[10]%2 )
{
    d += (b-b)%100;
    f += (f-a-e-m)%100;
    e = (g+a+m+l+d)%100;
    l -= (m+k-d*g)%100;
}

```

```

        c -= (n-n+h*f-d+g-
m+a+i+b+n)%100;
    }
    else
    {
        m += (b*h-j*e-b)%100;
        i -= (f*m+j)%100;
        e -= (m+i*l+j-h)%100;
        a -= (n*e+m+k-c+j)%100;
        k -= (j+b+h+c-g*c-k-
m+l+k+l+n+h+a)%100;
        l = (j-i+j*i-g+n-h+n)%100;
    }
    m += (i+j*n-f+a-a+k*g-
k*e+g+a*e+d)%100;
    m -= (c-l-k+n-i+j)%100;
} while( ++DOcnt[9]%5 );
d += (m*g-i-b*d+c-l+f*h*j-l-f*n-
a)%100;
f += (h-n-g)%100;
h -= (g+n+b-m-m-k-d)%100;
m += (i+a-j-e*a*n-l*c+c*n)%100;
f -= (f+g-k*g+g-g-d-l-e+n)%100;
}
g -= (m-j-j-f-k)%100;
j -= (m*f-d*h+b+a*k*l+l+k+d+n-
h)%100;
c += (a+c+m*n*j*a*e*k+j-n)%100;
}
d -= (l+b)%100;
f -= (g-m+h+h-j+j+a-g-j+h+b+m-
b+d)%100;
b += (l+e*e*f-l+g+a*f-m+n+n*i)%100;
d -= (j-e-g+l+n-d*f-n+b-c)%100;
j += (j-f+c+c-e-m-d+d)%100;
j += (m-i*n)%100;
h -= (a*k)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F6(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[13]%5 )
    {
        do
        {
            for( ; ++FORcnt[20]%5 ; )
            {
                for( ; ++FORcnt[18]%5 ; )
                {
                    if( ++IFEcnt[12]%2 )
                    {
                        g -= (i*j-m*j+a+g)%100;
                        b -= (e-a-n*e*a+l*m-k+g-c-m-
n-g+m)%100;
                        a -= (i*m*m-b-n+n+m*h*k*g-
m*e)%100;
                    }
                    else
                    {
                        n += (n*i+i-a)%100;
                        a += (h+m)%100;
                        c += (n-e-h*i)%100;
                    }
                }
            }
        }
    }
}

```

```

        h -= (h+e+l*n*f*m-e+c-j-
b)%100;
        e += (d-k+k*d+d-i-
d+c+h+n+a)%100;
    }
    while( ++WHILEcnt[11]%5 )
    {
        j -= (j+h+i+j+m+l*j-f*h+g-j+l-d-
h)%100;
        d += (i+i-c+c-a+c)%100;
        n -= (h+h)%100;
        m -= (k-g-e-f*b-i+j-f-d-
d+f+f+l)%100;
        l -= (m+d*g-j-a*f)%100;
    }
    do
    {
        l = (g-a+g*m-e+k*n-n+m-e-
i*f)%100;
        k += (i*f+b)%100;
        n += (c-a)%100;
        k += (k*f-c*m-b-j-k-i-
n*c)%100;
    } while( ++DOcnt[12]%5 );
    if( ++IFcnt[9]%10 )
    {
        i = (l*f-e)%100;
        f += (e-d+l-e+c+n+j+h-l-
m+f)%100;
        k -= (h+a-i-i+i+d*d-j-g-d-
g+c+k-k)%100;
        d += (h-g*h*n*j+n*g-f-n-c-c-
d)%100;
        a += (i-a-k*a+f+b*e-f-
n+m*e+j-a-m)%100;
        c -= (n-m+a+e*b-a+i*k+l+f-c+j-
b)%100;
    }
    l += (m-m+l+c)%100;
    for( ; ++FORcnt[17]%5 ; )
    {
        k -= (d*g+a*l+e-k+b+g-m-
a+d)%100;
        m -= (i-h)%100;
        k -= (j+m)%100;
        b += (a-l-l-n-l-g-i+i-b-m-
i+c)%100;
        j -= (n-m*e-c+a+h-i*e*n-
j)%100;
    }
}

switch( ++SWcnt[6]%3 )
{

    case 1:
    {
        if( ++IFEcnt[13]%2 )
        {
            i += (k+b+g-c*b+e+c-g)%100;
            b += (i+a+g-i-e+l+h+c-g)%100;
            b -= (a+j)%100;
            d += (j*c-e*i*d-e-g)%100;
            g += (k-k+b+h+j+d+b*i-
d+h*e*c)%100;
        }
    }
}

```

```

        i = (l+e+d-j-h-i-d+i*e+i*n-n-
l)%100;
    }
    else
    {
        g += (i+g+d-a+i-c+j+e-i-
n)%100;
        d += (i+h*b+m)%100;
        j += (g-k)%100;
        h += (m*i-j)%100;
    }
    while( ++WHILEcnt[12]%5 )
    {
        i -= (m*b-f-i-b+c*c*g+n*a+m-
g)%100;
        g += (g-a-j)%100;
        c -= (n-b*h*f-e-e-l+f)%100;
        l = (n+f+b-h)%100;
        i -= (f+n+c-j*f-n+i+n)%100;
    }
    do
    {
        f += (m*h)%100;
        n -= (b-c-c)%100;
        j -= (h+m+e-c-i-c*i-l*g+i+n-
h*b+j)%100;
    } while( ++DOcnt[13]%5 );
    for( ; ++FORcnt[19]%5 ; )
    {
        a -= (c+l-f*l+h-h+f+l-i*g)%100;
        h += (f+l+a+i*m+n+l-g-m+e-
m+a)%100;
        b += (k-f-e)%100;
        i -= (g+j+e+m+n+n-h-k)%100;
        k += (n+b-g-m*c*c+k+k-n-j+h-
m+c)%100;
    }
    h += (a+l*e+a-k*g)%100;
}
break;

case 2:
{
    d -= (h-h+k*e-g+e+g+e+e+a-
g)%100;
    n -= (e+l)%100;
    e += (k+f+i-c+n+i)%100;
    i = (g+d-j-c-c-d-h-f)%100;
}
break;

default:
{
    e += (c-e+n-d-k)%100;
    a -= (a*e-i+l-h*d+l-c-j)%100;
    f -= (m-i+f*i+n-f+e+d-l+e-j*h-
j)%100;
    g -= (l+b*g+b*j)%100;
    i += (e-c+h*k+i)%100;
    b -= (b-g+i+h*b-b+g-a-d-d+b-
k)%100;
}
}

d += (h-h-n+c*b-k+k+b*a)%100;
c += (j+h)%100;

```

```

    }
    j += (d-k-a+g-f)%100;
    e = (i-a+k+m+d+h*h-c-d)%100;
    d -= (k+b-d+h*n-l+c+e-i*k*f)%100;
    c += (b-j-b+l-i-g*f*j+l*n)%100;
} while( ++DOcnt[11]%5 );
a += (a+c)%100;
}
l -= (l+n+j)%100;
l += (e+a-e+j+c-e)%100;
f += (d+d+f-g+j+k)%100;
m += (l*l-d+m+g-b-k)%100;
h = (k-k+b+b+n-j+c-c+g-d-m)%100;
b += (k-e-j-e-l+l-f+a-e-a)%100;
b += (g*c+e+l-d+n*d-h+i+i)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F7(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[11]%10 )
    {
        for( ; ++FORcnt[23]%5 ; )
        {
            if( ++IFcnt[15]%2 )
            {
                b -= (d+b+b+h-k-j-i-b)%100;
                h -= (c*b*e+j+k+b-d-c)%100;
                e -= (h-b+a-b*m+n*n*b*b*c)%100;
                i = (m-j-d-d*f-d+a)%100;
                f -= (l-h)%100;
            }
            else
            {
                while( ++WHILEcnt[14]%5 )
                {
                    do
                    {
                        e -= (j-e+k-k-a-l)%100;
                        b += (h*e)%100;
                        m += (i-a-m*a+n+d-
n*d*e+h+g+f+a+m)%100;
                        e += (b-k)%100;
                    } while( ++DOcnt[14]%5 );
                    b -= (b*j+c-a+b*d+c)%100;
                    for( ; ++FORcnt[21]%5 ; )
                    {
                        i -= (m+n+n+f-b-h+f*i-n-
m)%100;
                        a -= (m-m*c*b-e-
d*a+m+d)%100;
                        f += (n*n-f*k-f-h+m+b-i+j-
n)%100;
                        e -= (m-f-i)%100;
                        d += (i*f+b*i-c-e+h-e*m-
d+i+i)%100;
                        n += (a-e-e+f-b*h+k-
c+a)%100;
                    }

                    switch( ++SWcnt[7]%3 )
                    {

```

```

case 1:
{
    b += (l-e+f+i+h)%100;
    f += (c+i+j*a-j-m*n-h-
a+a)%100;
    b -= (n*i*f-a-c-k-m-b+f+a+e-
b)%100;
    j -= (m*d-j)%100;
    e -= (b+h)%100;
}
break;

case 2:
{
    k += (b+f*f*f-k)%100;
    g += (j-k*a+i+h-f-n-k)%100;
    i -= (f-b*m-m-i+n+e+c+d*j-
j)%100;
}
break;

default:
{
    n += (g+k+h)%100;
    l -= (j*l*j*e-n-e*e*m-d)%100;
    j -= (c+k-l+h)%100;
    b += (g+g+c*m+m-l)%100;
    f += (a*f+f-m*f-a-m-n-m+j-
l+a)%100;
}
}

if( ++IFcnt[10]%10 )
{
    m += (m*l)%100;
    j += (l-l*n)%100;
    a += (d-a)%100;
    b += (l*h*h*b-j+a*d+l-
a)%100;
}
}

if( ++IFcnt[14]%2 )
{
    i -= (n-g-f+f+k-h-k-i+m-b)%100;
    m -= (l-f*n+k-j+b+l*i+l-d*l*c-
l+c)%100;
    j -= (m+m*c*e*j-h+c-g-g)%100;
    i -= (l-f+b*b+f-a*h+f+e*e)%100;
    g += (d*l*n)%100;
    g += (e-e-i*i*d-i-k)%100;
}
else
{
    while( ++WHILEcnt[15]%5 )
    {
        h += (m-k*g*j+m+b+k-
b)%100;
        f += (d*n-i-c-f+g+l+a-j*a*e-
j)%100;
        j += (h-d-f*m-i-d)%100;
        b -= (a+a+b+g-b-g+a)%100;
        j += (a+l+j-i+n-c-
g+e*m+l)%100;
    }
}
do
{

```

```

i = (j*n-a-d+e+d-h-k-
m*i+f+d+g+h)%100;
k -= (b*j+i+i-i-l*f*a-
f+i+n)%100;
d -= (d*n-n-e*c-d*j)%100;
l -= (f+f+g*a)%100;
n += (c-g+k*g-g+m)%100;
} while( ++DOcnt[15]%5 );
for( ; ++FORcnt[22]%5 ; )
{
h += (j*l+k-c+j+n-c-c-a)%100;
d = (k-h)%100;
g += (m*e-m+a+j+m*h+e-
j)%100;
a += (f+n-b+m-f+n-j-l*i*f+d-
c+n+i)%100;
}
k += (c*d)%100;
g -= (i+c*c+n*a+b-k+n-
e+g)%100;
h = (g+m+c-n+l*j+e-e-g-b-
b*g)%100;
}
}
f += (a-l*f*j-f+d+c*e*a)%100;
i += (i+f+l+m+l)%100;
i -= (g-e+j-e+b)%100;
g += (h*k-a-e-l+k+e*g+m+m+g)%100;
}
l = (l*n+c+l)%100;
c = (c*j-h+d-l+h*a-i+b-f-k)%100;
i += (e+n+k-m)%100;
m -= (b*c*e*l)%100;
n += (m-i)%100;
f -= (j-c*k+k-d-f-h+h*m*i+b-d)%100;
e += (m-b-e-n)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int main(void)
{
int l;
clock_t StartTick = clock();
for(l=0; l<19; l++) IFcnt[l] =0;
for(l=0; l<25; l++) IFcnt[l] =0;
for(l=0; l<12; l++) SWcnt[l] =0;
for(l=0; l<25; l++) WHILEcnt[l]=0;
for(l=0; l<25; l++) DOcnt[l] =0;
for(l=0; l<38; l++) FORcnt[l] =0;
long int sum=0;

sum += F1( ) ;
sum += F2( ) ;
sum += F3( ) ;
sum += F4( ) ;
sum += F5( ) ;
sum += F6( ) ;
sum += F7( ) ;

{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
for( ; ++FORcnt[25]%5 ; )
{

```

```

if( ++IFEcnt[17]%2 )
{
l += (k+j+n*c*d+k-k-h+h-k-
l)%100;
h -= (m-m-n+c)%100;
j -= (c-g-e+g-m*c-d-m-g+l-l-d-k-
e)%100;
e += (a-c-b+j+d-
f+j+d+l*a*c)%100;
}
else
{
while( ++WHILEcnt[16]%5 )
{
do
{
e += (c*l)%100;
b += (f-m)%100;
k -= (j+n+i*f+c-e*j+h-
c+l+g)%100;
c += (f-l+j+i-m+h-k)%100;
h -= (h*a)%100;
d -= (j-c*i*e+d+k+a-a)%100;
} while( ++DOcnt[16]%5 );
if( ++IFcnt[12]%10 )
{
h += (l+c+l+f+a+e+m*n)%100;
a += (b*m-d-k+a)%100;
f += (a*g+a-l+l-a+e*n)%100;
d -= (m-a+g)%100;
i -= (a+h-m)%100;
}
g -= (k-c-k+e*d+e+i+n-n-f-l-
m+f)%100;
for( ; ++FORcnt[24]%5 ; )
{
j += (d-c+f+n+a-k+h+l)%100;
}
}

switch( ++SWcnt[8]%4 )
{
case 1:
{
if( ++IFEcnt[16]%2 )
{
g -= (c+l-d-e*d+n-n+i-g+i-
i+n)%100;
a += (n-d-e-l*i)%100;
d -= (f-m)%100;
g += (l+c+c)%100;
}
else
{
h -= (g-j*f)%100;
g -= (l-e-l)%100;
b += (h*c+l-g+b-d*d-
a*n+j*b)%100;
m += (h-g*i-g+a-l+c+f*k)%100;
i += (i+g-l+c-f+j-l*c*e)%100;
}
}
while( ++WHILEcnt[17]%5 )
{
j += (e+l+l+c+m*k+i+c)%100;
k = (j+b)%100;

```

```

f += (d-a)%100;
l -= (k-l)%100;
c += (e+b-e-n+l)%100;
h -= (h-h-m)%100;
}
}
break;

case 2:
{
b += (b+i+h+l-k-k-f+l+b)%100;
m += (j-g)%100;
j -= (k*e*k)%100;
g -= (i-k)%100;
m += (c-l-k*d-e-d*b-
c+j*l+c*a+a)%100;
}
break;

case 3:
{
n += (n-j+g*m-j*m-b-i-i+b-i+m-
j*l)%100;
i -= (k-j-k-e-k-j+i+d+h+k+b+k*j-
e)%100;
j += (a-h)%100;
}
break;

default:
{
c += (n-e*i-l-k+f-a-j+e-m-
m+h)%100;
e += (e-
b+l*n+h+d+m*b+b)%100;
l -= (b-e+n-f-g+a-e+m-a*i)%100;
m += (n*h*e-
b+b+l+e+d+f)%100;
d -= (i*d+d-j-d+g+f+j+m)%100;
}
}

g -= (j+c*i*b-c-a-h-e+k)%100;
c += (m+n+g*j-n-m*j+j-d)%100;
j += (j+n*k-m)%100;
}
i -= (n+a-m*h+k)%100;
j -= (f-i-j-h-d-l-i+n-i*e+b-n)%100;
}
f -= (j+n-d+h)%100;
i += (k-e+c-i*g-f+c)%100;
d -= (d*f+f+d*j-b-h-a*d+l+b*m-
g)%100;
h += (k+k-g+c-e*f*f+j+f+m)%100;
m += (h-b-a+e-e-b+i)%100;
h = (g+b)%100;
n += (e-l+b-f-i-d-g-g+d+m+g+k-i-
c)%100;
sum +=
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

cout << "\nChecksum = " << sum;
for(l=sum=0; l<13; l++) sum += IFcnt[l];
cout << "\nIF frequency: Static = "
<< 13 << " Dynamic = " << sum ;

```

```

    for(l=sum=0; l<18; l++) sum +=
    IFEcnt[l];
    cout << "\nIF-ELSE frequency: Static =
    " << 18 << "   Dynamic = " << sum ;
    for(l=sum=0; l<9; l++) sum +=
    SWcnt[l];
    cout << "\nSWITCH frequency: Static
    = " << 9 << "   Dynamic = " << sum ;
    for(l=sum=0; l<18; l++) sum +=
    WHILEcnt[l];

```

```

    cout << "\nWHILE frequency: Static =
    " << 18 << "   Dynamic = " << sum ;
    for(l=sum=0; l<17; l++) sum +=
    DOcnt[l];
    cout << "\nDO frequency:   Static = "
    << 17 << "   Dynamic = " << sum ;
    for(l=sum=0; l<26; l++) sum +=
    FORcnt[l];
    cout << "\nFOR frequency:   Static = "
    << 26 << "   Dynamic = " << sum ;

```

```

    cout << "\nRun Time = " <<
    double(clock()-
    StartTick)/CLOCKS_PER_SEC << "
    sec\n\n";

    return 0;
}

```

## b.)BM1BMProgram6

```
#include <iostream>
using namespace std;

#include <time.h>

int
IFcnt[28],IFEcnt[38],SWcnt[19],WHILEcnt[38],DOcnt[38],FORcnt[57];

int F1(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[3]%5 ; )
    {
        if( ++IFEcnt[2]%2 )
        {
            k += (k*f-c+g*h-a-n-g)%100;
            c -= (l+d-f*c+c*c-j*j+b-g+h)%100;
            n += (g+h*m-a+h-d+d-k+k-j+i-m-l)%100;
            e -= (f-m-j-h-k+k*g+l-i+a)%100;
            i += (m*k-i+b)%100;
            f -= (d-b)%100;
        }
        else
        {
            while( ++WHILEcnt[1]%5 )
            {
                do
                {
                    if( ++IFcnt[0]%10 )
                    {
                        k += (m-a-e+j+f+l+n-b+n)%100;
                        h -= (g+a*j+d-i+b+d+j+c-m)%100;
                        j -= (l-c)%100;
                    }
                    b = (j+b-b*g+a)%100;
                    for( ; ++FORcnt[0]%5 ; )
                    {
                        g -= (l*h+e+b+j-k)%100;
                        c += (k+a+a-a-n-m-h-l-k-b)%100;
                        n -= (j-i)%100;
                        g += (e-b-b+i*i*d*i)%100;
                        a -= (j+a+l+j+n+g+l-l*n+c*a-j-f)%100;
                    }

                    switch( ++SWcnt[0]%3 )
                    {
                        case 1:
                        {
                            c = (e*b+a-c*k-m-h-a*i-h*j+e*a)%100;
                            k += (c+i*f+n+j+d-e*g+l+i+k-k+b)%100;
                            g -= (b*j+c-f+c-j*k*n)%100;
                            i += (h-a+i*k+m-b*i-h-e)%100;
                        }
                        case 2:
                        {
                            n -= (e*h-j-a*j*l+e-j-j+d-j+h-m*l)%100;
                            k += (d-i+b-j)%100;
                            b -= (j-b-e-i+i+g*l-i-d+l+d+e+k+m)%100;
                            m -= (h+g*d+d+a-n)%100;
                            b -= (m-b*e-a+a+c+m+h)%100;
                            f -= (k*c*m+a-h*n-l)%100;
                        }
                        break;

                        default:
                        {
                            n -= (i*a-k+d+d)%100;
                            g -= (c-i)%100;
                            i -= (g-l+l+h-c+k*n)%100;
                            f += (j-d+b-l+g-a+e-n+j-l)%100;
                            m += (g+d+h+e*j*d+c-a-d-a*l+j)%100;
                        }
                    }

                    if( ++IFEcnt[0]%2 )
                    {
                        a += (h+g+j)%100;
                        l -= (l+f-h+h)%100;
                    }
                    else
                    {
                        g = (b+h-i+e-i-f+g-b)%100;
                    }
                } while( ++DOcnt[0]%5 );
                while( ++WHILEcnt[0]%5 )
                {
                    do
                    {
                        a += (l+b-k+d+l*i*e-e+g*g*n+e)%100;
                        e += (g-l+n+n-e*a+f*h-g)%100;
                        a -= (f+k*e*b+h+i-b+a+l+k-f*d+b)%100;
                        c -= (i+j*e*f-i+a*d-n+d*f-k)%100;
                    } while( ++DOcnt[1]%5 );
                    for( ; ++FORcnt[1]%5 ; )
                    {
                        l -= (e+m-m+k-g-b)%100;
                        b = (a+b)%100;
                        n -= (e+b+g-m+n-n*k+c+h)%100;
                        n -= (k*i+k)%100;
                        h -= (j-b-j*j*b*j-n+b-k-b+e*m)%100;
                        e += (e+g+c)%100;
                    }
                }
            }
        }
    }

    if( ++IFcnt[1]%10 )
    {
        i = (i-a+i-m*e*a-j*j)%100;
        h -= (f-b-g*n-a+e-c-b-b+n+d-k)%100;
        g += (k+m+h+b-c+m)%100;
        c += (k-e*j+e*g*h+h*j+g-i-j)%100;
        g += (j-l*l-h+j-m+a)%100;
    }
    for( ; ++FORcnt[2]%5 ; )
    {
        n -= (b+b+l+f-d-f-j-m+m+f)%100;
        b += (c-e+n+h-l-h+l-g+e+k+i+c+j*n)%100;
        e -= (a-f*e+l+m-d+b-k-i-e)%100;
    }
    if( ++IFEcnt[1]%2 )
    {
        d -= (l-g*b+b)%100;
        f += (l-b+h-m-k*m*a-d-f-k+e+m-n-m)%100;
        h += (f+k)%100;
        j -= (c*h+l)%100;
    }
    else
    {
        m += (f+i-b+h*l-e+k-b)%100;
        i = (g-g+i+l+i-h+a-g-g-e+j+a+l-j)%100;
        d -= (c-m)%100;
        f -= (f+g+e+n+i*b)%100;
        b += (m*c)%100;
    }
    c -= (d-l)%100;
    k -= (a+c+f+m-m)%100;
    a -= (k-h-k-c*d+b+d)%100;
    n -= (l*d)%100;
    l = (f*b)%100;
    c += (h+f+n*j+a*n+l*k+n-b*m)%100;
    c -= (c-l-g-f-a)%100;
    j += (d+k+l-b)%100;
    l += (e+k-c-b*n*h+m-f+f+m+k-g+h-k)%100;
    f += (m-h*i+k-l-h-c-d)%100;
    g += (d*c-k-g-k-j+b+m-c-j-b-c)%100;
    c += (g+d+a+d+f+j-f-d*k+a+k)%100;
    m += (f+j-a-i-f)%100;
    c += (m+a*m+f*f-l-b)%100;
    g += (e-h+n-c-k-n-b+h)%100;
    n += (n*h*d-b-h+g-n)%100;
    i -= (l*h+f-e-f+j+b-h*j*e)%100;
    l -= (i+j-c+g)%100;
    return
    (a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}
```



```

}

int F2(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[4]%5 )
    {
        do
        {
            c = (f-n*a-k+g-f-j)%100;
            for( ; ++FORcnt[6]%5 ; )
            {

                switch( ++SWcnt[1]%4 )
                {

                    case 1:
                    {
                        if( ++IFcnt[2]%10 )
                        {
                            k -= (i-f-k+m-d-a*g+g-l+l-n-i-
i*n)%100;
                            f -= (j+d+a+n+f)%100;
                            m -= (m+e-g-d*i+b-l+d-j+d+f-l-
j+i)%100;
                            l -= (j-i*l+j)%100;
                            b += (a*b-f*i)%100;
                        }
                        if( ++IFEcnt[3]%2 )
                        {
                            i -= (f-i+k+l+e*c-j-a)%100;
                            l -= (n+g+l-g*n+h+j-j*g-
c*c*l+m)%100;
                            f += (k+m-i)%100;
                            g -= (f+b-b-b*l+l-j+l-
l+b+l+e)%100;
                            k += (k-g+k+m+e+c-h-l)%100;
                        }
                        else
                        {
                            n -= (g+k-c*b+e*i*e)%100;
                            m = (e-g-m*e+n+h*d+j)%100;
                            i -= (b*l-d+h+l-a-l-c-j-b*n+f-
e)%100;
                        }
                        while( ++WHILEcnt[2]%5 )
                        {
                            h -= (i+h)%100;
                            j -= (c+e+d+i-f-c+b+d-
b+d)%100;
                            n += (b+e+k-e+d+n)%100;
                            g += (j*l+n-l+m-j+d*l-j+n-
j)%100;
                        }
                        do
                        {
                            k += (i-g+h+c*c*b+f+h-f+m-
e+a+a-j)%100;
                            c += (n-h)%100;
                            d -= (c+c-i-g+l)%100;
                            h -= (j+n-i*i*j+d-b-
l+i+a+k)%100;
                            n += (f*b+d*f*i-k+a+i*f*j-
n)%100;
                            c += (h+b*j*j-h+j+e-l-n)%100;

```

```

} while( ++DOcnt[3]%5 );
for( ; ++FORcnt[4]%5 ; )
{
    l -= (b-i+b+h-f)%100;
    i += (d+g+m-i*d)%100;
    k -= (d-m+d+b+h+b*j-j-c-b+l*i-f-
c)%100;
    m += (i+h-m*l*j-k*b)%100;
    i -= (c+j-d+c+n)%100;
}
for( ; ++FORcnt[5]%5 ; )
{
    a += (b+d+d-n+g-l+a+c*d*g-
c)%100;
}
break;

case 2:
{
    if( ++IFEcnt[4]%2 )
    {
        f -= (j*i-k+d+g)%100;
        m += (g-n-l-k+k+n+d-
m*i)%100;
        l += (h-l*j*c)%100;
        i = (f*n+h*c+f+m+k*l-d+e+b-
a*l-k)%100;
    }
    else
    {
        f = (b-a+j+m)%100;
        n += (g*c-j+d+k)%100;
        m -= (d-j)%100;
        n -= (m-d)%100;
        j -= (d-c+a*n+i*j-d*k+m)%100;
    }
    while( ++WHILEcnt[3]%5 )
    {
        d += (l*l-e*b-f*b-g-
f+n+l)%100;
        e -= (b+i+f-h-d*k+g-b*d-
d+b*j)%100;
        a += (i+j+j*f+f+a)%100;
        h += (d+h+e+h-i+m-j-a+f-l+c-
b)%100;
        j += (l*j*f-l+g+a*b-k*i*b-n-
h)%100;
        e -= (b-n+c-c-f)%100;
    }
    break;

case 3:
{
    do
    {
        h += (j+n-a*f+j+h*e*m*f-k-
n-d)%100;
        l -= (b-c-m+l+b+i-c+j-c)%100;
        k += (d-f+j-
h+h+a+c+a+j+j+j+h)%100;
    } while( ++DOcnt[4]%5 );
    e += (e+n+h-i+l+b+a)%100;
    k -= (e+m+k+b-e-e+h-g-k+c-d-
e+c-i)%100;

```

```

    a -= (c+k-l-k+c*i*h+c-m+b+g+g-
a)%100;
    m += (f*j+f-b*e+l-l+l+b-j)%100;
}
break;

default:
{
    j -= (n+n-a+f-j*d+i-c+f-c-l-
c+i+j)%100;
    m -= (c-b-a*e+m*j*g-m-
n)%100;
    g = (c-b-h+g-b+e+c+a-g+i+n+f-
f)%100;
    j += (h-g+c*l-a-h-a-
k*l*a+n*j*f)%100;
    j += (n-e+h-e*f-c-j)%100;
}
}

j += (j+m*f+m)%100;
m -= (g*b+j-h-m*h+m-k-
a+j*b+f+l+k)%100;
j += (f+n*d*f)%100;
}
n -=
(i*g*n+h+j+b*k+m+k*g*c)%100;
d += (b*k+k+g)%100;
l -= (m-j+j+i-e-g-g-c-i)%100;
} while( ++DOcnt[2]%5 );
m += (b+d-a+b)%100;
f -= (e-c+a+c-k-b-a)%100;
j += (d-f-m+n-l+l+j+l+d-a+a*j-e-
e)%100;
g += (a-l*n+j+l+l+m+m-n+f+n-
a*l+k)%100;
}
f -= (k+e)%100;
c -= (j-k)%100;
m += (a*a-j-j+k+e*c-e)%100;
e += (b+n*m-e+b-e+d*d*c)%100;
h -= (d*e-d+m*h+d-k-a)%100;
j -= (n+g+g+l-n-m*h+g+m)%100;
e -= (j-m*m+g-a-e-m+l+i+f+n+k-
k)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F3(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[4]%10 )
    {
        k += (f-m-c+d-f-e-b+n+m-
d+g+k+i)%100;
        for( ; ++FORcnt[10]%5 ; )
        {

            switch( ++SWcnt[2]%4 )
            {

                case 1:
                {
                    if( ++IFEcnt[6]%2 )

```

```

{
    if( ++IFCnt[5]%2 )
    {
        i += (d*k)%100;
        f -= (n+d*g-m-f+g-c+f+d)%100;
        e = (a*e-a*j+n*h+h*d-e+k-i-
l+d)%100;
        m += (c*h-k+h+c*d-c-
j+n+l+b)%100;
        h += (a*b+g*d*d-b-f+j-n+d+n-
n+i+k)%100;
    }
    else
    {
        b -= (n-g*g-g*m-d+m-
d*e+d)%100;
        f += (i*l+d*a-g+c+l-g)%100;
        g += (e+d)%100;
        h -= (a-l*a-l-c*l-g+h-
h+k)%100;
        e += (n+g)%100;
        i = (h*h)%100;
    }
    while( ++WHILECnt[6]%5 )
    {
        c -= (a-h+m-e+d-l+k*l-f-
a*m)%100;
    }
    do
    {
        k -= (g-m*m-d+d+n)%100;
        j += (e+b-f*j*c-n+h)%100;
    } while( ++DOCnt[6]%5 );
    i -= (g+f*d+k-c-h-c+f)%100;
}
else
{
    while( ++WHILECnt[5]%5 )
    {
        n += (c+f-d)%100;
        e += (k-g-e)%100;
        k += (l+c+f-k)%100;
        i += (k+h)%100;
    }
    do
    {
        m -= (l-k+l*a+l+c-a+h-
i*g)%100;
        d -= (a-a+c-d-m+i+f+e+h-f-g-
f)%100;
        k += (k+a+b*h)%100;
        m += (f-n+m-l*l*a-f*n)%100;
        j -= (h-n*l-c*m)%100;
        a += (h*n*f-c)%100;
    } while( ++DOCnt[5]%5 );
    for( ; ++FORCnt[7]%5 ; )
    {
        d += (g+j+g-i+b-h-k-n*e+i-h-
m+d)%100;
        g = (j+m*k+l-c+j-
h+g+f+e)%100;
        h = (d-l-n+a-j+n*c-m)%100;
        h += (m+m+a+g+a-i-b+g*j-
e*l+h-e*j)%100;
        d -= (i+m)%100;
    }
}

```

```

if( ++IFCnt[3]%10 )
{
    g -= (b*i*m-k*e-f+d+d)%100;
    n -= (c*f*k+f-i+d+a-
j+l+g+i)%100;
    h -= (j*d+h*l)%100;
}
for( ; ++FORCnt[8]%5 ; )
{
    e -= (m*a-n+i+c-m)%100;
    h += (m-i+h+f*f*f)%100;
    f -= (i+m)%100;
    c -= (c-m-e+g-h*c)%100;
    i -= (b-m)%100;
}
for( ; ++FORCnt[9]%5 ; )
{
    switch( ++SWCnt[3]%3 )
    {
        case 1:
        {
            i -= (k+b*m-l+j-h-b-a-e-g+i-
b+d-h)%100;
            f -=
(h+b+b*c+j*g+b+h*h)%100;
            d += (c+e*a+g+l-n+n+k+g-n-l-
g)%100;
            c -= (d+h+e-g-i*j+f+g-k-e*j-
g+i+f)%100;
        }
        break;
        case 2:
        {
            e = (k*l+e+g+m+k)%100;
            m += (j-e)%100;
            b -= (m-g+i+m*c-b+e+g)%100;
            l += (k*k+l+c+d-a-c*f+l*f+b+b-
a)%100;
            a -= (g*h*j+b*b*l)%100;
            j -= (c+m-d-k-l-f*k-l+n)%100;
        }
        break;
        default:
        {
            b += (b-c-a-j+n-c*l-d+e+c+g*a-
k-i)%100;
            e += (f-b*a+n-n+k-c*k)%100;
            b += (k*l)%100;
            c = (h*l)%100;
            b += (d-c)%100;
        }
        c -= (i+b*h-n)%100;
        m += (d+g+n-j)%100;
        f += (a*k*b-e*k-l+k*a+k+d+d-
d)%100;
        e += (h*b+f-d+m-f+j)%100;
    }
    b -= (i*b+j-e-k-e-k+a*b*c)%100;
    d -= (h-h*n+m+l)%100;
}

```

```

    e = (i+f*k-h-e-e)%100;
}
break;
case 2:
{
    b = (i*g)%100;
    c -= (j+k*a-j+g+j+f)%100;
    f += (b+h-a+l*i+j)%100;
}
break;
default:
{
    h += (d+h)%100;
    h -= (m-e)%100;
    b -= (n+a*k-b-n-g-f+n-j)%100;
    l -= (k*i)%100;
    m -= (f+d)%100;
}
l -= (c*k-m+e-g-a)%100;
l = (h+d-m*c-a-b-i-k*j+k-i)%100;
g -= (a+a+b*m+l-n+c+g-
d+i+l+j)%100;
h += (a*h-i+n-i+a-k-a+n+c+d-c-
m)%100;
b += (h-n*k+d+h)%100;
}
m -= (f*e-h-b-e+n*g)%100;
f -= (j*d*h*g)%100;
}
f += (l+n*h)%100;
f += (d+n+d+k-i+b-m+f+e+a-h)%100;
f += (c+m*j+l-j-a-i+d*c+l*n)%100;
e -= (n-b+l+d*l-a-d+l-j+j)%100;
g = (l-d)%100;
i = (k+f*i-l)%100;
n = (b+n+d-d+g-k-j-n+k-
i*b+k*m)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}
int F4(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCnt[6]%10 )
    {
        if( ++IFCnt[8]%2 )
        {
            e -= (f*i-k-k+a-a-k+d+b)%100;
            c += (h-k-m-e*i*a*k+j+i)%100;
            h -= (i+b-l-i-j-m+a-j-a)%100;
            d += (k*k)%100;
            h += (m*c-a*h)%100;
            d += (c+b-j*d+n-g*n)%100;
        }
        else
        {
            while( ++WHILECnt[8]%5 )
            {
                do
                {

```

```

for( ; ++FORcnt[11]%5 ; )
{
    l -= (k*c+e+f-a+c+h+j)%100;
    k += (c*e*i-a-n)%100;
    c = (b*i*b-b*d-b-m+i*j-j-
f)%100;
    j -= (h+j+a+j+n-f*e+c-
k+c)%100;
}
for( ; ++FORcnt[12]%5 ; )
{
    j = (j+c-h+j-f+n-e*m-j-e)%100;
    g -= (e+i+l+g+n*b*i*c+f-a-
b)%100;
    g += (n+c-e-l+a-n*m-
m*m)%100;
    m += (c-i*k-a-a+i+l)%100;
    e += (j-b)%100;
    n += (c+e+b+g-
j+j+i+b+b)%100;
}
if( ++IFEcnt[7]%2 )
{
    c += (c+j*i*n-a+m+f+e)%100;
    n -= (f*m)%100;
    g -= (d+m-m+g-k-i-f+c*g-f+d-
j+n)%100;
}
else
{
    a += (b*h-b-g-
e+g+i+g*e+d+b)%100;
    a -= (l-g-n-i+b+n*e-
i*g+a)%100;
    h -= (i+l-c)%100;
    m = (i*b+g)%100;
    g += (b-j+e-e+k+c-j-j+d*d-
f*i*e+b)%100;
}
while( ++WHILEcnt[7]%5 )
{
    g -= (j*n-c*k-h+k)%100;
    a += (f*f*c*k-l)%100;
    k -= (d-f-d-h+a-a+h*d)%100;
    n = (h-d+j+a)%100;
    d += (f*d+d-d+l*j-h-g+j-n+f+n-
d)%100;
}
do
{
    a -= (f+c)%100;
    d -= (m*i+i+c+e*d+b+l*e*i-l-m-
e*i-l-f)%100;
    d -= (b-f+n*g-a)%100;
    j -= (b+f+n+j-h)%100;
} while( ++DOcnt[8]%5 );
} while( ++DOcnt[7]%5 );
if( ++IFcnt[5]%10 )
{
    e = (l*b-d-g*c-c+n-d)%100;
    for( ; ++FORcnt[13]%5 ; )
    {
        n -= (h-e*e-g-i+a+h*d+h-
m)%100;
        a += (n*d*f)%100;
        f += (h-j-k)%100;

```

```

    f -= (i*c-j+j-m-c+b)%100;
    k -= (c-b+c-l-e*a-g)%100;
}

switch( ++SWcnt[4]%4 )
{
    case 1:
    {
        b -= (k*n)%100;
    }
    break;

    case 2:
    {
        g += (e+d+n-e-m)%100;
        e = (l+b+i-i-n-e*e-h+l)%100;
    }
    break;

    case 3:
    {
        b -= (b+l*k-b-b-e-a*f)%100;
        b += (a-e+k)%100;
        l += (m-k*b+f+a+j+b-
l*h)%100;
        c -= (k-d+m*h+i-
h+g*a+m+a+c+g-m+i)%100;
        h -= (d+b*e+m+i*n)%100;
    }
    break;

    default:
    {
        l -= (f-g+g-c+g-e)%100;
        g += (f-e-h-m-j-d)%100;
        c -= (i*m*i*h-k-j+g+i)%100;
        n -= (n+b)%100;
    }
}

i += (n+l+d+m+g-c+e-n)%100;
n = (n*n*c-n+l+d+g*h-j+c*n-
a)%100;
c += (e+l+l*f-
n*j+e+k+e*e+b)%100;
}
g += (e-j*e)%100;
i = (f+m+b*g*b)%100;
k += (a+k+g+e-b+g-l+b-m-j)%100;
}
j += (g-f*f*j)%100;
f -= (g-g)%100;
j = (a*e-g+l)%100;
i = (n*e-n)%100;
n -= (e+j-e+h*i-l-h-l)%100;
}
i -= (k+f-j+b-n*e+l)%100;
g -= (a-k*g+f+j+d-h+j+i)%100;
d -= (d+e)%100;
}
h += (f-l+g+h+h-h-a+h-b-l+j-h-n-
g)%100;
m -= (i+k*b+e-c-h+l+e-l+k+i*c)%100;
j += (f+b-c-i*l)%100;
c += (i+h+i-a-i+j+n-g-i-l+f+b)%100;

```

```

j -= (j+a-n+h-g+i-i-n-l-l)%100;
i += (m+e-d+i*j-l-j-a+c*a+k-k)%100;
g -= (g+h+c-b*f-c+d-g+n)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F5(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFEcnt[10]%2 )
    {
        h -= (f*j)%100;
        a = (a-m+k-l+k+f*h+f-k)%100;
        d += (d+n-g-h+n-a-e*a+e-f*i*l)%100;
        e -= (i-b-f)%100;
        g -= (f*k*f+e-e*f)%100;
    }
    else
    {
        while( ++WHILEcnt[10]%5 )
        {
            do
            {
                for( ; ++FORcnt[15]%5 ; )
                {
                    if( ++IFcnt[7]%10 )
                    {
                        m = (h*d+f-b)%100;
                        g += (h-c-i+f-f*j-n+i)%100;
                        h += (g*d*m-d-k*i+a+j)%100;
                        d = (c-m*i-e-l-n-m*k+g+f-
d)%100;
                        k = (e-i*m-e-k-c*i+e+f+n+c-a-
c)%100;
                        l -= (l+f+k+j*d)%100;
                    }
                    for( ; ++FORcnt[14]%5 ; )
                    {
                        b += (d-j+d-g+f+l*e+d*j*e-
m*m*m+m+m)%100;
                        n += (e+j-e+g-n-c-e-
c*n+g+h+b-l)%100;
                        n += (k+l-n+h*i*g-j+i)%100;
                        h += (f-i-m-l-b-a+m)%100;
                        a -= (c-k+g)%100;
                    }
                }
                if( ++IFEcnt[9]%2 )
                {
                    d += (e-c+k+c+l-k-g-a-b)%100;
                    g -= (l-l*j)%100;
                    l -= (i-m)%100;
                    f -= (l*a*e-e+l+l+l*g-l-
i+b+e*k*f)%100;
                }
                else
                {
                    l += (m+n*a-n*a)%100;
                    i += (d+c+c-i*j+a+l*e-
g+d)%100;
                    i -= (n*f-b-c)%100;
                    j -= (k*b-h+b+b-f*n+b-
e+j)%100;
                    c += (m+b-m+c+e-l-g-a)%100;
                }
            }
        }
    }
}

```



```

    }
    break;

    case 2:
    {
        h += (f+m+g+h-k-a-
d+m+n)%100;
        l -= (h+n*c+h*i*n+h-c-f-
a)%100;
        h += (g+m-f+e-a+c)%100;
        h -= (m*k+h+a-m+e*i-d*g*a-
m)%100;
        l += (i+g)%100;
        b += (a+g+i-a-l+e)%100;
    }
    break;

    case 3:
    {
        b += (l*c-d+f*n)%100;
        n += (f*c+h-g+e*f-j)%100;
        b -= (d*j*e+f-c-j-g-b-i+d-d-c-
d*a)%100;
        g = (a+l*c-b-n+c-g+l+n+b-
h)%100;
        b += (k-m-e-j*f-d*a)%100;
    }
    break;

    default:
    {
        n -= (g*k*c-d*a+k+f+n-
h+b)%100;
        m -= (j+k-c+g+j-m-m-f-i-
e)%100;
        j -= (m+d+e+a+k+j+m-
e+l+f*k+i+b+f)%100;
    }

    e -= (k+a+f+c+i)%100;
    l += (a+e+h+m+m)%100;
    } while( ++DOcnt[12]%5 );
    f -= (b*b-b-d+c*g+l+g-j-n-b)%100;
    l += (a*h-e+g*j+d)%100;
    m += (l+d-i-k)%100;
    }
    n -= (i+h+f*n+h+h+d-d+j*j-j)%100;
    }
    k -= (l+l-n-n+h+d+d+d-d+j-b+k-
d*e)%100;
    c -= (n-i+m-b-k-k+k*k*i+b)%100;
    h -= (e-a-d+g-d*f-d+g+f)%100;
    b += (k*j-g*c+n)%100;
    h += (f+b-c+k*i+d)%100;
    n -= (j+f*g+k-i+n*c*b-i+c+e-b-
l+e)%100;
    k += (i*k)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F7(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;

```

```

a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
if( ++IFecnt[15]%2 )
{
    a = (l-i-g+n+j-j*k-d-f+a-m+c)%100;
    b -= (e+g-a*b*i)%100;
    m += (m-i-a*a-b)%100;
    b += (k+j+j-b+k+g*n+d-n)%100;
    b = (k+d+m-m*j-n*k+k-c)%100;
}
else
{
    while( ++WHILEcnt[15]%5 )
    {
        do
        {
            for( ; ++FORcnt[20]%5 ; )
            {
                if( ++IFecnt[10]%10 )
                {
                    f += (l*e)%100;
                    l += (d+b*i+k+a*d-
b+e+d*g*g)%100;
                }
            }
            for( ; ++FORcnt[22]%5 ; )
            {
                if( ++IFecnt[13]%2 )
                {
                    g -= (g+k-j+l+l-b*e+e+j-b-
m*b+b+i)%100;
                    f += (n+j*b-h-a-k+f*b-c+d-a-
m)%100;
                    d -= (k-j+d+a*f-k+d+l*i)%100;
                    b -= (a*i-l-j+m+d-i+l)%100;
                    g -= (j-h-c-j)%100;
                    c += (n*m-k+e*k*n*c-f)%100;
                }
            }
            else
            {
                f += (h-h-m+a*j+b*m+b-a-
j)%100;
                c -= (f+a+g-m-g-g-
m+a*c)%100;
                i = (l*h*a)%100;
                e += (m-d*d)%100;
            }
            while( ++WHILEcnt[13]%5 )
            {
                i -= (n*h*b)%100;
                f += (g+e)%100;
                e = (d-j+a-i-m-a-j*h*a)%100;
                g += (f-i+i-i-m-e-b*f*n+l)%100;
                f += (l-g-k-d-k)%100;
            }
            do
            {
                j -= (k*j+l+n+i-g+c-n*d+j)%100;
                b = (l*d+c)%100;
                l = (k*l+h)%100;
            } while( ++DOcnt[14]%5 );
            f -= (a-b-f+a+j+l+k+b-b-
f+g*n+j)%100;
            for( ; ++FORcnt[21]%5 ; )
            {
                c -= (m*k-e)%100;
                n += (c+k+b-a)%100;

```

```

                c += (n-l+j*m-l+m*e)%100;
                b -= (e*c+d+f*f-e+n-m+m-
l*h*i-l-h)%100;
                d = (k*m+g*k)%100;
            }
        }
        switch( ++SWcnt[7]%3 )
        {
            case 1:
            {
                if( ++IFecnt[11]%10 )
                {
                    l += (c*c-j*b+f-b+d)%100;
                    e -= (m+f)%100;
                    g -= (d+l-e+i*j+m+i+b-g+h-k-
l)%100;
                    f -= (i-h+c-j-
i+c+e+h+a+a)%100;
                    l -= (n-g)%100;
                    g += (h-i*g+j-f+h+m+e-f)%100;
                }
                if( ++IFecnt[14]%2 )
                {
                    m += (c*a)%100;
                    j -= (d-f)%100;
                    a -= (m-a+m-c-h+j-l*c+a*h-
h)%100;
                    b = (h+k-n+b-d+l*c+g-h+b-n-
m)%100;
                    n -= (l-i-j+k-e-d*e)%100;
                }
            }
            else
            {
                k -= (m+h*m+n+a)%100;
                e += (g+h+i-b*f+a+n-a-
d+k+i+h+b+h)%100;
                e += (n+d*f-j-h-l-b*a-c-f*k-
n+i+n)%100;
                g += (b-b-g-i+m-j-l*d-c-
j+l*a+d)%100;
                a -= (j*b)%100;
            }
            while( ++WHILEcnt[14]%5 )
            {
                e += (m+l+f*n*e-m-n-i*d-a-
k*j+h-g)%100;
                g -= (a-b)%100;
                a -= (f-k+h+f+c)%100;
                k -= (f+e-i+k+f-i-
d+i+b+h+h+b+i)%100;
            }
            do
            {
                l -= (d-a)%100;
                m += (n-a-c-i+h*j+n-f-k-
n+e)%100;
                m += (h-j+g)%100;
                c -= (c*j-b*d*b-n+c-f+b*b+j-
k)%100;
                m -= (e+f+h+c-f+n*c+a+d-
m+b-f-m)%100;
                i -= (j+d-i-n+f+g)%100;
            } while( ++DOcnt[15]%5 );
        }
    }

```

```

        break;

        case 2:
        {
            c -= (k-l-i*m+m+a+f-f-n-h+e-
d)%100;
            n = (k*k*a-l*k)%100;
            e = (h-e+h-j-a-
e+e+a+c+g)%100;
            d -= (n*i*g-d+g+h-a)%100;
            h -= (e+b-i-n-g-e-
d+f+f*a+d+e)%100;
        }
        break;

        default:
        {
            a -= (n+e)%100;
            n -= (b*i+d+e-m+i+c+a*b+i+g+j-
e)%100;
            i += (f+b+e-i-a-k+l)%100;
        }

        e += (k-e*j-l+n)%100;
        m += (a-j-n*n*g)%100;
    } while( ++DOcnt[13]%5 );
    k -= (f-b*f+b-a+f+j+c+e-f+c-a)%100;
    a += (n*e+g-j+i-j*l+n*g+f+e+b-
g)%100;
    l += (h+d)%100;
    d += (n*i+k)%100;
    c -= (m+k*c+k+e-i*m-m-e)%100;
    }
    a += (h+i-k+a-i+c*k-k*b+f+k)%100;
    d -= (g+g-l*i+g+f*e+b*f-g)%100;
    h += (c+d+g)%100;
    }
    d += (d+i)%100;
    a += (f-a+i-f+m)%100;
    k += (j*h+k-l-e-j*i+d)%100;
    l += (f+g-l*k-e-e*b+i-n-n-f+a-b)%100;
    h += (k-a-l+e+e-h-f-m-b)%100;
    m -= (m-n*l-f-n*j+e*i-
a*b*g+j*n)%100;
    b += (i+g-a-j-j-d+a-a*b-a-e-
g+k*i)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F8(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[26]%5 ; )
    {
        for( ; ++FORcnt[25]%5 ; )
        {
            if( ++IFEcnt[17]%2 )
            {
                m += (g-i+h+m-e+j-a)%100;
                h += (b+d-h*a*d-g+i*j+l*m)%100;
                i -= (f-l*m-k+g+e+h+f-g-f-i+h-h-
c)%100;
                l -= (c*k*i)%100;

```

```

            }
            else
            {
                while( ++WHILEcnt[16]%5 )
                {
                    do
                    {
                        j += (g*f-c+j+c)%100;
                        d -= (c+j+a-e-l-i+h*n+j+a-i+l-
b)%100;
                    } while( ++DOcnt[16]%5 );
                }
                if( ++IFcnt[12]%10 )
                {
                    m += (b-b*h-h-f+e*h*k-a-
f+e+b)%100;
                    for( ; ++FORcnt[23]%5 ; )
                    {
                        e += (d+k-c)%100;
                        e += (d-c)%100;
                        g -= (i-m*a*l-l+h-j+n*a-
k)%100;
                        h = (f+m+j*j+c+i-g-e+g-l+h-
a)%100;
                    }

                    switch( ++SWcnt[8]%3 )
                    {

                        case 1:
                        {
                            m -= (j-a-g+g+b-j-k)%100;
                            h += (m+i*k-f+c+f-c-e-l+e-d-
c+m)%100;
                            n += (b-e*d*f*k-i-m-f+c-
d)%100;
                            d = (f-j*e*h*h+e+g+f)%100;
                            f -= (a-l-g-e*i-c-m-m-h-e-k-
n)%100;
                            j += (l+c+h+b+c-j-l+n-
a*c+g+k*k)%100;
                        }
                        break;

                        case 2:
                        {
                            d -= (e-m+k+j+b+i+e*b)%100;
                            b -= (g-f+b*f+f+l-f-d*d*c+b-
h)%100;
                            c += (l+b+j*i*c*f)%100;
                            f -= (a-b)%100;
                            e += (n-k-i*n-b-n-l)%100;
                        }
                        break;

                        default:
                        {
                            d -= (a+i+f+d-n-l-e-n)%100;
                            a += (a+l-k-i*c+e-e)%100;
                            n -= (a-c-a*j-c-k+e)%100;
                        }
                    }

                    if( ++HFEcnt[16]%2 )
                    {
                        n += (a+d-h)%100;

```

```

                        b = (g+c)%100;
                        l -= (f-d-e-e+e)%100;
                        i += (b+h-k*j)%100;
                    }
                    else
                    {
                        i -= (c-h-l*d-j*b-c+a+c-f-
k)%100;
                        i -= (n-a-m*c-b+a+g-i-l+g*k+j-
l)%100;
                        a -=
(m*j*e+e*g*i+k+e+n+h+c+d)%100;
                        h = (a*g-i+e+c)%100;
                        n += (l-a-g*h*a-d-g*m-
k)%100;
                    }
                    while( ++WHILEcnt[17]%5 )
                    {
                        k -= (i+i+l-e-m-j+b+n)%100;
                        b -= (c*h-l+k*e)%100;
                        d -= (g+c*f*n+h)%100;
                        a += (h-a-d-m*d+h+h+k)%100;
                        h += (f-k+c*a-f-
e+d*i+k*m*c*a)%100;
                        e -= (j*c-
f+n+j*h+b+c+h+l+d)%100;
                    }
                }
                do
                {
                    for( ; ++FORcnt[24]%5 ; )
                    {
                        e = (f-m)%100;
                        k += (k+c+k-i+e+h+k*e-
i+m*h+c*a+e)%100;
                        h += (l-n+c-l+k*j+i-g+c*c+b-
n+g)%100;
                        e += (g+e-h+b+n-g*c*e-i-
g)%100;
                        d += (j-d-g+b)%100;
                    }
                    h += (a-h-k+f*a)%100;
                    b += (b*i-d+f-g*k*g-j-m-c)%100;
                    k += (d*f+g-k-h-d-d)%100;
                    h -= (h*h+c-l-b-k*b-i+l-
n+a*f*a+a)%100;
                } while( ++DOcnt[17]%5 );
                k += (e+k)%100;
                m -= (n-l*m-k*f-m+h+e+a)%100;
            }
            n += (k+e*c+c+h+g+d)%100;
            b = (d+b+e+b)%100;
            n += (a-e+g+g+m+c+c)%100;
            b += (d-n-e*c-n+e+f+h*j*d-e+e-
i)%100;
            m -= (m-i-g*a-f)%100;
        }
        g += (e+h+h*k-k+k*f*j*h+n-
a*l+f)%100;
        d -= (l+n)%100;
        h += (c-a-m+m+g*c+f*i)%100;
    }
    b += (f+n)%100;
    f -= (c+h+n+c+a*k-i)%100;
    i -= (b*h-c+d*n+g-g-a-i*f-h)%100;
    f += (d+d-f-c+k)%100;

```

```

l += (h-i-d-d-c-h+c)%100;
g -= (l-g)%100;
a += (b*a*d+i*j*m+i)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F9(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[14]%10 )
    {
        for( ; ++FORcnt[30]%5 ; )
        {
            if( ++IFEcnt[20]%2 )
            {
                k -= (b-h-g-n-b-d+e-c+i+j)%100;
                m += (b-e-j-h-j-b*i*g+m-
h+f)%100;
                h -= (c+b-i)%100;
                l = (n*k-b+j*i+c-h-g+j)%100;
                n -= (j+a-e-m-j*f*n-c+c-
l+f+n)%100;
            }
            else
            {
                while( ++WHILEcnt[18]%5 )
                {
                    do
                    {
                        g = (l+n+f-a-d*g*d+f)%100;
                        g += (b-e+f+h+g-e+j-
e+j+i)%100;
                        b -= (k-f-m-a-m+f-
m*n*f+k)%100;
                        f -= (g-l)%100;
                    } while( ++DOcnt[18]%5 );
                    c -= (l*b*g*d*d*l+f-g-g-
l+a)%100;
                    for( ; ++FORcnt[27]%5 ; )
                    {
                        a += (h-l+n*f+a+l+l-f*i-j)%100;
                        l += (l*c+a-k+i*b-e-j-m+f*c-a-
e)%100;
                        b -= (d+g-n-c-j+l-k)%100;
                        j -= (d+f*l+a+c+g-e-g-e-
m+g*b+m*h)%100;
                        i -= (l-f+c*c*b+c)%100;
                        l -= (d+b-c+k-n-f+l*a-d*b-
n)%100;
                    }

                    switch( ++SWcnt[9]%3 )
                    {
                        case 1:
                        {
                            d = (i*d*e+c-b-d-h+a)%100;
                            m += (k-g+i*j-l-
j*j+d*l+h)%100;
                            m += (d-d+b)%100;
                            f -= (l+k)%100;
                            m -= (f+a)%100;
                        }
                        break;

```

```

case 2:
{
    j += (g*g*m+d+m-n+h)%100;
}
break;

default:
{
    e += (i+f-h*m)%100;
    h = (k*j)%100;
}

if( ++IFcnt[13]%10 )
{
    f += (e*h-j+l-g+l+b-j-e+n-m-
e)%100;
    i += (d-e-b*c+b*l-a-a-
m+b)%100;
    l = (m+e-c-h+m-l+e-i-n-f-
c)%100;
    a -= (n+f-d*m-a+d)%100;
    k -= (j-f-k-e+h*m+e-
h+k*b+n)%100;
}
}
if( ++IFEcnt[19]%2 )
{
    if( ++IFEcnt[18]%2 )
    {
        g -= (d+j*m-d-h-
k+e*e+b+c+i)%100;
        a += (k-l*i*h-i+l-a+h-n+n-c-c+i-
k)%100;
        m += (l+g+l+a-i-k+a-e+f*k+h-
b-k-m)%100;
        b += (b-i-d+b+c-j+l+c-k*k-
h+l*g+m)%100;
        g += (n-m-e+g-l*e+j+e*h-h-
d*n-c)%100;
    }
    else
    {
        c -= (i*j+e*m-l+b*c*j+e-
l+e*i+m)%100;
        j -= (a+b)%100;
        n -= (d-i)%100;
        c -= (g-c-f*i-h+e*g-e)%100;
        j = (a-m-d-e*i+d-
h+i+j+b*f*h+h+f)%100;
        b -= (e+i-k-k-g-l*i*d+n)%100;
    }
    n += (d+l+c-e+b+c)%100;
    n += (i*f)%100;
    g -= (e*c+c-a-b-d*k-
e+k+i+a*e+n)%100;
}
else
{
    while( ++WHILEcnt[19]%5 )
    {
        m += (i-f-a+m+l+h-i+l+i)%100;
        a += (a*i*d+b-h+f*e)%100;
        l -= (f*g+g-b+d+m*n-j-
m)%100;

```

```

        m += (n+b*b*j+i+c)%100;
        l -= (j+a*d*n-f-b+c-m-
m*e+n*j)%100;
        j -= (k-i)%100;
    }
    do
    {
        n += (b-f*i*g+l-i-i*i*e)%100;
        j = (b-i*f)%100;
        b -= (f+c)%100;
        j += (l+l+d*m+c+d-
c+e+j+f*a+g)%100;
        i += (b*i-f-m+e)%100;
    } while( ++DOcnt[19]%5 );
    for( ; ++FORcnt[28]%5 ; )
    {
        a -= (l-n)%100;
        l -= (a+f+h-l-j*a+f-j*e*c-
m+i)%100;
        h += (a+j*i)%100;
    }
    for( ; ++FORcnt[29]%5 ; )
    {
        c += (f-b)%100;
        k -= (m+a+n-i+n+i-l-h*c-l-
n+k)%100;
        m += (c+e+c-n*m+h)%100;
        e -= (b+j-g+g+l+g+f)%100;
        m = (g+k+d+n+g+g+i+i+b*i+h-
e)%100;
    }
    }
    n -= (e*d-n+c)%100;
}
l += (e-j+j-j-g-a+a-n-f)%100;
b -= (k+c+g+l+a+g*i+g-
b*m+m+m)%100;
e += (i+n-g-m-b-m-f+m-d+e*f-
n+k)%100;
m += (e+e-h+n-k*b+j-d+b+j-e+c-
j+l)%100;
}
l -= (i+k*f-h)%100;
e -= (m+e+m+l+b+b+f+j)%100;
m += (n*d*a-k+k*c-l+l*l-d+i+n-
b)%100;
h -= (e-l+l-k-c+c+g*m-n+c)%100;
n -= (j*f-g+j-j-f-f+n*i*d-g-k-c-i*i)%100;
}
c += (c+l-k*f*f-h-a*d-h+b*n)%100;
h -= (e-l+k-b+i*g+a-i-i*j)%100;
j -= (h+i-l+d-k-h)%100;
l -= (c-b*n)%100;
g -= (n+a)%100;
n += (e*b+c-m-k-n*h)%100;
a += (c+n+c-n-g-d-e-h-i-i)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F10(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[21]%5 )
    {

```

```

do
{
    if( ++IFcnt[16]%10 )
    {
        n += (n+c-i+k-n)%100;
        for( ; ++FORcnt[31]%5 ; )
        {

            switch( ++SWcnt[10]%3 )
            {

                case 1:
                {
                    n -= (a+n-m*g+n+e*f*i-l-d+f+k-
h+d)%100;
                    l -= (g+m+f+m*g*i+i-f-
n+d)%100;
                    j += (b-f*g+b+a)%100;
                    m += (l-n+i)%100;
                    h += (a+b*d-d+d+d*m-
i+b*n)%100;
                }
                break;

                case 2:
                {
                    e -= (n*i*d*g*i)%100;
                    k += (a+l)%100;
                    i += (m+a-d-h+m)%100;
                    h -= (d-i+e+a-j+f-c+d+b-a-a-
h*a)%100;
                }
                break;

                default:
                {
                    c -= (b+f)%100;
                    m -= (h+l+i*g+c*g*d-m-m-i+e-
l+i*i)%100;
                    h += (i+n+m+a+h+d*j-l-
l)%100;
                    g += (f+j)%100;
                    m = (g+h)%100;
                    n -= (l+h+m*n*k-m+c+l+k+j*b-
g)%100;
                }
            }

            if( ++IFEcnt[21]%2 )
            {
                c += (m+f+l+a-i-e*g+j+i-
e)%100;
            }
            else
            {
                g += (l-f)%100;
                k -= (n+f+m+b-a*g-d-
l+k)%100;
                c += (e+j-i+g+e+k)%100;
                f += (k-h-c)%100;
                n -= (f-i+c+m*d*g-a*l+e+d*i-
k*a+a)%100;
            }
            while( ++WHILEcnt[20]%5 )
            {

```

```

                m += (m+j-a-
i*d+n+b+m)%100;
                g = (g-a+c*b+a-e*g*i-
j*i)%100;
            }
        }
        do
        {
            for( ; ++FORcnt[32]%5 ; )
            {
                l += (d-l+h+d*j+b)%100;
                c += (b-h+f+k+m-b+a-i+e*h+g-
l*h)%100;
                a -= (b+b-g*a-h+l+g+k+m-
f*a*a+k)%100;
                c += (e-n+e*l-f*m*d+j-
g)%100;
            }
            if( ++IFcnt[15]%10 )
            {
                l += (d-m)%100;
                n += (g*i+j+a+b*k)%100;
                m += (e+c-l-b*k-i+k+h)%100;
                n -= (c-l*m)%100;
                n += (j-m*j)%100;
                l += (i-a+f-m*j*f-i+d+l+b)%100;
            }
            for( ; ++FORcnt[33]%5 ; )
            {
                i += (f+m-c-n-c-h-n+c+d+f-n*j-
h)%100;
                a -= (j+b)%100;
                m += (e*m-f*n-g+b-
b+h+j+m)%100;
                m -= (f-b+d+l-n+a+j-n)%100;
                b -= (k+n-f-c+b-d-a*h+j)%100;
            }
            if( ++IFEcnt[22]%2 )
            {
                n -= (j+l)%100;
                b += (d-j+d*d+e-e*a+j)%100;
                i -= (d+m-h*k*i-a+m+m-l-k-d-
b)%100;
                a += (f*m+j-b+g-g)%100;
                b -= (i+e-n-j-f+h+e*c-i*n-
k+a+m+h)%100;
            }
            else
            {
                n -= (e-n+a+b+m*d-f+h-
k*f*k+f+m)%100;
                e += (l+b+d-m)%100;
                m -= (g*l-e+f-
n+m*f+b+g)%100;
            }
            c += (m-h)%100;
            } while( ++DOcnt[21]%5 );
            a -= (f+b*e)%100;
            c += (k+c+g+n)%100;
        }
        h += (h+h+c-m-g-h*b-m)%100;
        k += (f-i)%100;
        i = (h+m-n-j+n*l+f-m-f-d)%100;
        c += (h-k)%100;
        b += (k+h+l)%100;
    } while( ++DOcnt[20]%5 );

```

```

        f += (d-m-f-a+g-b-c)%100;
        e += (b+n-c+f-g-d*f*i-c-l)%100;
        g = (g*g*b+f-g)%100;
    }
    n += (a-i*f*e-f-f-j-n*e)%100;
    g -= (e-c-h*j-c+d+j+k-l*i+i+j)%100;
    g -= (k+j-l-e*i)%100;
    n -= (a+h+n+n*i-a+c-n-n*d-k)%100;
    d += (k*c-n+j*b+i)%100;
    n = (e+g-k*g-a-m-i)%100;
    i += (k*f+e+c+e+j-h*b+l-i+k)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F11(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[24]%5 )
    {
        do
        {
            n += (g+n-e+m)%100;
            for( ; ++FORcnt[36]%5 ; )
            {

                switch( ++SWcnt[11]%3 )
                {

                    case 1:
                    {
                        if( ++IFcnt[17]%10 )
                        {
                            l += (d-h-m*d-b+d+a+m+d*c-
a-g-m+b)%100;
                            b += (d-m-l*k-h+c-j-
n*a+e+i)%100;
                            n += (h*g-e+l-l-f-f-c)%100;
                            k -= (l+m-j+j+e+i-b)%100;
                        }
                        if( ++IFEcnt[23]%2 )
                        {
                            k += (a+j-f)%100;
                            k -= (b+m+h+e)%100;
                            b += (j+m-m+j-n+m+c+k-
g+h+c)%100;
                            d -= (f+f)%100;
                            e += (b-n-c-j+g-b+k*i*b-
m)%100;
                        }
                        else
                        {
                            e -= (m+c+m+n+f+m*k+h*f+n-
i*b)%100;
                            n += (b+m+f+i+h-j-
g+a*i+m+l)%100;
                            j -= (h-l+g+g+i*b+l+f*f+i+b-
h+g+m)%100;
                            b -= (k-i-d*j-l-l-f)%100;
                            b -= (n+f+m-h+d*n*k-
m)%100;
                            e = (e+n+l+k-k-b-h+k+n-b-g-h-
n-i)%100;
                        }
                        while( ++WHILEcnt[22]%5 )

```



```

{
    i = (n*j-g-k)%100;
    c += (d*d)%100;
    f += (e*m-f-e+f+i*i)%100;
}
do
{
    f = (k*l+m)%100;
    j -= (d+e+k-d-d-f-i)%100;
    b -= (a-f+f-e*e*f+g+j+l+e+d-
j+g-k)%100;
    m += (h+i-h-b-b-c*j+l)%100;
    g += (f*k+d-m+i+f+h-e-
e*g+l+j)%100;
} while( ++DOcnt[23]%5 );
for( ; ++FORcnt[34]%5 ; )
{
    a += (n*g*d-n*d)%100;
    e -= (m-e+j-l+f+i-
d*f+c+m*f)%100;
    h += (b+k-l-i*b+l+h+h-b-
n)%100;
    n = (g-k+g*g+k-l+b*k-l-
k)%100;
}
}
break;

case 2:
{
    for( ; ++FORcnt[35]%5 ; )
    {
        c += (c*m-f*f-k-h)%100;
        h -= (a-j+l-m*d+m+b+j)%100;
        j -= (l*h+b)%100;
        i -= (j-c+e-f-n-j+c-d-f+i-f)%100;
        h -= (l+b+k+m*e+a)%100;
    }
    if( ++IFCnt[24]%2 )
    {
        c -= (l-m)%100;
        a -= (g+b-d+k+j)%100;
    }
    else
    {
        a += (f*h*a+e-l-j-b-f-g)%100;
    }
    while( ++WHILEcnt[23]%5 )
    {
        h += (b-m+f*n+d)%100;
        l -= (a-c-m*g-e-d)%100;
        d += (e*n-i+l-g-f*a-c-f+i)%100;
        e = (c-h)%100;
        h -= (c-m+n-i-e-b+c+n)%100;
    }
    do
    {
        f -= (e*m*k+f+a-
b+f+c+k+j+h)%100;
        n += (b-c-l+i+i+d-c+a-
m+i)%100;
        n -= (e+g+d*d+k*g-m-n-l-c-h-
g+m)%100;
        l += (l-k-f-l+c+b-
i*g+m*b)%100;
    } while( ++DOcnt[24]%5 );

```

```

    if( ++IFCnt[18]%10 )
    {
        f += (m*n+a-a-h+b+i-c)%100;
        k -= (c+n)%100;
        g += (h-a-a-c+i-c-b-k-j-c*j-
f)%100;
        c += (d-a+i-h*j+a+g+b-e-
m+e)%100;
        h -= (n-k+i-j*j*l*e+b-a-
d+g+h+j+c)%100;
        k += (m-c+m-c-c-d-
b+i*m)%100;
    }
    l += (c-k+g-h+i-j*c-h*n)%100;
}
break;

default:
{
    d = (j+n+n*f)%100;
    i = (d-m*b+n*n+j-i-m-k)%100;
    c -= (k*n-n-e-k+k+j+b-j-l)%100;
    k += (b+m)%100;
    c -= (h+c+n+h-j-e-b*c+d-
h)%100;
}

j -= (k-i)%100;
h += (c+j-e-h*h-e+m-f+d)%100;
f += (h+n+f*f+h+k-g+g*l-n-h-
j+f+a)%100;
i -= (n*n-b-e+b+m+a+b-k-
k*f+l+h)%100;
}
e += (j+j-e+g)%100;
i += (b-f)%100;
a += (i-n+k*d+m+n*h*m*h)%100;
c -= (n-l*k*e*d+b-b*a+j+g-f-
l+k)%100;
} while( ++DOcnt[22]%5 );
n += (g+f)%100;
b = (e*c-a-m+f+c)%100;
j -= (i-m*n+f)%100;
}
g += (e-l+k-i-g-c-m+a-c)%100;
e += (k*j-n-i)%100;
n += (b-l+m-a-i-h+c-a+k*k*b-m)%100;
n -= (i*m*n-l+e*n-n*l+j+a-m-
j+a)%100;
f -= (f+d)%100;
f -= (m-j+l*f)%100;
n -= (j+d*n+h-l-c)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int main(void)
{
    int l;
    clock_t StartTick = clock();
    for(l=0; l<28; l++) IFCnt[l] =0;
    for(l=0; l<38; l++) IFCnt[l] =0;
    for(l=0; l<19; l++) SWCnt[l] =0;
    for(l=0; l<38; l++) WHILECnt[l]=0;
    for(l=0; l<38; l++) DOcnt[l] =0;

```

```

for(l=0; l<57; l++) FORcnt[l] =0;
long int sum=0;

sum += F1( ) ;
sum += F2( ) ;
sum += F3( ) ;
sum += F4( ) ;
sum += F5( ) ;
sum += F6( ) ;
sum += F7( ) ;
sum += F8( ) ;
sum += F9( ) ;
sum += F10( ) ;
sum += F11( ) ;

{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[39]%5 ; )
    {

        switch( ++SWCnt[12]%3 )
        {

            case 1:
            {
                if( ++IFCnt[25]%2 )
                {
                    for( ; ++FORcnt[38]%5 ; )
                    {
                        e -= (h*m-f)%100;
                        e -= (k*e*j-n-f+i-g-e+j)%100;
                        b -= (l+c-a-h+i+h+n-
e*j+i)%100;
                        l += (i-m+e-h+g-g+b-d+i)%100;
                        i -= (l-g*a)%100;
                    }
                    d += (i-m-c-a-j-l+g+d)%100;
                    n = (k+a-c-h+b-j-e*b)%100;
                    c -= (a*l-j*b*c+g*j*k-e+h-
n*d+e)%100;
                    l += (k+h+n-f+k*e*a*d-b-h-
b)%100;
                    n += (k+j)%100;
                }
                else
                {
                    while( ++WHILEcnt[25]%5 )
                    {
                        f += (a-f*m+c*h+b-j+j+e*d+m-
m-e+c)%100;
                        j += (a+d)%100;
                        m += (b*m+d-c+d+f+n+c+m-b-
g+k-b-a)%100;
                        d += (h-b-a*e)%100;
                        g += (n+m+l-e+k+n-
j*n+g)%100;
                        g -= (b-n-j*j)%100;
                    }
                }
                do
                {
                    g -= (n+k+e*h*j+j+g-e+h+c-n-
b-c)%100;
                    h += (j+k)%100;
                    i += (j*f+g+l+i*j)%100;
                    f += (k+l-b+b)%100;

```

```

        k = (l-b)%100;
    } while( ++DOcnt[25]%5 );
    for( ; ++FORcnt[37]%5 ; )
    {
        d -= (l-k+f+n*b)%100;
        g = (j+l*f+k*j-n-e-d-l+g-
k*j+h)%100;
        b -= (m*f-k-l-a)%100;
        e -= (n+e+j-h+h+k)%100;
        d -= (a-a+g)%100;
    }
    if( ++IFcnt[19]%10 )
    {
        f -= (i*c+l+d-i+m-f*g+b-
k+d)%100;
        e += (d-k+d+j-j*i*n-g*k-n-j-
n)%100;
        c += (d-a-c*i*f-l*h-
n+e+b)%100;
        l += (d+a*k+k*j-f-k+a+b-
g+b*h)%100;
    }
    }
    f += (h-j-h+g-i)%100;
    n -= (a+a+i-b-j-a+e-j+f*d*a)%100;
    h += (j*d)%100;
    f = (c-c+c-l+c+j+g*c)%100;
    }
    break;

case 2:
{

```

```

        l -= (h-l*g+h+f*d-e-j+n)%100;
        j -= (a-g+d-l+i*a-n+b-a+g*-
n*d+c)%100;
        n += (h+a*n+a+a-l+n-i-f-h)%100;
    }
    break;

default:
{
    k += (k-n+b+l-n*h+g-
b+k+i*b)%100;
    h -= (i+n*e+l)%100;
    l -= (k+e-h-l-l*a*k-e-m-j)%100;
    a += (d+k+j)%100;
    b -= (n-c+f-g)%100;
}

g += (i*c+j-f-a-k-c)%100;
k -= (k+d+k+c+b*a+k)%100;
}
d -= (k+e*k-k)%100;
c += (b*i+c+b-i-m+i)%100;
l += (c-b+b*e+b-c+j*e)%100;
i -= (l-b+l+k-j-n+h+m-d)%100;
j += (n+d+c-a-a-j-l-m-f-d*e+j+i)%100;
d += (f-d+a-j*e+i*m*-
g*n+d+a*h)%100;
n -= (m*l+c-k+k-a+d-m*j)%100;
sum +=
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

    cout << "\nChecksum = " << sum;
    for(l=sum=0; l<20; l++) sum += lFcnt[l];
    cout << "\nIF frequency:    Static = "
<< 20 << "    Dynamic = " << sum ;
    for(l=sum=0; l<26; l++) sum +=
lFEcnt[l];
    cout << "\nIF-ELSE frequency: Static =
" << 26 << "    Dynamic = " << sum ;
    for(l=sum=0; l<13; l++) sum +=
SWcnt[l];
    cout << "\nSWITCH frequency: Static =
" << 13 << "    Dynamic = " << sum ;
    for(l=sum=0; l<26; l++) sum +=
WHILEcnt[l];
    cout << "\nWHILE frequency: Static =
" << 26 << "    Dynamic = " << sum ;
    for(l=sum=0; l<26; l++) sum +=
DOcnt[l];
    cout << "\nDO frequency:    Static = "
<< 26 << "    Dynamic = " << sum ;
    for(l=sum=0; l<40; l++) sum +=
FORcnt[l];
    cout << "\nFOR frequency:    Static = "
<< 40 << "    Dynamic = " << sum ;
    cout << "\nRun Time = " <<
double(clock()-
StartTick)/CLOCKS_PER_SEC << "
sec\n\n";

    return 0;
}

```

## c.) BM1BMProgram8

```

#include <iostream>
using namespace std;

#include<time.h>

int
IFcnt[38],IFEcnt[51],SWcnt[25],WHILEcnt[51],DOcnt[51],FORcnt[76];

int F1(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[3]%5 ; )
    {
        if( ++IFEcnt[2]%2 )
        {
            m += (l-c+j-j-l-l+a-e-l)%100;
            b -= (m*f)%100;
            j += (n+i+k)%100;
            d = (m-h+l-f+a-d)%100;
            f -= (l*n+k+k*i*j+k-h-c+g+c-i-
f+k)%100;
            c -= (c-c*g)%100;
        }
        else
        {
            while( ++WHILEcnt[1]%5 )
            {
                do

```

```

                {
                    if( ++lFcnt[0]%10 )
                    {
                        k += (l-e-j-k+f+h-j+f*c-b*b-
d)%100;
                        n += (h+c-d-
b+f+e+n+b*b+n)%100;
                        g += (a+h-j-h+m*h+k-
m+d*d*c+d)%100;
                    }
                    l -= (b*m-d-j-e*m*j-
l+d*c+e+k)%100;
                    for( ; ++FORcnt[0]%5 ; )
                    {
                        d += (d-c)%100;
                        n += (l+k-h)%100;
                        j = (m-j+a-n*g*f+c-g+h-
b)%100;
                        m += (e-a)%100;
                        f += (n+l+a-j+n-l*b+d+h+f+d-
n)%100;
                    }

                    switch( ++SWcnt[0]%3 )
                    {
                        case 1:
                        {
                            f = (d-d-d+c+e-h+h+g-
m)%100;

```

```

                            f -= (j+b*g*h+f*n-c*a+l)%100;
                            n -= (d+k-n+b-n-f-m)%100;
                            h += (l-e-m+b)%100;
                        }
                        break;

                        case 2:
                        {
                            a -= (j-j+l+d-a-i)%100;
                            b += (m-k+d+m+c-f+c-k*f+c+f-
n)%100;
                            g -= (k+g*l+n*e-a+n*m-
c+m+h*d+d)%100;
                            k -= (n-n+i+c*-l-e-j-h-m)%100;
                            j += (h-g-k)%100;
                            c += (f-g+m+k-l-l*b-l*-
d)%100;
                        }
                        break;

                        default:
                        {
                            n -= (c-j+b)%100;
                            k -= (e+c)%100;
                            h -= (m*a*b-j+b-c)%100;
                            n -= (m-k*i-f-h+m+h)%100;
                            b -= (i+i+l*c+b-b-c+l-
c+n+h+b*j)%100;
                        }
                    }
                }
            }
        }
    }
}

```

```

        if( ++IFEcnt[0]%2 )
        {
            d += (i+e+i+j-f*a*e+j-
g*j+i)%100;
            a += (f*f+m+j-j-e+a)%100;
        }
        else
        {
            d -= (m*h-a+i+d-c*c)%100;
        }
    } while( ++DOcnt[0]%5 );
    while( ++WHILEcnt[0]%5 )
    {
        do
        {
            j -= (f+m+f+h-m+f+g-c-l*m-
j*h+n-i)%100;
            k -= (i-l-c)%100;
            e -= (k-m)%100;
            c -= (j-c+k+j*l-k*g+j-k+g-m-
b+l)%100;
        } while( ++DOcnt[1]%5 );
        for( ; ++FORcnt[1]%5 ; )
        {
            m -= (l+k-h-b+l*g)%100;
            n += (h-i*g+m-k+e+b*h*f-d-
n)%100;
            k -= (n-n+l+j*a-c-f*b*e*n+n+f-
k-b)%100;
            j -= (g-b*c)%100;
            l += (a-j+g+c+e-e-g+c+b)%100;
            l += (d+g-h+g)%100;
        }
        if( ++IFcnt[1]%10 )
        {
            c -= (f+h-d+n*j+h*n+n-
m+b+a+m-c-i)%100;
            h += (n-c-k-e-c-d-e-a*g-i-
a*c)%100;
            h += (e-l*d+h+i-d-
l+h*j*n)%100;
            f += (d+j-k+n+k+a)%100;
            i += (i+f-k-g*h-c+f*a*l+b+a-
e+f-j)%100;
        }
        for( ; ++FORcnt[2]%5 ; )
        {
            n -= (g+d*j+f-m+a*e-
j+l*i+n)%100;
            i += (j+e)%100;
            k = (i*i+f+d+c+i+m+j-m+j-
g*b)%100;
        }
        if( ++IFEcnt[1]%2 )
        {
            g += (g+k*m+n+i+h-b*f+e-g-i-
d+d)%100;
            m += (a-c-b+d*d-m*a)%100;
            g = (l-f-c)%100;
            a -= (e-g+a-i-b-
h+h*e*j+h+e)%100;
        }
        else
        {
            l -= (e-b-j)%100;

```

```

            l += (e-g-c+b-a*e+f*h+f)%100;
            b = (n+n+h-a-m-e-b-
e+b+m*k+e-n)%100;
            h -= (i+e+g+l-d)%100;
            b -= (f*l+g*c*g-n+a-l-n-
l)%100;
        }
    }
    h -= (b*c-g+f+i-h+f*g-n-b)%100;
    e += (d+m+j+i-e-f+l)%100;
    b += (c*l+h+d+j+n-f-e*c+b+n-c-
c)%100;
    l += (g-n-b+g-f*n+f*g+l+m)%100;
    }
    k -= (k*k+e*g+j-c*h)%100;
    m = (i-d-a+f-e*i+l-k)%100;
    d -= (n-g*j-e-f-c)%100;
    }
    c += (g-l-h+g+d*i-h-b+a+n-j-g-
c)%100;
    k -= (f+l*l)%100;
    b -= (m+b)%100;
    c -= (c-a-i+n+c+n-n+m)%100;
    }
    g += (j-i*n+l*g)%100;
    m += (g-f-j)%100;
    e += (b+e-b)%100;
    h -= (a-h*l+k-c-d+i-n-e+n)%100;
    j += (n*h+l-e-n-g-a*i)%100;
    l -= (i+j+k+d*j+g*g-i+k-a-g)%100;
    d -= (c-i)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F2(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[4]%5 )
    {
        do
        {
            c += (e-h*g-h+d+n+c*b+m+h+n-
k)%100;
            for( ; ++FORcnt[6]%5 ; )
            {
                switch( ++SWcnt[1]%3 )
                {
                    case 1:
                    {
                        if( ++IFcnt[2]%10 )
                        {
                            m -= (b+a+j-f-n-l+k)%100;
                            c += (k+f*m-c+h-h-
a+l+d)%100;
                            n -= (f*l+l+f*l+g)%100;
                            g += (b-d+m-e)%100;
                            h -= (l*n-g+b-k-h+a*n-b-
e)%100;
                        }
                    }
                    if( ++IFEcnt[3]%2 )
                    {
                        k -= (c*j-a-h+d*a*a+k)%100;

```

```

            h = (k+c-k+i)%100;
            e -= (d+j-l+h+i-a+k-
l+i+a+a*m)%100;
            m -= (b*j-e-a*e-e-k)%100;
            b -= (m+f)%100;
        }
        else
        {
            i -= (f*e+c-k-j-f-a-l+b)%100;
            h += (b+c+l-g-e*c+d*h+f-d-
j)%100;
            i -= (l*b-k+g+b-f*m*n+g-c-f-
c)%100;
        }
        while( ++WHILEcnt[2]%5 )
        {
            d -= (c*c+h)%100;
            f += (i+d-g+d*m-g+n+e*i-n-
l+g-a)%100;
            i -= (b-a+j-h-k*f+a*b-
e+j+j)%100;
            i -= (j-f+k)%100;
        }
        do
        {
            f -= (j-h+i-e+e-j-f-e-c)%100;
            f -= (k+n-h+c*a+m-a*m)%100;
            k += (d+f*i)%100;
            c += (a+i-l-k*k)%100;
            j += (d*g)%100;
            l += (e-
i+d+a+e*i+l*k+m+h)%100;
        } while( ++DOcnt[3]%5 );
        for( ; ++FORcnt[4]%5 ; )
        {
            f -= (m-e*a-n-j-f+g*a+d-e*j-d-
m+d)%100;
            f -= (i+n*d-g-h*n)%100;
            b -= (d+c-n+j+d+m*m+j-e+g-
b)%100;
            e += (i+j+e*m+n-c+d+k*k-
m+j)%100;
            h -= (g-n+f-f*l*i-m+e)%100;
        }
        for( ; ++FORcnt[5]%5 ; )
        {
            n += (n+m+n-n-l-f+i*c*c+g*j-
a-k)%100;
        }
    }
    break;

    case 2:
    {
        if( ++IFEcnt[4]%2 )
        {
            a -= (e-d+j+m+j+i+g+a-n-l-
j*g*g)%100;
            e -= (k+b)%100;
            b -= (j+b+l+c-b+a*m-c+b-b-g-
l)%100;
            b -= (c+k)%100;
        }
        else
        {

```

```

        m -= (a*k+j-h+j-i+l-
c+c*k*i*b)%100;
        k += (c+k+c-g-n-f)%100;
        k += (m+c*b*i+a)%100;
        i -= (d*i+b+d-j+n-l*m+l-a+c-
l+g*c)%100;
        j -= (m-g*i-d*g)%100;
    }
    while( ++WHILEcnt[3]%5 )
    {
        d -= (n+i*b*g-l)%100;
        c -= (i+c-g-g)%100;
        l += (j+a+d+a+n-g*c-d-
n*n+a+c-b-m)%100;
        l -= (b*n-b+f+b*e)%100;
        g += (l+k+n-m-
b+m+l+d+a)%100;
        j += (a-i)%100;
    }
    break;

    default:
    {
        do
        {
            f = (d+a-m+e+a-m*m*j+m*e-
b+i*f)%100;
            m -= (a+d)%100;
            f -= (i+l+i*b-b-d*f-
h+l*a+b+d)%100;
        } while( ++DOcnt[4]%5 );
        d += (h+m+e-d+a)%100;
        m -= (d+c-j+n-m*n+g)%100;
        d += (m*j+i*k-i-g+b*c-
c+n+c+j*i)%100;
        h = (f+e*i+e)%100;
    }

    l -= (g+f-a-b+j-g*h-b+c+k*k-k-
e)%100;
    e -= (l-l)%100;
    f -= (a+m+c+c+d-g-d-k)%100;
    }
    n += (b*k)%100;
    e += (d+a*m-e-n-n+j+b-e-g-j-
d+d*k)%100;
    d = (d+n)%100;
    } while( ++DOcnt[2]%5 );
    l = (g-c+n-b*d)%100;
    e += (d+b)%100;
    a -= (e+n+l-d*e-b+h+g*j+a+d-k-
l)%100;
    m -= (h-e+e*b)%100;
    }
    k += (n-c+b-c*g-c)%100;
    d = (e+e-h+b-m*b*m*i)*h-
j+c*e+n+m)%100;
    f -= (b+n*i+i+g)%100;
    d += (n-b-a-f*a-i+c+f-a+a+i*m-m-
i)%100;
    f -= (b*g+k+m+l-d+e+f)%100;
    i -= (m-g-f+n-a*a-k+a-b*h*i)%100;
    f += (d-b+d+k)%100;

```

```

    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F3(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[5]%10 )
    {
        d += (f-l)%100;
        for( ; ++FORcnt[10]%5 ; )
        {
            switch( ++SWcnt[2]%3 )
            {
                case 1:
                {
                    if( ++IFcnt[6]%2 )
                    {
                        if( ++IFcnt[5]%2 )
                        {
                            b -= (i+n+i+l+g-d-k*i)%100;
                            b += (n*i-f)%100;
                            n += (d*i+j-i-g-i-l+b)%100;
                            f -= (j-b+f+c*f+l+g+m*c)%100;
                            l += (e+l+m+h+h+f+b-c-l-
d)%100;
                            b -= (b*d-n*l-n)%100;
                        }
                        else
                        {
                            i += (i*e+g)%100;
                            n = (b+a*i*d+f-a)%100;
                            d += (d-c)%100;
                            b -= (g+i*n-m-g)%100;
                        }
                    } while( ++WHILEcnt[6]%5 )
                    {
                        f += (g+f)%100;
                        h -= (e*h-e+j-j*h-k-i*g+i-j-
l)%100;
                        i -= (e-l*f*b+l-m-i+f-m)%100;
                        j -= (m-e+n+n-k+d*e+e)%100;
                        j -= (i+a*a)%100;
                    }
                    do
                    {
                        h += (c+g-f-m-m*a-j*d)%100;
                    } while( ++DOcnt[6]%5 );
                    b -= (a*b+i+m*b-b-a-i-i+j+m+g-
i+k)%100;
                    for( ; ++FORcnt[9]%5 ; )
                    {
                        b += (n+g)%100;
                        m -= (j+f-i+m+h-b-c-m+b+b-
f+k-c)%100;
                    }
                } else
                {
                    while( ++WHILEcnt[5]%5 )
                    {
                        d += (l+f*m*n-d-m*i+b*a-a-b-
h)%100;

```

```

        e -= (i-e-k-g+e+b-i+k-g-m-g-
j+j+f)%100;
        d += (d-a-m)%100;
        k += (b-d-i-k-m+a-
m+c+i*m)%100;
        e -= (m-k-l*f*b+b*j*f-
e+m*c+n+f-m)%100;
    }
    do
    {
        m += (e*e-a)%100;
        i += (e-g-i)%100;
        n -= (b-g-l-j*f-a+i-b-i-a-l*b-
i*d)%100;
        j += (j-e-l-d-b-d-b-f-c)%100;
    } while( ++DOcnt[5]%5 );
    for( ; ++FORcnt[7]%5 ; )
    {
        c += (d+g+m+m-h-e+m+m-h-
d)%100;
        n += (j*d+d-h-
b*m+g+f*e+i+a-d*m)%100;
        f += (l+g+e+c-d)%100;
        c -= (f-c-c-l+d+b-h*k-g-m+d-
k+g)%100;
        k -= (k+d-n+b+c+e-j)%100;
        l -= (e+h-d+g*k+b+l+e+f+c*l-
m)%100;
    }
    if( ++IFcnt[3]%10 )
    {
        i += (k-
a+c+c+n*e*a+b*h*e+k)%100;
        f -= (k-f-l+g+e+n+i*n-k-i-
i)%100;
        n = (d-g-j+d-m*k)%100;
        n -= (l+f)%100;
        m -= (h+d*d+c+h+h)%100;
    }
    for( ; ++FORcnt[8]%5 ; )
    {
        m -= (g-f-e-j-b)%100;
        k += (g-f)%100;
        d += (d*c-i+e-k)%100;
    }

    switch( ++SWcnt[3]%3 )
    {
        case 1:
        {
            if( ++IFcnt[4]%10 )
            {
                i += (j-g)%100;
                m += (i+i-a-a-k-l+m+f*m+c-
d+n)%100;
                g -= (h+e+d-i+h-h-a-h*e)%100;
                g += (g*k*d+m)%100;
            }
            if( ++IFcnt[7]%2 )
            {
                e -= (c*i+f*m-d)%100;
                b = (d-c-f-d-j-l-d+e+b-
g*b)%100;

```

```

        c -= (j+c-d-h*k+n*m-
j+a+i+k*k)%100;
        i -= (i-j-e+i-b-
j*g+f*m+j+n+j)%100;
        l += (j+b-d+b+e)%100;
    }
    else
    {
        d -= (c+g-a*i-g-l)%100;
        c = (l+d-d-m-h-h*e+h)%100;
        c -= (g-m)%100;
        g -= (n*j-l+j-l-
d*i+a+j+l*f+b*c)%100;
        f += (k-f+a+f-f-g-j+n-d*g-j*l-b-
g)%100;
        c = (n-e-l-a-n-i*n)%100;
    }
    e -= (h+m)%100;
    b -= (i+e-a*e+l+i-j*m-b-m-j-l+f-
g)%100;
    h -= (c+k-e+c*d-k+n)%100;
}
break;

case 2:
{
    c = (e*c+a-e+i*f)%100;
    f += (h-f+j+i-h-c+d-f-b)%100;
    k += (e-n)%100;
}
break;

default:
{
    i += (l-l-l+d-k-a-g*a-a-k-b-
g)%100;
    k += (a*c-k*g*a*m-f)%100;
    i += (k+c)%100;
    a -= (d-h*a-a*j-l+a+k-e+m+c-
n*h)%100;
    i -= (n-a+a+a+h+h-i)%100;
}
}

    n += (k*g*b+c*f-j-a*j-g-
e+i+k*f+h)%100;
    k += (f+m-d-e+e)%100;
    j += (a-b+d+l*c-k+j-a*k)%100;
    e += (f*m+h*j-n-l-c+l*e-c+k-
l)%100;
}
break;

case 2:
{
    e += (e-e*m+h+b+m-f)%100;
    j -= (l-c+m*l-
e+j+l+k*h*i*n+e*n)%100;
    j += (d+n-n)%100;
    n -= (h*i*i+g-k-
c+h+a*c+n*i)%100;
}
break;

default:
{

```

```

        l -= (m+j-b-e-i)%100;
        m += (k-e*c+k-k*e*n-n-l)%100;
        i -= (e+f*a+g-k-a*g-f)%100;
        b -= (f*e-h-f+j+e+e)%100;
        c += (l+k*b+a-g*i*l)%100;
        c -= (h+d+g+f*l+g+c*k-a)%100;
    }
}

    b -= (f+f-g+g+e+j+l+i-a+c-e-
e+n)%100;
    a += (e-c)%100;
    l -= (i-c-l+d+k-i-h*g*e-c-l-l)%100;
}
    e -= (d-l*l-k-a+k)%100;
    i += (d-n)%100;
    j -= (k+h-a+l-c-b)%100;
}
    n += (k-m+e+n-l-k-h-f-e+g*j)%100;
    c += (a-j-f*m-
k*k*m*e+m*m*g*b)%100;
    d -= (n+h-c*b-l-f+g+k+m+g+h-m+e-
j)%100;
    e -= (g*b+c*e+b-a-a*g-a-h-
f+g+m)%100;
    j += (a-m+k+i+i-a+i+a-b*l)%100;
    i = (b-i+d+h)%100;
    h -= (c+j+e+h+e)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F4(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[8]%5 )
    {
        do
        {
            for( ; ++FORcnt[13]%5 ; )
            {
                for( ; ++FORcnt[12]%5 ; )
                {
                    if( ++IFECnt[8]%2 )
                    {
                        n += (m-f)%100;
                        l -= (g+k*j+k*a-j-g*l*g+h-h-
f*g+f)%100;
                        b -= (n+f+f+i-a+m+e+e+j+m*i-
c*j+c)%100;
                    }
                    else
                    {
                        f += (f-j-k-j+b+g-b+h*i*k+a-g-
j)%100;
                        k -= (h*d*j+b+e-h)%100;
                        m -= (h-j-c+d-c+l-
j+d+i*i+d*l+h)%100;
                        m -= (h*d-k*j-e-d+m+h+f-g+h-
c-n)%100;
                        l += (a+c*c+g-c*m+f)%100;
                    }
                }
            } while( ++WHILEcnt[7]%5 )
        }
    }
}

```

```

        f += (n+d-f*d+k*g*l+j-i+f-l-l-
b)%100;
        f = (c-e)%100;
        a -= (c-m*k-i-
a*d+e+f+j+l)%100;
        j -= (m-k-d-e+n-l+m-f+d)%100;
        m -= (k*n+n-g+m*i-
m*d)%100;
    }
    do
    {
        f -= (l-j+e+k)%100;
        c += (l-m+c+l+c+l+g-c-
b+m)%100;
        a -= (c-i)%100;
        e += (d+b+n+g*n-l+h*i-d-b+k-
g+i)%100;
    } while( ++DOcnt[8]%5 );
    if( ++IFcnt[6]%10 )
    {
        h -= (m-i*i+f-i*b-b+k-e*b*a-
n+i)%100;
        i += (k+a-d-a+i*a+m+h-
b+d*m+l*f)%100;
        a -= (m+e*m*b-j-l+h+m*j+b-
b)%100;
        i += (l-j*m+l-l-b*g-k+c+l-
c*l+i)%100;
        c += (l-l-k+m-j-
n+j+m+n+f)%100;
        a += (k+i+m-c-g+f-l+l+c-l-
l)%100;
    }
    h -= (i-n+b*j*g+i+e+e*i+l-
a)%100;
    for( ; ++FORcnt[11]%5 ; )
    {
        c += (i+n)%100;
        g -= (i*n-m+l-f-a+i+c+j-
h)%100;
        g -= (c-c)%100;
        j -= (j+g+b+i*g*b*m+b*f-d-
a*n+b)%100;
        j += (l-l-d+n-b)%100;
    }
}

    switch( ++SWcnt[4]%3 )
    {
        case 1:
        {
            f += (l*a+d+n+h-k+e-l-g+h-
f)%100;
        }
        break;

        case 2:
        {
            l -= (k-d+b+f+a+k+h-j-g+c*j-
c*n)%100;
            i = (m*j+l-g+j+h+l+g-g-i)%100;
        }
        break;

        default:
    }
}

```

```

{
    a -= (e-g+i-f+j+i+b-
k+j+n+m*c+k)%100;
    g += (e+c*m-i-h-
b*j+k+b*j+i*m)%100;
    c += (i-k-f-d+a+e-d-k-
g*a*l+d)%100;
    a -= (n+g+b-h-e-a-i-m+b-a-g-
k+b)%100;
    g += (a-a-l*m-i-e*g+j*i+d-e-
n)%100;
}
}

k += (a-k-c-d+g+d)%100;
l -= (b-b)%100;
}
m += (k-c*j+d)%100;
l += (f*j+f)%100;
b = (m-f-h-i+k)%100;
f += (k*k+k-e+l+e+g+a*b)%100;
} while( ++DOcnt[7]%5 );
n += (c+b+a-e)%100;
a += (l*m)%100;
i -= (m+m-d-e+h-e-h-g-
l+b+g+g*n+j)%100;
a += (c-b+j)%100;
}
g += (c+d)%100;
f += (k*l-i*m+a+b-l*f*b-m)%100;
e += (a+b+d-e+f+m-a+e)%100;
d -= (e-f)%100;
d = (l-k)%100;
m = (a-e-h*j*k+m+j)%100;
f += (d-h)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F5(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCcnt[10]%2 )
    {
        a += (b*b*k*a-l)%100;
        c += (h+j-f+b)%100;
        h += (m-i*b-l-n-h+e-j+l+b+d+d+g-
n)%100;
        k -= (m-d+e*l-k+m)%100;
        m += (j+f+g*j*b)%100;
    }
    else
    {
        while( ++WHILEcnt[10]%5 )
        {
            do
            {
                for( ; ++FORcnt[15]%5 ; )
                {
                    if( ++IFcnt[7]%10 )
                    {
                        m = (d-f)%100;
                        k -= (j+e-e-m+c+f-b-l*c)%100;
                        j -= (i+k)%100;
                        h += (g+h)%100;

```

```

g = (d*i-k-e)%100;
}
for( ; ++FORcnt[14]%5 ; )
{
    a -= (g-j+c)%100;
    f += (k-c-l+g-c)%100;
    c -= (j-k+b+k-k+f+k+m-j-
g)%100;
    n -= (a+l+a-e+d+e+j+b-
b)%100;
}
if( ++IFCcnt[9]%2 )
{
    n -= (m+n-h-a-i-h+f+l+l)%100;
    d += (k-l-m)%100;
    m -= (b+f-i+h*i+k*d*b)%100;
    c -= (a-k)%100;
    n -= (n-i-m-d-m-m-n+b)%100;
}
else
{
    k = (i+m+f-b-e-d*l-e+d-
m)%100;
    k -= (f-a-k*b-f+c*h*j*n-i-m-
h)%100;
    c = (b+j+c-d*b-j-g*l-f+g-
g)%100;
    n += (h-d+d+b-j-e-l*a-b-c*e-
k)%100;
    g += (m-b)%100;
    e -= (n-f+c*j+h+e*e-f-c*e-
d+m-k+h)%100;
}
}
while( ++WHILEcnt[9]%5 )
{
    do
    {
        g -= (m+g+m-m-a*a+h*j+n-l-
e)%100;
        l += (h-l-i+m*e+b+f+j-a-
g)%100;
        g -= (b*a+n+k-h-f*j+k-c-
c*k+j)%100;
        a -= (e+e+n)%100;
    } while( ++DOcnt[10]%5 );
    h += (h-d-h+d+j-e-d+j+d*e+j-d-
j*n)%100;
    for( ; ++FORcnt[16]%5 ; )
    {
        i -= (f+k-f+g+m+i)%100;
        f -= (j+g)%100;
        h -= (g-m*g-k+c-k*b-g+l-l-
e)%100;
        c += (k-k)%100;
        a -= (m-k)%100;
        f -= (l-i*j*h)%100;
    }
    switch( ++SWcnt[5]%4 )
    {
        case 1:
        {
            b += (i-k*d-f+b+j-j)%100;

```

```

h += (g+m-e*k-h+a-k-h-e-
c)%100;
f += (l*k)%100;
c += (a*f+k+b-e-d+d-i*n-j+k-
m*j-j)%100;
f = (f-b*k*k+k-a+f-l+f-l+e+f+g-
k)%100;
}
break;

case 2:
{
    c += (e+m*c+d)%100;
    l += (j*e*g-n)%100;
    d -= (c-f-l-j-l*m-a-n*a-
m*f+b+j-e)%100;
}
break;

case 3:
{
    b += (h+l-b+c+j)%100;
    l += (b+j*d+g+g+a-g-k+c-
e*g*f)%100;
    a -= (k-f*l+m*g+h-e)%100;
    d += (k*n*n*e*c+h-f-
h+m)%100;
    k += (h-n+k*k-f-e-a+g+l-c-h+c-
b-a)%100;
}
break;

default:
{
    n -= (m+b-e+c-h)%100;
    n = (f+b-d)%100;
    l = (n+h*c-c-j-e-
h+l+k+e*l*j+l+g)%100;
    e += (i-a+d-g-f-n-i-j-
i*k*k+c+k+g)%100;
}
}

if( ++IFcnt[8]%10 )
{
    g += (i+d+d-g+c-m+g)%100;
    l -= (k-a-m)%100;
    h -= (i*f*n-n+d-l+j+e*f*b-k-f-
d)%100;
    d -= (l-b+i+h-i+l+b+i*i)%100;
    k -= (b+l-b-d-e+f+b*d*d+a-
b*e+n)%100;
    b += (f-l*b*k-d+d+f)%100;
}
}
j -= (a+i-b+d-e+d-l-k*c+n)%100;
e += (b*b-m*m-f+b+j)%100;
b -= (k-f-e+a+f-b+e-h*n+g)%100;
} while( ++DOcnt[9]%5 );
a -= (l+f-b+g-a+i-m-e+g-f*d)%100;
m -= (i-e+d-l*m-m*a)%100;
m += (n+g*i+b)%100;
b = (b-a)%100;
a -= (f+b+g-e-c*d+g+j-g)%100;
}
a = (i+g+d-f-n+d+b-f)%100;

```

```

        b -= (h+g*n)%100;
        b -= (b+g-k-i+k-c+e+k-m+f-
c*e+d+e)%100;
    }
    h += (a-m-d+g-h+g+a+f+i)%100;
    e = (g+g-e+f-b+l+l-b+i-a-j-b*l)%100;
    k -= (i*e*b*b*b)%100;
    m -= (g+g*e+n+i+h)%100;
    a -= (m*g+d-h*n*k+a+g+b+c+g)%100;
    k -= (c+e-m)%100;
    l -= (g-n*g+l*f)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F6(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCnt[13]%2 )
    {
        m += (a*e+n+g-k+i-d*c)%100;
        d += (d+l-e*k-e+e+j)%100;
        k -= (b+n+g*c-b*n-d+i)%100;
        b = (d-d+i*j-m+h+a*f-b*f-b+d-
h)%100;
        j += (i-b+a*g+c)%100;
    }
    else
    {
        while( ++WHILECnt[12]%5 )
        {
            do
            {
                for( ; ++FORCnt[18]%5 ; )
                {
                    for( ; ++FORCnt[17]%5 ; )
                    {
                        g -= (j-m*j-b)%100;
                        i += (m+a*m+a*a+c+l+d-
b*g+h-i)%100;
                        e += (m-j*h-i+k-i-j)%100;
                        e -= (c+l+b-f-j-m-
m*m+j+i)%100;
                        g -= (f+k*i*i+h-d+b-e)%100;
                        f += (c+d+e+j)%100;
                    }
                }
                if( ++IFCnt[11]%2 )
                {
                    c -= (b-c)%100;
                    d += (g*j+i)%100;
                    b += (i-l+k*k-d-e*j*a*l-
h)%100;
                }
            }
            else
            {
                i += (a+h+b+b-k)%100;
                j += (n+h+i-a-e+e-l+m-
i+h+m*h-n+n)%100;
                n += (b-k)%100;
                h -= (f-l+n*f-j+f+g+g-l+j+k-j-
h)%100;
                j += (b-b+d-b*m-j)%100;
            }
        }
        while( ++WHILECnt[11]%5 )
        {

```

```

            i = (k-g-j+h+e+i*l*f-n-e*n-c-
l)%100;
            m += (j+e+l*k)%100;
            b += (e*j-f+b+n-m-d-a)%100;
            a = (b-n*f*m)%100;
            f -= (d+j-l+g*k-i+k-c)%100;
        }
        do
        {
            k += (c+k*g*l*m*k*f)%100;
            b -= (d-i*n+l-h+i)%100;
            c += (n+f+d-k*h*e-k*c*f-
c)%100;
            a -= (h-l*d*a-k*n+g+f)%100;
        } while( ++DOCnt[12]%5 );
    }
    if( ++IFCnt[9]%10 )
    {
        k -= (b-k*a+a-a*k*a-a-g)%100;
        for( ; ++FORCnt[19]%5 ; )
        {
            n -= (h*k+b-m-i*a)%100;
            d += (f+d*k*m-d*a+a)%100;
            h -= (k-i-j-d+f+d*g*f+g)%100;
            b += (c+g)%100;
            f -= (j+b-n-h*a+e)%100;
        }

        switch( ++SWCnt[6]%3 )
        {
            case 1:
            {
                c += (l*b)%100;
                a += (j+l+g-h+a-e-
e+f+c*n)%100;
                m += (j+l-g)%100;
                j += (h-m)%100;
                h += (b-a)%100;
            }
            break;

            case 2:
            {
                m -= (g+c*b-j+m-g-i+f*m-
e)%100;
                c -= (n+f+d-j+j-j)%100;
                j += (f+b)%100;
                g += (a-k)%100;
            }
            break;

            default:
            {
                i -= (i-b*b+b-f+l+h-m-
g+h)%100;
                e = (h+c)%100;
                n += (j-k-g+d*e*m-l+n)%100;
                i += (e-d-c+d-c-c-e+f+a-
a)%100;
                f -= (c-j-m+g*n-n*e-h+g-f+k-
k)%100;
                j = (i+k-h+b-c+k*g+c)%100;
            }
        }
    }

```

```

        if( ++IFCnt[12]%2 )
        {
            i = (c*k-n+c-b*k+e*i+a+g-
k)%100;
            f += (g+m)%100;
            m -= (g+h+b*a)%100;
        }
        else
        {
            h -= (d*e+j*e*l+a-m-
b+i)%100;
            i += (a+g*d+c+h*j-l*n)%100;
            a += (a-n-
d+l+k+b+m+i+n)%100;
            l -= (f+f)%100;
            n = (h+g-e-n-m*e)%100;
        }
        d -= (a*n+g-c+g+i+k+j)%100;
        c -= (f*h)%100;
    }
    b += (g+c+c-b+a-e+d+n-
a+h)%100;
    b -= (f*n+m*h*g*e)%100;
    c += (l-d)%100;
    } while( ++DOCnt[11]%5 );
    f = (g-k*a+n-e-f+m*j-d-c-d-m-
l+g)%100;
    }
    }
    e -= (c+b-e+k+b-n-j*a-k+m-e*f*k-
a)%100;
    n += (d-l*j+c+n)%100;
    c -= (b+i-k+c+a-n+a+l*h+j*e+h+b-
d)%100;
    h += (e-c)%100;
    f += (k*g*a+g*a+j+k-c+j)%100;
    k += (m*i*a*f)%100;
    m += (j*j-j-j-a-j)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F7(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILECnt[15]%5 )
    {
        do
        {
            for( ; ++FORCnt[22]%5 ; )
            {
                if( ++IFCnt[10]%10 )
                {
                    for( ; ++FORCnt[20]%5 ; )
                    {
                        h += (c-
g+h+b+m*d+h+g)%100;
                        k = (h+i-f*c+f+c*n+b-d-
b)%100;
                    }
                }
            }
            if( ++IFCnt[15]%2 )
            {
                if( ++IFCnt[11]%10 )
                {

```

```

l += (a+b+a-a)%100;
l = (f+g+a+b-m+m+n)%100;
f += (g+i+k+i+d-a)%100;
l -= (d-m*l-l-n*k-b+a-a-
i+l*l)%100;
l += (g+b-l*j-a-l-j+i+j-c-l+i-
f*e)%100;
}
if( ++IFECnt[14]%2 )
{
g += (n*m+k)%100;
h = (b*n-l)%100;
n -= (n-i-e+k-n*e-l+g+m-b+n-
i)%100;
l += (b*n-h+e*g+n)%100;
}
else
{
k -= (e+i*n-j+j-e-l+c+e+k*h-
m+h)%100;
l = (k-a*g)%100;
j -= (n*i+a+a+m+d-m-d-
g)%100;
k -= (m+d*b+m)%100;
m += (n-h+a-j)%100;
}
while( ++WHILECnt[14]%5 )
{
l += (j-i-m+n-l-m-i-e-j*n)%100;
g += (j-n-n*n-e+f*e+k-f-c*b*j-
c-e)%100;
b += (m+g)%100;
a -= (m-k-l)%100;
k -= (h*g-i-i+h+g+e-c*l-
e+k+b+e-a)%100;
d = (n-n-m*f)%100;
do
{
i -= (k+m)%100;
f -= (e+a+c-e)%100;
f += (m-j-m+j*m-a+d-d)%100;
a -= (h-m-m-n+b+c-n-
e*k+i*e)%100;
d -= (l+c*d+b)%100;
} while( ++DOCnt[15]%5 );
a -= (i*h-e*l+a-h+a+a-a-i-f-
b+k)%100;
d -= (g-l-h-i-l+f+c+k-c)%100;
}
else
{
while( ++WHILECnt[13]%5 )
{
l += (g*b*k+d-d*b+a+e-
g*k*m+d)%100;
c = (l-i)%100;
a += (l+g*l*b+e+a*d-n+f-
h)%100;
e -= (b-b-m-c+b)%100;
}
do
{
f += (g*l)%100;
n += (l+j-c-c+g+c+d-d+f-
h)%100;

```

```

c -= (d+l-b-c-i+a)%100;
g -= (h+i-m)%100;
l += (h+a+l-j+f*i-l*a-c*h)%100;
e += (f+m*l-i-n*i*i-b*h-
j+i)%100;
} while( ++DOCnt[14]%5 );
k -= (h-f)%100;
for( ; ++FORCnt[21]%5 ; )
{
f += (l*c-g-i-j+c+h+l-g*l-c-e-i-
i)%100;
i -= (k+m+b*h+b-e-h+a)%100;
a += (d+a-l+i*f*m-h+n-c*d-f-
a+k+i)%100;
f -= (d-d+k)%100;
d -= (i-e+c)%100;
}

switch( ++SWCnt[7]%3 )
{
case 1:
{
b -= (g+i*d*h+g+m*c-h-e-i-
f+c-m)%100;
h -= (k-m+g+f+n)%100;
e -= (l+j-c*a+m-n+b+k*d-
m+c)%100;
}
break;

case 2:
{
e -= (b+g)%100;
f += (d+f*e)%100;
l += (l-h+h)%100;
j -= (a-g*a+c)%100;
k += (j-l+i-k-f-a-l-d+c+j+h+f-
e)%100;
}
break;

default:
{
e += (c-m)%100;
k += (f-e+c-n*k*b*d)%100;
h += (k*m*i+l-l-h-l+j-k-
m*n)%100;
j -= (i*l*a-n*i-e-a+e*i+k-l-d-j-
j)%100;
}
}

k -= (j+a-c-i-l-e+l+c+e+c-
f+i+n)%100;
f -= (b+h+f)%100;
d += (e+j-g*f-m)%100;
}
l += (d*g+a+b+e-h*g+h+g+e-
d+g)%100;
n = (h-f*i-m-j*g+n*j*c-j+i+m-i-
n)%100;
j += (g-i)%100;
m += (m+e-i)%100;
c += (e-f+a-b+k-j+i*h*a)%100;

```

```

} while( ++DOCnt[13]%5 );
m += (a*f*k+f-k+j*d-g)%100;
a += (e+g+i-k+j-l*k-g*h*f-b)%100;
n += (f+a*g*f+l-h)%100;
}
b += (c+j-n-d-m*i+k-m*m+j+d+l-g-
d)%100;
k -= (e-a*k+j+b+a-k*l+g-c+m-h)%100;
f += (n*j*j+g*c+j-b)%100;
d += (f-j-l+f-h+e-c-i+m*i+h+i-
e+b)%100;
l -= (c-j)%100;
n += (i*n+b+a-j-g)%100;
d -= (f*d-e+a+l-d+c+e-i+m)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F8(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
for( ; ++FORCnt[27]%5 ; )
{
for( ; ++FORCnt[26]%5 ; )
{
if( ++IFECnt[17]%2 )
{
f += (h-k-f*m-d-d-g*c*b)%100;
a += (h-d-b+e*c-n-m-k-b+m-
j+i)%100;
b += (l-j-g-n+c-f*l+k-n)%100;
n += (f+m*i-n+d*h+l-m-i+f-
e)%100;
}
else
{
while( ++WHILECnt[16]%5 )
{
do
{
m -= (d+h)%100;
h += (k-h-j-c+d+h-e)%100;
c = (i-d+f-g-k+f-i)%100;
f += (f+e-n-c-a)%100;
h += (g-d)%100;
} while( ++DOCnt[16]%5 );
if( ++IFCnt[12]%10 )
{
h += (i*h*m*a*b*n-b)%100;
}
}
c -= (j-e*i*a+n-k+j-b-d)%100;
for( ; ++FORCnt[23]%5 ; )
{
a -= (j+d-m)%100;
b -= (c+b-b+k+m+c+b)%100;
}

switch( ++SWCnt[8]%3 )
{
case 1:
{
a = (k+l-l*f+a+a+m*m)%100;
j += (n*b-b-g+f+i-l-a-
m+g)%100;

```



```

        i -= (d-h-a+l*n+g+n-h-
g+d+m+a)%100;
        k += (i+e)%100;
        a += (j+h-l*a+e+h+m+e+c-b-
i)%100;
    }
    break;

    case 2:
    {
        e += (g-c+e-c-g-d-
n+k+g+c*m+a-m)%100;
        e -= (k*e*n-i-g*l)%100;
        j = (n-e+n+g*e-i*f*j+a+e-m-n-
k-h)%100;
        f += (m+m-
h*m+b+h*j*d)%100;
    }
    break;

    default:
    {
        g = (d+g*n-i-a-h-i*m+b-f-j-
h+g)%100;
        g -= (c-n-a)%100;
        b += (h-g+d-g+k-i+f)%100;
        g += (g+b-d-m+f+h+h)%100;
        m += (g+m-e*d-g+f-
g+h+i)%100;
        b = (n-k+h+i+k-m+e+k-
d)%100;
    }

    if( ++IFCcnt[16]%2 )
    {
        h -= (g+n+a-g)%100;
        n -= (f-a-c*e-a-h+e-d)%100;
        m += (a+d-c-c-g)%100;
    }
    else
    {
        i -= (c-j-d)%100;
        m += (d+d+m-n*e-d)%100;
        c -= (e+a*g+a*j)%100;
        m += (e+a)%100;
        d += (l+i-e-b+j-h+l+f+k+j-
e*k)%100;
    }
    while( ++WHILEcnt[17]%5 )
    {
        do
        {
            c += (l-a+l-a-i-d+l+d*h-c+c-
g+l+f)%100;
            l += (e-m-j+b+j+b-e+l+c+i-g-
c)%100;
            m += (b*f*k+h+m-g*d-l-
g+g+l*k)%100;
            j -= (l+i-d*h)%100;
        } while( ++DOcnt[17]%5 );
        for( ; ++FORcnt[24]%5 ; )
        {
            h -= (h+k-j-h+e)%100;
            h -= (c+h+l)%100;

```

```

        l -= (k+h+a-f+l-f+a*e+h+h-c-
d*d)%100;
        j += (d+l-a+l-n-f-
e+k*j+d*f*h+i+c)%100;
        k -= (a+i+i*e)%100;
        e -= (c+k*l+m*h*c)%100;
    }
    if( ++IFCcnt[13]%10 )
    {
        b -= (f-h-e)%100;
        l += (j-k+i*m*k-i-f+e-
i*k+c+d)%100;
        m -= (g-f*b+h+f+c+j*k-c+m-c-
a*e+j)%100;
        f -= (i+a-c+c-i-f+i-c-l*e+c-
f)%100;
        m -= (g-l)%100;
    }
    for( ; ++FORcnt[25]%5 ; )
    {
        b -= (g*l+h-h*h*m-d)%100;
        m -= (g+m+m)%100;
        b += (h-m-i+k)%100;
        d -= (g-c*k*l-e*e-h-e+n-
k+h)%100;
        f += (i-k+i-h+h*f*h-j-
j*n+b+j*d)%100;
    }
    h += (c-m+e*h*f+f+a)%100;
    }
    m -= (g-b+e+n-f+b+m+f+h)%100;
    j -= (k-c+e-m-d+k+j+a-h+a*g-n-
f)%100;
    }
    m -= (c-e+b-a-b-h+b+l)%100;
    g += (e-l+b-l*k)%100;
    g += (a+b+f-k-h)%100;
    e -= (g-i-j*f*i*b-n+m)%100;
    }
    i += (c-d-l-i*m-j-f-e*d)%100;
    g += (b*m+j+g)%100;
    }
    n += (b-d+m-c-e+c+n-c-
l+m*f*c+n)%100;
    d += (n+c*g*i)%100;
    e += (l*e+e+j+l*m*e+n*d-
g*c+g+k+b)%100;
    e -= (b*n+m-g-k-j+c)%100;
    n += (h+n+d-b+i+b-c+a)%100;
    k += (l+l-m+j*m-b+h-g+n*a-j-
b+a)%100;
    f += (l+f-c+b+d+e-j-n*c-a-g+a+f)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F9(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCcnt[20]%2 )
    {
        f += (d+f-d+g*n-c+l-a+g+n*f-
b*f)%100;
        h += (m*i-k-j+j-i+c-d+l*i)%100;
        c -= (e-d+n+b)%100;

```

```

        a += (c*b+n+g-i-n+k-n)%100;
        h = (j+j)%100;
    }
    else
    {
        while( ++WHILEcnt[20]%5 )
        {
            do
            {
                b -= (e-l*b*h+j+a*h+i-h-
n*k)%100;
                for( ; ++FORcnt[28]%5 ; )
                {

                    switch( ++SWcnt[9]%3 )
                    {

                        case 1:
                        {
                            m += (k*n-f+c-g+m+f)%100;
                            j += (h-a-b-f*h)%100;
                            c -= (j+f*n*g-l*m-e-
l+g+a+d)%100;
                            k += (b+h-i+b+f+c-b-c-k-b-c-
c*l-i)%100;
                        }
                        break;

                        case 2:
                        {
                            n += (g+a+g+b*i)%100;
                            c += (k-g-k-l*k+l-j-e-j*d*e*g-
l*k)%100;
                            g += (e+h+j-d-j-n-f*a-e+n+i-
b)%100;
                            k -= (l*g+f-j-c+g*b-
m*n*n)%100;
                            a += (n+n-c-c)%100;
                            c -= (k+a-i+k)%100;
                        }
                        break;

                        default:
                        {
                            l += (n+a+l-f+h+m-
m*g*c*g+c*l+g)%100;
                            b = (f*i-h-k+a*m-c-f+f-i-
m)%100;
                            k -= (m-k+h+a-i+n)%100;
                            j += (b-n+n*d*j+l+l*m-a-l-c+a-
g)%100;
                            h += (i*l+g*j+h*d-f+j)%100;
                        }
                    }

                    if( ++IFCcnt[14]%10 )
                    {
                        j += (l+l*g+d+a+a-a+m)%100;
                    }
                    if( ++IFCcnt[18]%2 )
                    {
                        c += (c+f)%100;
                        h -= (a+f+k+l)%100;
                        j -= (a-m)%100;
                        m -= (l+g+e-b+e-e*m+h+j-k-l-l-
k-h)%100;

```

```

        c -= (n*n-g+m)%100;
    }
    else
    {
        i += (h-f+h*h+d+a-c+i)%100;
        j += (i-k+d+i+b-f-i+g*f+j)%100;
    }
    while( ++WHILEcnt[18]%5 )
    {
        l -= (k+l*i)%100;
        g += (c-h*f+a*i+k)%100;
        f += (g-i*i-c+l*i+k+h-b*e-
j+c+g)%100;
        n += (m*f+n-i-n+i+c)%100;
    }
    do
    {
        l += (e+l-f+k+f-j)%100;
        l -= (e+m+j-k-n-e)%100;
        j -= (e*e*a*b*a*m+c+j+j+n-f-
f)%100;
        m += (a+k-c)%100;
        k -= (h-n-a+h+m-f-n)%100;
        f += (a*d+g*i+l+n-h*i+j+b-
i*j+i+e)%100;
    } while( ++DOcnt[19]%5 );
    }
    for( ; ++FORcnt[30]%5 ; )
    {
        for( ; ++FORcnt[29]%5 ; )
        {
            e += (i*d)%100;
            f -= (k+i+c-k*j*j-
a+d+a*f*n+d*a-c)%100;
            a -= (d-g-l-h*a+b+i-l)%100;
        }
        if( ++IFcnt[19]%2 )
        {
            a += (g-k-a-l-e-l-m)%100;
            h -= (i+e-d-f*b+j+k)%100;
            i += (k+d-b+d*h-g+d*n+d+n-
a)%100;
            g -= (e+k*j-f+b+l*c+c+c-
h)%100;
        }
        else
        {
            a += (d-i*h+a+a-c*m)%100;
            e -= (h*b*h*j-h-m-n*j-l-f-e-
c+b)%100;
            e += (b+b-c+n+n)%100;
            g = (b+a-d+i*h)%100;
            b += (j-b-j)%100;
        }
        while( ++WHILEcnt[19]%5 )
        {
            a += (f-e-d)%100;
            b -= (e+k+l+d-n*n-
e+c+j*k+m)%100;
            h += (n-j-d*h)%100;
            e -= (b+m-g+n-e*d-k*h+g+a-a-
k)%100;
            c -= (f+b)%100;
            c += (c+h+c-m-i+d)%100;
        }
        d += (k+i)%100;

```

```

        d += (l*j)%100;
    }
    } while( ++DOcnt[18]%5 );
    h -= (l*e+e-d-m+e-l-h+e+c-c-i-
c*e)%100;
    h += (k+g*m+a-d)%100;
    m += (b-h+l-e+a+c*k)%100;
    e -= (e+g-d*c*h)%100;
    }
    k += (j+f-d-l-g*k-c*a-n-e+i)%100;
    b += (k-i)%100;
    c -= (h+i*a-g+m-h+l-n-g-a-b-
a+j*j)%100;
    k += (d-f)%100;
    m += (l-j-h*h*k-l-d+a+c+f-k*g-
g)%100;
    }
    l = (h+e-h*a+f+m*a-j-b-e)%100;
    g -= (g-c-l+e-h)%100;
    j = (l-a-a+g*a+m-e*j+e)%100;
    d -= (e*i+e*f-g)%100;
    g += (c*b+a-m)%100;
    m -= (j-e+g+k+a-k+h-
d*g+m+g+d+e)%100;
    d += (m+l+k+j-f+n-c*j+i-b+k-c)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F10(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    do
    {
        if( ++IFcnt[16]%10 )
        {
            f -= (l-h+e-e+f-e*g+m)%100;
            for( ; ++FORcnt[33]%5 ; )
            {
                switch( ++SWcnt[10]%3 )
                {
                    case 1:
                    {
                        if( ++IFcnt[21]%2 )
                        {
                            f += (e-f+e*n+f-
l+j+f+m*n+i*f+n-e)%100;
                            d -= (c+l-b-n-h-e-h)%100;
                            k = (g*h-m*g)%100;
                            n += (b*m-d-i+l*c*j*d)%100;
                            c += (i-j*h-h+l-b-f+n-
g+e)%100;
                        }
                        else
                        {
                            j = (l*k*a+j-l-j-g+a-d-j-
d)%100;
                            f += (a*k-i-k+f)%100;
                            h += (b+k)%100;
                        }
                        while( ++WHILEcnt[21]%5 )
                        {
                            l += (e+c)%100;

```

```

            e -= (g-b*i-i-c+c-f*i-a-
j*k+k)%100;
            j = (c-g+l-j-e-h+d+g*j*a-i-m-b-
e)%100;
            g += (d+h+c+d-c-k*c+j-b+h+c-
c)%100;
        }
        do
        {
            j += (j*g-l+a-
j*b+i+b*g+e)%100;
            m = (j-e+g-c-f+n-d+g+g-j*b*c-
n-k)%100;
            f += (m*g*i+l-c+n+e+g+a+c-
l*g-n-n)%100;
            n += (e+f+d-n*l+g-h-c)%100;
            k -= (b+a+a+k*j+e-g-m+h+h-f-
k+i-b)%100;
            l -= (j*a+n-e+n-k-a+j+a-
j)%100;
        } while( ++DOcnt[21]%5 );
        for( ; ++FORcnt[31]%5 ; )
        {
            f -= (l+j*n*h+n+h-e+m-
g+m)%100;
            e -= (d*j+c-j-a+d*j-g+j+n+d+a-
f)%100;
            g = (h-k-e-j-m+a*m-f)%100;
            h += (k+c)%100;
            j += (a+j*k-c-e+l)%100;
        }
        if( ++IFcnt[15]%10 )
        {
            m += (k*f)%100;
        }
        break;
    }
    case 2:
    {
        for( ; ++FORcnt[32]%5 ; )
        {
            e -= (n+m)%100;
            g -= (g-b+g*b+c+n+i*h)%100;
            b += (a*j+j-i+h)%100;
            c += (l-l-i-c-n+n+n)%100;
            f = (a+h+c)%100;
        }
        if( ++IFcnt[22]%2 )
        {
            m -= (m*c+i+n*i*h-h+k+f-b-
f)%100;
            i -= (d+l*m-g-n)%100;
            n -= (b+n)%100;
            d += (n*i+c-g-c-m+h+m-
i+a*m*i)%100;
            a -= (n-n*l+e+k+l-
e+m+b)%100;
            l -= (b*i*h+f-m-n-g*i*g-
d*h)%100;
        }
        else
        {
            j += (b*h)%100;
            h -= (h-e*i*h+c+l-c-d+k+d-g-i-
b*g)%100;

```

```

        f += (e+m*k-n*g)%100;
        h += (n*j+i-e+h-j+k)%100;
    }
}
break;

default:
{
    while( ++WHILEcnt[22]%5 )
    {
        d -= (c+i-j+e*c-m-
d+e+m)%100;
        a += (i*i+e*n-i-h*e+a*h)%100;
        m = (e-h-c-l-h)%100;
    }
    do
    {
        i += (d+j+n-c+g)%100;
        l += (k-j+b+e-d)%100;
        e += (n+m+l+f-e-g-
m+k+g)%100;
        j -= (l-e*c+h+m*m*i-b-
d+m+a+j-d+d)%100;
        j -= (c*b-d-f*a-e-i+g+h+c-
m)%100;
    } while( ++DOcnt[22]%5 );
    c -= (f+b+l-i-l*i-h+j)%100;
    d += (f+m+j*e)%100;
    j -= (e*a+c+c-k)%100;
}
}

f += (c+j+m+g)%100;
f -= (d*c+k-b-c+c-h+i+d+b+f-a+g-
m)%100;
j += (m-n+i+b*d-a-m-a-n+i)%100;
j -= (j-b*l-c*k+b*b*l+n+l+m)%100;
a += (b-j-e+k-a-e+g*l-h-b-
f*n)%100;
}
g -= (i+h*a+e*j)%100;
l += (f-l*e-n)%100;
}
n -= (e+n+k*j-a+m)%100;
c -= (k-g)%100;
j -= (c-j+f-c+l+l+f-
m+n+h*n*i*e*c)%100;
m += (e*d*h+e)%100;
} while( ++DOcnt[20]%5 );
l -= (b-n*n)%100;
l += (d-l*h+g*n+b-d)%100;
i += (g+m*i+e-b-k*c+d*m-e)%100;
m -= (i*g-h-h*a+c+k-m+e-m-a-
k*a)%100;
a = (b+k-n-i-n+a*e)%100;
a += (d-k)%100;
m += (k+l*j+h+n-l-e-d)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F11(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[37]%5 ; )

```

```

{
    switch( ++SWcnt[11]%3 )
    {

        case 1:
        {
            if( ++IFcnt[18]%10 )
            {
                if( ++IFEcnt[24]%2 )
                {
                    while( ++WHILEcnt[24]%5 )
                    {
                        b -= (c-e*a+m*m+g+c-
b+e*b)%100;
                        i += (m+b*i+h-h-h-e)%100;
                        k -= (e+c*c*m-j+g-a*j)%100;
                        e -= (k-g*j-f-
i*e+l*g*n+n+e+c+g)%100;
                        a += (k-i+f+a-b+j*a-j-
n+l)%100;
                    }
                }
                do
                {
                    c -= (n-k-b*b-d-a-e+c-m-
d)%100;
                } while( ++DOcnt[24]%5 );
                if( ++IFcnt[17]%10 )
                {
                    b += (a+k*i-j*k)%100;
                    c += (j*d+d-c-j-n-a+j-
k*g)%100;
                }
                h -= (d+l*k+f)%100;
                for( ; ++FORcnt[36]%5 ; )
                {
                    k += (d-k-i)%100;
                    j -= (f*i-l+l+k*j-b-b-g*g-f-
m*k)%100;
                    g += (m*b+h-l+b-k*h-a*a*j*g-
m)%100;
                    i -= (f-i+b*j-i-d+c)%100;
                    a += (l-h-i-k-g+m*k-
d+n*i*k+e*g)%100;
                }

                switch( ++SWcnt[12]%3 )
                {

                    case 1:
                    {
                        h += (h-e+d+b-f+g*m-
m+e+a+h+m-i)%100;
                        l -= (j*n-
b+k+b*m+k+e+n)%100;
                        c += (a+l+f*g-m-i-f-
m*d*h+k*h)%100;
                        n -= (d-e-l+n+f+l+n+n+l*h-h-
j+j)%100;
                    }
                }
                break;

                    case 2:
                    {
                        e += (j+b)%100;
                        b -= (i-j)%100;

```

```

        c -= (n+b+b+m*f+g+l-l-
a+i+h)%100;
        e -= (a-n-e)%100;
        j -= (h+g-f+d+f+a-h)%100;
        d += (g*d)%100;
    }
}
break;

default:
{
    k -= (g+n*n-n+a)%100;
    i -= (h+e+i+c-l-a-l-n)%100;
    f += (i+f*f*a+a+l-d*a+b-
g)%100;
    b += (m-n-f-d+a+i-c-c+j)%100;
    i += (a*k-e+n+f-c*b*n-
n+h)%100;
}
}
else
{
    while( ++WHILEcnt[23]%5 )
    {
        d += (h-l-e-f-m+i+i)%100;
        j -= (k+g-c-a*m)%100;
        m += (g*h)%100;
        f += (b*m-m+e-n-g-e+l)%100;
    }
    do
    {
        a -= (j+h*g-f+k-m)%100;
        k += (d*i+h-e-d*d*g*d-a*h-i-
g)%100;
        i -= (b-g)%100;
        l += (c-l*m-m-h-k+c*a+k)%100;
        j += (l+j+a-g*d*l*k)%100;
        g += (k-j*i*h-d+l*i*i-m+c-
c+n+l)%100;
    } while( ++DOcnt[23]%5 );
    for( ; ++FORcnt[34]%5 ; )
    {
        j += (h+a)%100;
        a += (e+f-e-h-l+i-g+m-g-l-
n*k*n-g)%100;
        h += (k+f-h+e+i*b)%100;
        g -= (d+f-j-h+m-n*c+c+g-i-
h)%100;
        e += (f*d+l+j+a*d-e-l)%100;
    }
    for( ; ++FORcnt[35]%5 ; )
    {
        g += (d+b-l+a+k+a*f-a*j+c*h-
l)%100;
        b = (e+e*i*m-e*c+i)%100;
        f -= (d-i+f-l-m*h+l+j+c)%100;
    }
    if( ++IFEcnt[23]%2 )
    {
        a += (i+d-l-e+f-c)%100;
        k += (c*e+a+i-c-c)%100;
        c += (e+m)%100;
        n -= (n+f+j+e*g+f-i-d+l)%100;
    }
}
else

```

```

        {
            d -= (f-n)%100;
            g = (h-l+h*e*f*h+g-g*c+g-
h+e+c+f)%100;
            d = (i*l-f+b-e-h+c+m+l)%100;
            l -= (b+n*k+j)%100;
            e = (i-e*c-i*d-c)%100;
        }
        n += (m+e*j-m+n+d+h-e-f-n+j-
i)%100;
        k += (d+l)%100;
        m += (k-k+e*c-j)%100;
        d += (c+b+c-l)%100;
    }
    e = (n-l-d-b-m+i+j+m-g+j-b)%100;
    l -= (j+k+k-k+n+a+c+n-b-k+i-
d*b+j)%100;
    l += (f+a*e+h+m*n+f+g*l-l-i)%100;
    e -= (i+k+c)%100;
    l -= (h*b+n*l)%100;
}
break;

case 2:
{
    i = (h+n-c-n-m*i+h*k+g-e+k-e+d-
h)%100;
    f += (g-h+j-e-c+a-f)%100;
    m -= (g+f-h-e-l+l+m-c+k)%100;
}
break;

default:
{
    m -= (l+k)%100;
    n = (d-e)%100;
    k -= (h-a-k*h-j-g*b*d)%100;
    c -= (d-b)%100;
    l += (j-g-c-e*c+a*e-l+n*f*i*i)%100;
}
}

g += (b+m+c+c+h+k-f-l+e+l)%100;
f += (h-b)%100;
m += (n-i+a-d*i*i*k-c-d*k)%100;
}

a = (f-c+m*b*b-n+e+b)%100;
g = (n-g)%100;
k -= (k-f-m)%100;
k -= (i-l-g+l*k-a-i+l-c-h-g+g)%100;
b += (e+a+e-g*h+h+i-c)%100;
j += (k*g)%100;
m -= (e-e+m+h*m)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F12(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFCcnt[27]%2 )
    {
        i += (j+c+l-c)%100;
        d -= (k+k-m*m+c+m*j-d)%100;

```

```

g -= (g*k-i*d+m-d-c)%100;
b += (d-i+j*k*n+m*j+e*g*m)%100;
n -= (h-l+b)%100;
}
else
{
    while( ++WHILEcnt[26]%5 )
    {
        do
        {
            for( ; ++FORcnt[39]%5 ; )
            {
                if( ++IFcnt[19]%10 )
                {
                    a = (f+n*k-e-b-c+m-n)%100;
                    m -= (f-e+c-k-i-j+j)%100;
                    b += (c+k-l*b+l+l-m-f-c+c-c-
j*d*h)%100;
                    b -= (b*j-i*c-g-n)%100;
                }
                for( ; ++FORcnt[38]%5 ; )
                {
                    j -= (f-e+g-c-a+n+g)%100;
                    e -= (j*m*l-l-j+m+a-d-j+l+n-
e+e*k)%100;
                    h -= (i-j-c-e-a*d+k+c-c-
d*f)%100;
                    j += (g+n-k+e+b+d*g-d-c-c-
f)%100;
                    k -= (b+b+c-
d+f+b+d+j+a)%100;
                    j += (g-n)%100;
                }
                if( ++IFCcnt[25]%2 )
                {
                    e -= (e*n-n+h+m+k+e*j-
e+m)%100;
                    d += (n-e-l*m+n+a-b-b)%100;
                    b -= (b-f-c-d+f+n-g-c-d-e-h-
n+f+c)%100;
                }
            }
            else
            {
                b += (l+m+g-c+l)%100;
                e += (m+b+i-l*h+g-l-
b+h)%100;
                h = (h+e*e+m-n)%100;
                g -= (n+k+d-c+e+l-m*e+d+h+i-
k)%100;
                k += (g*g-f-k-
d*g+j*a+k*f+n)%100;
            }
            while( ++WHILEcnt[25]%5 )
            {
                g += (j-n)%100;
                l -= (a-d*b*l-k*n+h-
n+k+f)%100;
                k += (g-l+m+k+b-l)%100;
                j += (e+g-c-m)%100;
                j -= (a+l+c*m+e*a+g+m-n*a-
f)%100;
            }
            do
            {
                a += (j-k-f+n-n-j)%100;

```

```

g -= (n+c+g-k-h-e+d-b*g-
c+j*n-n)%100;
i -= (e*d-h*b)%100;
h = (c+g+f-k+c)%100;
} while( ++DOcnt[26]%5 );
}
l -= (f*h-f*n)%100;
for( ; ++FORcnt[40]%5 ; )
{
    switch( ++SWcnt[13]%3 )
    {
        case 1:
        {
            f += (g+f-l-f+b+j+b)%100;
            a += (e-f+l+a*d)%100;
            k += (n+m-b-n)%100;
            d -= (l+a+j+n-n-f-e+c+c+g-b-n-
d+e)%100;
            c -= (h*f+n-n-d)%100;
        }
        break;

        case 2:
        {
            b += (d-c*l+g+m)%100;
        }
        break;

        default:
        {
            c -= (m-d*g+h-n)%100;
            e -= (g-m+c+a+e)%100;
        }

        if( ++IFcnt[20]%10 )
        {
            l = (h*g-n+b-k+e+j)%100;
            f -= (n+b-j*g*n-b+g*l-f-m-
h)%100;
            k += (a+a+h+c+g)%100;
            n += (k+e-k+a*d+e+c)%100;
            e -= (l-a*l*a+m*h-i*k)%100;
        }
        if( ++IFCcnt[26]%2 )
        {
            j -= (n+g*a-b-f-g-e-
m+e*h+n)%100;
            d += (a*n)%100;
            h += (g+m*j-g)%100;
            e -= (d+j+a-c+l+d+j-k)%100;
            j += (m+b-j+d+m+e-b)%100;
            d += (c-g*g-h+n*b+l-d-f-c+d-
c)%100;
        }
        else
        {
            l = (a*k*m-n-n+d+h*d+e-g+k-
e)%100;
            c -= (d-a)%100;
            j += (l+m)%100;
            g -= (a*i+c-f)%100;
        }
        b -= (c+b+b+m)%100;

```



```

else
{
    n -= (b+k+c-l-k*e+e+f*a*i-
e+f*f+m)%100;
    j += (a-n-n+g-l+a*j-a-a+k-
k)%100;
    c += (e-
a*d+g+d+e+g+e*k)%100;
}
while( ++WHILEcnt[30]%5 )
{
    h += (c*g-d+k-g+m+m-i)%100;
    j -= (e+b-e-c-n+d*f+m+b-i-n-
i)%100;
    f -= (c-c-a+h-n+k-n-i-h-n+b+f-
a)%100;
    l -= (m+h-d+e-d+j-e*b+g-a-
i)%100;
}
do
{
    n -= (k+e)%100;
    e -= (c-n+a*h+a+n+k+e-g+n-
b+c)%100;
    b += (j+e*f*f+e-n-k+d)%100;
    b -= (d+e-m-h+d-k+g*m+b-b-g-l-
j+l)%100;
    b -= (n-b+m-n+j-g-c+k-d-
j)%100;
    e -= (d+m+l-a*n-j+g-j-n-
m)%100;
} while( ++DOcnt[30]%5 );
d -= (f+i*e-j-d)%100;
for( ; ++FORcnt[44]%5 ; )
{
    m += (n+a-f*a-a+g-j*m-l-
f*c+n-h)%100;
    j += (d+c*c+h-i+j)%100;
    f -= (h-n-l)%100;
    a = (j-g-h*g-k*c*d+l+n-
k)%100;
    n -= (m-g)%100;
}

switch( ++SWcnt[15]%4 )
{

case 1:
{
    if( ++IFcnt[22]%10 )
    {
        j -= (j-g+m-a+n)%100;
        m += (i+h-m-e*a*c-l-d-m-
l)%100;
        l += (j+d+c+d+f-k-
d+d*a*n+m)%100;
        b -= (k-d-f+i+m+e+c+f+j-
b)%100;
    }
    if( ++IFcnt[31]%2 )
    {
        k -= (c*a-a*m)%100;
        j -= (i*m-b+d-g-g+k-c-l*f+n-
k+j)%100;
        g -= (j-g-m)%100;
    }
}
}

```

```

c -= (c+h-f-m-i-j+a-f*a)%100;
m += (d*k+b*j*h)%100;
}
else
{
    h += (d-d+b+j-i-
c+a+h+n+i+k+d)%100;
    m += (e*a+k+b-a)%100;
    l -= (a-j)%100;
    j += (n-c+f-a+k)%100;
    d -= (h+l-m-m-m)%100;
    d += (f+g+n-j)%100;
}
while( ++WHILEcnt[31]%5 )
{
    j -= (n-m-c-h+d+k)%100;
    n += (b-e-m+k-k-c+c-g+g-
h+d+e)%100;
    e += (j+d*c)%100;
}
do
{
    k += (d+m*k-n-g-h-b+b)%100;
    l -= (k*f*b*c+n+i-k-a)%100;
    b += (e-n*g+k+i-c-i-b*j-
h)%100;
    g -= (n*e+f+n*i*i+h-m-e-
b*k)%100;
    n += (i+g-d-b*m-g)%100;
} while( ++DOcnt[31]%5 );
for( ; ++FORcnt[46]%5 ; )
{
    j -= (l+d-d-m-g*j*f-
l*h+j*n)%100;
    h += (f+k+e+i+b*f+i+b-
n)%100;
    k += (c-g-c+c+c)%100;
    k = (f-l-d-g-h-e-k+e)%100;
}
break;

case 2:
{
    m += (j+g+g+m-i-f-g*d+d)%100;
    n += (m+k*j*j+d)%100;
    n = (d+f-m*m+f)%100;
    f += (m+l*a+j+a*i)%100;
    e -= (b+c)%100;
    d += (m+l)%100;
}
break;

case 3:
{
    a = (j+h-h-j-d*n+l+j-
h*e+d*c+i)%100;
    j -= (d*n+h)%100;
    l = (k-n+d+b)%100;
    i += (j-g+j-j*f*d+f+n*k*f-j-
j)%100;
    b -= (b+d-h+n*b-
f+n+k+k*g*i+i+e-a)%100;
}
break;

```

```

default:
{
    e += (j+j-j+k+k+d+n-a-
g+m+l)%100;
}

m -= (b-g+m)%100;
d += (h-a+b-l-a)%100;
h -= (d+m*h)%100;
h += (k-c-e-k+k+l+l-m+f)%100;
}
k += (c-l*e-f-n+m-l+n+k+c+i-f-
k)%100;
d -= (m-b*n-c-n*d+b*h-i-a-h-
n+j*j)%100;
b -= (n-d-n+l-a)%100;
}
j += (f*j-f-e+f*l+h-m-k+j-m)%100;
b += (b+m-g-f+e-g+l+j*d+k+f)%100;
h += (g-g*m*b-d-k+m+i*d+l+f+d-e-
b)%100;
k -= (k+c+d-a-n+k+h-b)%100;
} while( ++DOcnt[29]%5 );
b -= (i-m)%100;
h -= (h+n*l+a)%100;
l -= (m+j)%100;
e += (h+d-e+k-g)%100;
d -= (b*n+f*c+h+n+a-j+g+i)%100;
a -= (a+f+h*h+h+b)%100;
g -= (l-g+i*g)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F15(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[50]%5 ; )
    {
        if( ++IFcnt[33]%2 )
        {
            c += (f-l)%100;
            g += (b-b-d*m-h-m-b)%100;
            l -= (c-e*a-e*e-c*d)%100;
            g += (c*h)%100;
            j += (n+e*d+k)%100;
        }
        else
        {
            while( ++WHILEcnt[33]%5 )
            {
                do
                {
                    if( ++IFcnt[24]%10 )
                    {
                        h -= (l-j+m+l+f)%100;
                        g += (d*b*c+b+a+m-l-l-
i+m+n+b-h)%100;
                        e -= (a-d+k-f+i+d-n*n-
f+k)%100;
                        b += (i-b*m*c*k+f-k*b-
e*k+e+l-e-c)%100;
                        a -= (f-a+e-b+d+l-b*j-f-
d)%100;
                    }
                }
            }
        }
    }
}

```

```

}
e += (c+a-m-k+e+d-h*e)%100;
for( ; ++FORcnt[48]%5 ; )
{
c = (c-n-l-m-f+b+l*b*g)%100;
d -= (a+j*i-n-m+k)%100;
a -= (c+d-c+b*d-j+h-f)%100;
}

switch( ++SWcnt[16]%3 )
{

case 1:
{
l += (c*f+d)%100;
m += (f*c+e)%100;
c = (d-h-g-k+a+a+m-m-
d*d)%100;
b += (i*k+a+l-g-g-c+h-m*m-
b+n+j)%100;
l -= (c*c+g*c)%100;
}
break;

case 2:
{
l += (c-k-h+m-n)%100;
f += (h-g+g-l-a-f-b)%100;
i -= (e-a-l-d+c+g+m+d+e-
c)%100;
f += (g+l*f-j*i)%100;
}
break;

default:
{
i += (l+m-d*a-e+n-f+m-
d+h+m*i-n+m)%100;
e += (g-h-a+j-l-d)%100;
e -= (e+i+h*c+j-a-n+f-g)%100;
a += (j*e*e-j*b-k+i-c-
c*h)%100;
a -= (e-a-c+f+g-l-e+g+i-b-
f)%100;
d += (g*i+m+g-l+c+m+k-
h+f+f+c)%100;
}

if( ++IFcnt[32]%2 )
{
c += (b-a+c+m+e+e-i-
g*a)%100;
a -= (f-k-e-d-e-f-k*f*g)%100;
n -= (j-d+d+n-e)%100;
c += (e+e*m-m*j)%100;
n -= (k-b+l-b)%100;
}
else
{
m -= (m-c+m*a-c+a*b-l-k-
n*e*d)%100;
g += (j+b)%100;
a -= (f*i-l-k-n+i+k)%100;
e += (a-e-b-h+e*i+f+g)%100;
b -= (n-d)%100;
}
}

```

```

}
while( ++WHILEcnt[32]%5 )
{
c -= (k-g+f-e+j-m+g*i)%100;
i -= (g+n-e-h)%100;
k += (k*e-m-c-k-m+k-i)%100;
f = (e+g+f+c-d+k)%100;
}
} while( ++DOcnt[32]%5 );
do
{
for( ; ++FORcnt[49]%5 ; )
{
m -= (n+k-h-d+l+i-k-i*i*f-a+c-
g+l)%100;
f += (m+a+d-f-l+k*l-l-i*i)%100;
c = (a+e*g)%100;
k = (j-e-n)%100;
b -= (e*c+b*f*n*k*j+n*j-k-e+j-
m+f)%100;
}
if( ++IFcnt[25]%10 )
{
i += (d-k-i+f-e+i+j+a+d*e-l+g-
b)%100;
k -= (l+l-e+n-
b+b+j*f*d+b+i+f)%100;
f += (g+d*c*a*d-b*l*a+f-
h)%100;
}
e += (e+i+f-m+a+d*l-
h*l+f)%100;
c += (j-j*n-a+d+k+i+h-j-b-g-
i+j*k)%100;
k -= (k-j+n-h-c-l-g)%100;
f += (f-l-c-f-a+f-
d+b*e+c*j)%100;
} while( ++DOcnt[33]%5 );
k = (b-j*h-m+i+d+i-n*e-d-
j+h+n+a)%100;
d += (f+f+b*b*d-g+f-b+e-d-c+n+k-
d)%100;
}
j -= (b+g+h-m+h+d*l-m-i-k-l)%100;
d -= (i+l+c)%100;
b -= (e-c*i+h-f+n)%100;
b -= (a*b+g)%100;
}
m -= (h+f*j+i*f-d+l)%100;
}
k -= (b-j+h-c+l*k*e*n+l+h+m+d*k-
b)%100;
a -= (m-a+d-d*e*b+e+c+k-j)%100;
c -= (d+f*b+f+c-d+n-g-n)%100;
c = (k+l-h-l)%100;
c += (e-n+a+b-k)%100;
e += (g-j+d+l-l-b-j-e-j-e)%100;
i += (d+c-h-j+b*m-d+e-a+d+k-i)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int main(void)
{
int l;
clock_t StartTick = clock();
}

```

```

for(l=0; l<38; l++) IFcnt[l] =0;
for(l=0; l<51; l++) IFcnt[l] =0;
for(l=0; l<25; l++) SWcnt[l] =0;
for(l=0; l<51; l++) WHILEcnt[l]=0;
for(l=0; l<51; l++) DOcnt[l] =0;
for(l=0; l<76; l++) FORcnt[l] =0;
long int sum=0;

sum += F1( ) ;
sum += F2( ) ;
sum += F3( ) ;
sum += F4( ) ;
sum += F5( ) ;
sum += F6( ) ;
sum += F7( ) ;
sum += F8( ) ;
sum += F9( ) ;
sum += F10( ) ;
sum += F11( ) ;
sum += F12( ) ;
sum += F13( ) ;
sum += F14( ) ;
sum += F15( ) ;

{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
for( ; ++FORcnt[52]%5 ; )
{
if( ++IFcnt[34]%2 )
{
i -= (i+a+c-n+g*j-j*d*a-n+m-
j)%100;
f -= (g*c*n-a*a-m)%100;
h -= (h-i+c+j*f+n+k)%100;
h -= (i-f-b)%100;
k = (b-l+h+f+c-m+i+f-j*a)%100;
}
else
{
while( ++WHILEcnt[34]%5 )
{
do
{
l = (f+b-f-d+i+n*l+k)%100;
} while( ++DOcnt[34]%5 );
j += (b-g-m)%100;
for( ; ++FORcnt[51]%5 ; )
{
n += (k-n+g-k-a-d+a+c+e-
n)%100;
j -= (e-g-c+j*j-j-g+k+l-j-g*f+h-
l)%100;
}

switch( ++SWcnt[17]%3 )
{

case 1:
{
i += (d-f-l)%100;
l = (n-d-m+n+l)%100;
c -= (b*k+c+c+f+a+e*b*c-k-
c+h-n)%100;
a += (c-k+j-l+l+c+i-c+a-a)%100;
f += (e+c-j+d-f+c*n+d)%100;
}
}
}
}

```

```

    }
    break;

    case 2:
    {
        g -= (i-a+e-m-i*f-m*a-e)%100;
        c -= (j+l)%100;
        c -= (f*d-k*j-i)%100;
        c -= (i+e-g-d-a*e+a)%100;
    }
    break;

    default:
    {
        f += (g+i)%100;
        a += (a*h+i*n*d*g-f-a-f-
a+a)%100;
        e += (j*f*h-e-l-f-d-i+g-n+g-
i)%100;
        e += (g-e+c*f-a-k-e*k*d+i-d-
c+a)%100;
        m -= (d+l*f-n-d-a-f*d-
l+n+b+f*e-c)%100;
        n -= (a-m*e-h-e+k-c-h*f-
c+k*e*e)%100;
    }
    }

    h += (h-f+f*m*a-i)%100;

```

```

    }
    h -= (d*a+b-f-l-e-a-b+b+i+k*g+d-
e)%100;
    c += (e+k*h+b*n-f+g*k*n+n-h-
a*d-k)%100;
    l -= (j-a-c-n+h*m-j-j-
k+j*a+m*k)%100;
    m += (k+n+b-j)%100;
    k += (c-g-a+d*c*h+c-m)%100;
    }
    f += (h*f+h-b+e+i-h)%100;
    d += (j*c)%100;
    n += (a+f+c-h+n)%100;
    }
    i -= (g*m-g-b-d-n)%100;
    g -= (c-l+c+i+c+k-n+d-f+d-n-d)%100;
    m += (e-m*d*i+i*n-h+m*d*-
f+e+e*f)%100;
    i -= (b*l-n-c*c-d-e-k-g+a)%100;
    a += (b+e*g-b-n*h+n+i*n*h-h)%100;
    f -= (c+f*a-j+a*g)%100;
    h = (j-m)%100;
    sum +=
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
    }

    cout << "\nChecksum = " << sum;
    for(l=sum=0; l<26; l++) sum += l*fcnt[l];

```

```

    cout << "\nIF frequency:   Static = "
<< 26 << "   Dynamic = " << sum ;
    for(l=sum=0; l<35; l++) sum +=
l*IFcnt[l];
    cout << "\nIF-ELSE frequency: Static =
" << 35 << "   Dynamic = " << sum ;
    for(l=sum=0; l<18; l++) sum +=
l*SWcnt[l];
    cout << "\nSWITCH frequency: Static =
" << 18 << "   Dynamic = " << sum ;
    for(l=sum=0; l<35; l++) sum +=
l*WHILEcnt[l];
    cout << "\nWHILE frequency: Static =
" << 35 << "   Dynamic = " << sum ;
    for(l=sum=0; l<35; l++) sum +=
l*DOcnt[l];
    cout << "\nDO frequency:   Static = "
<< 35 << "   Dynamic = " << sum ;
    for(l=sum=0; l<53; l++) sum +=
l*FORcnt[l];
    cout << "\nFOR frequency:   Static = "
<< 53 << "   Dynamic = " << sum ;
    cout << "\nRun Time = " <<
double(clock()-
StartTick)/CLOCKS_PER_SEC << "
sec\n\n";

    return 0;
}

```

## d.)BM1BMProgram12

```

#include <iostream>
using namespace std;

#include<time.h>

int
lFCnt[57],lFECnt[76],lSWCnt[38],lWHILEcn
t[76],lDOcnt[76],lFORcnt[115];

int F1(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[3]%5 ; )
    {
        if( ++lFECnt[1]%2 )
        {
            f -= (c+d+i*h+k-d-h+h*b-
b*j+m+f)%100;
            l = (n*a-a+n-e)%100;
            e -= (m+l-m-n-g)%100;
            l += (j+f+e+l+h*c)%100;
        }
        else
        {
            while( ++lWHILEcnt[1]%5 )
            {
                do
                {
                    if( ++lFCnt[0]%10 )
                    {
                        i -= (n+h-e*d+f+d-g*b*i)%100;
                        d -= (f-k)%100;

```

```

                        l -= (k+a-c)%100;
                    }
                    n += (g*i+k+j)%100;
                    for( ; ++FORcnt[0]%5 ; )
                    {
                        b += (g+c+i-m-n-
k*g+c+j+c)%100;
                        g += (c+h-k+k+m)%100;
                        l -= (j-i+j+f-n+n+h+j+g-b*f-
k)%100;
                        h += (c-a-j*d+k+d+a+f-m-f-
k+d*g)%100;
                        m -= (g+h+m+b-a-
j*j+e+k+b*i*f-a+g)%100;
                    }

                    switch( ++lSWCnt[0]%4 )
                    {
                        case 1:
                        {
                            b -= (e*i)%100;
                            f -= (b-b*j*d-l+c-a-
b*l+e*n*l)%100;
                            c += (b*g-a*l*j+h+f+k-i-
l*n)%100;
                            b += (b*m-e)%100;
                        }
                        break;

                        case 2:
                        {
                            b += (a-a*f-l-n+e)%100;

```

```

                        b += (k-d+d+j*e-i+d-l+a-a-
g)%100;
                        f += (k-a*k*k*e+l*i*d-d-
a+d+g+l)%100;
                        b -= (b-d-f+g-a)%100;
                        d -= (d+b+a*l+m*e-b+m-d-
m*n+g+m)%100;
                        i = (l-e)%100;
                    }
                    break;

                    case 3:
                    {
                        k += (a*n-i-a+d)%100;
                        b -= (h*k)%100;
                        i -= (a+g+a-l-m+n)%100;
                        f -= (m+a+m-i-h+d*b-c-b+f+f-
d-f)%100;
                        j = (j+l-k+j*k-g-e*l-d+a+n-
a+m)%100;
                    }
                    break;

                    default:
                    {
                        i += (j+l-k)%100;
                    }
                }
            }

            if( ++lFECnt[0]%2 )
            {
                l += (a-g-h)%100;
                f += (b*f+j+k+k-i+l)%100;

```



```

        k += (m+k-d+j*d+l-c-g+b-i-
b+l)%100;
        i -= (b+a*i*m-m+g+b+g-a*d-
j)%100;
        h += (h*n)%100;
    }
    else
    {
        i += (l-k)%100;
        k += (k+g+k-g*n-
e+a+b+k*d*j+f*e)%100;
    }
    } while( ++DOcnt[0]%5 );
    while( ++WHILEcnt[0]%5 )
    {
        do
        {
            b += (j-i-n+f-j-b)%100;
            m = (d-d-l-i+c+j)%100;
            j -= (k+i-l+f)%100;
            i = (d+c+h*n+b+c-d*c-
f*b+h)%100;
            c -= (c+e)%100;
            n -= (g*b+h-l-f*h-d*i-
b+e+c)%100;
        } while( ++DOcnt[1]%5 );
        for( ; ++FORcnt[1]%5 ; )
        {
            e -= (j*c)%100;
            f = (i+c-d-
j*l*e*f+k+e+k)%100;
            f -= (b-e-j*e+b-l+d*n+h)%100;
            k += (j*n+a+m-b)%100;
            i -= (n-j*k+n*n-b+e+g-
g+e+f+i+h)%100;
        }
        if( ++IFcnt[1]%10 )
        {
            h -= (i+k-d*c+g+h-b-
k*e*j+g+j)%100;
            b -= (e-m)%100;
            n -= (g-d-k-j*l-f*g-m)%100;
        }
        for( ; ++FORcnt[2]%5 ; )
        {
            l += (b-a)%100;
            n -= (i+b+h+i+l+e)%100;
            g -= (i-f)%100;
            l -= (m+h*h-j-k-k-
b+l+l*n+b+k+l-n)%100;
            h -= (m-i-h+j-j)%100;
        }
    }
    b = (f-e*c-k+m*d-e+f*c)%100;
    b -= (i*k+f-h-d+n+k+g)%100;
    b += (g*e-i*h-h)%100;
    e = (e+i+d+e-d-n-j-b+e+g*e+e-
n+e)%100;
    }
    i = (k*g-i-c+h-n-k-j-e+b*f-h-
k+n)%100;
    n = (m-i*k+h*c+c)%100;
    g += (c*a*h*c+n)%100;
    }
    l -= (a*e)%100;
    f -= (m*m*h-n+k*d*k)%100;

```

```

        f += (g+l+g+h+d-d-n-i-i-g)%100;
        l -= (m+d)%100;
    }
    m += (h+d-a*d-i-e+i-m)%100;
    e = (c-n+l-d)%100;
    f += (k+m+f+n*a-d-e)%100;
    c += (b-n+d-h)%100;
    h += (b-b+g-i+l-c+f*n*f+d*b-
a+h)%100;
    f -= (n-i-g)%100;
    i += (j-i+d*j+i+l*a+i+l+c*c)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F2(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFEcnt[4]%2 )
    {
        j += (g+m+g*i-f)%100;
        m += (h+m-c-c+a)%100;
        f -= (j-k-f+c-i-f+i)%100;
        d += (c+e*d-l-j)%100;
        f = (d-f+d+k)%100;
    }
    else
    {
        while( ++WHILEcnt[4]%5 )
        {
            do
            {
                d -= (d-l)%100;
                for( ; ++FORcnt[4]%5 ; )
                {

                    switch( ++SWcnt[1]%4 )
                    {

                        case 1:
                        {
                            d += (e+a*b+a+i+b-
c+k+h+a)%100;
                            h += (e*i)%100;
                            i += (n+i*h*n*n-l)%100;
                            n += (c+b)%100;
                            b -= (m-m+m-n+b+g)%100;
                            i -= (c+n*g-e-b-g-f)%100;
                        }
                        break;

                        case 2:
                        {
                            a += (d-j-k-f+b+b)%100;
                            h = (b-a-n+g-j*k+k)%100;
                            c -= (g*a-a-a*b+e-l)%100;
                            h -= (i+g*a-d-f+h+a-m)%100;
                            n -= (j-b+j-b+g-m)%100;
                        }
                        break;

                        case 3:
                        {
                            n -= (k+g)%100;
                            k += (l+n+j-d+j-k+a-j-g)%100;

```

```

        j -= (a-m+b+b-g*h-j)%100;
    }
    break;

    default:
    {
        f += (n-b*j+j+k-l-m+n+e+k+g*i-
m)%100;
        a = (j+n-l*n*n-c+g-j-g-l*m-
c+c)%100;
        c -= (m*k+e-f+i+f-g-a*e-
e+n+h-g)%100;
        h -= (i+b*h-l+h+e-h)%100;
        i -= (b+i+l-f+e+a-k)%100;
    }
    }

    if( ++IFcnt[2]%10 )
    {
        b += (i+k)%100;
        b -= (j+k+l-f+n-i-i*e+k*j-h-
i+i+g)%100;
        e -= (e-n-i+b+e*i+l-
c+d+c*g)%100;
        c += (i-c+m-n+e-e+c-m)%100;
    }
    if( ++IFEcnt[2]%2 )
    {
        c += (l-g-f+h-l-l*g-
c*a*b+m)%100;
        j -= (g-j-a-h-e-n-a+l-f-j)%100;
        g = (d-j*d+f-j-l+j+g-f+l)%100;
        m -= (j+i+m+f+g)%100;
        i += (b-m+m-i*n-k-b+b-l-
j+k)%100;
    }
    else
    {
        m += (j+c-l+m*f+h+m-a-
k+c+h+n)%100;
        g += (i*g+l+k-n-g)%100;
        d += (a+j*h)%100;
        i -= (e+b*j-j-h-c+a-c)%100;
        m += (f-n+h+h-j-j+n*g)%100;
        n -= (i*a*d)%100;
    }
    while( ++WHILEcnt[2]%5 )
    {
        b = (d-k-n-b+k)%100;
    }
    }
    do
    {
        for( ; ++FORcnt[5]%5 ; )
        {
            f += (k+i-m-i+a+h)%100;
            h += (f*b*k+k+j*e-a-d+g-
g*n+a)%100;
            b += (j+b-e-i-b-e+f-b+b)%100;
            l -= (n+l)%100;
            l = (f+f+a*i-e+i*n+g-
m)%100;
        }
    }
    for( ; ++FORcnt[6]%5 ; )
    {

```

```

        m += (l-f*g+g-i-l-l-l-f+k*j*h-
g+h)%100;
        k = (e+i+g+b+i-h*n)%100;
        a += (l+m*e+n*g+m+a+n-
a)%100;
        h -= (i+m+g-c+d*k*c*d-n-
m)%100;
    }
    } while( ++DOcnt[3]%5 );
    if( ++IFCnt[3]%2 )
    {
        a += (g-j*g-b-f+l-f-a*l-e+b*j-k-
b)%100;
        g -= (c*c-l-h-n+b+h*e-
c+h+b+g)%100;
        c = (g*e+m+f*e-d)%100;
    }
    else
    {
        while( ++WHILEcnt[3]%5 )
        {
            l += (j+j+d+e+a+j-a-d-
k*l+h*e+b)%100;
            b += (j-i)%100;
            a -= (c+m)%100;
            i -= (b-g+f-a-f-d+b)%100;
            k -= (n-b-a-n+k*n)%100;
        }
        e -= (b-c-i+a+c*b-g+c-i)%100;
        j -= (f-n+a+l-f+a-m+f+d+h)%100;
        b += (b-i)%100;
        c += (c-j*j*m*m-g)%100;
        k = (k+d+l*h*)%100;
    }
    c += (i*g-h)%100;
    } while( ++DOcnt[2]%5 );
    a -= (m+i+m+h+i-m-b-i)%100;
    f += (a-l+n+d-n+e-d)%100;
    c -= (h*i+j+a+d+k-n+h-m*d+k-
h)%100;
    c += (g*e+g-d-c+l+c*k*a-i+m-
c+b)%100;
    }
    e += (g*b-b-c-j+i*d-c-i+g-j)%100;
    n += (k-d+m-h-d+n)%100;
    g = (m-c-h+b-e+j+h-n-b+n*b-
e+k+l)%100;
    g -= (g+b-n-h+i*f*h+b-e+b-c-e-j-
j)%100;
    l += (l-g*i+g-f-b+b+a+n+d-e)%100;
    }
    a -= (n+m)%100;
    b -= (e+i*h+n-a-b-b+g-a)%100;
    e -= (m-a*l-i+j+a*d-j)%100;
    l -= (e-j+j-m)%100;
    j += (k-g+f+g-l+m-j-i-j)%100;
    n -= (l+l+b)%100;
    j -= (k-a-f+e*k*b-m-i+l-b-g+l)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F3(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;

```

```

do
{
    if( ++IFCnt[4]%10 )
    {
        f -= (i-e)%100;
        for( ; ++FORcnt[9]%5 ; )
        {

            switch( ++SWcnt[2]%3 )
            {

                case 1:
                {
                    if( ++IFCnt[5]%2 )
                    {
                        c += (h+i-g-h+b-f+l+l-
a+h+f+k)%100;
                        f = (j*e+n-a*k-j-d*h-m+i+k-
i)%100;
                        h -= (l+f)%100;
                        e -= (k-d-k*f-n*h-k+m+k*k-
a)%100;
                        m -= (b*d+j-a-h+g+h*i+c-a-
h+l-n*e)%100;
                        j -= (j+j)%100;
                    }
                    else
                    {
                        b += (h+e+j*m)%100;
                        n -= (j*h)%100;
                        d -= (b*d*d+d-e+d+i-m-l-
c+i+a+h*e)%100;
                        k += (e+e)%100;
                    }
                    while( ++WHILEcnt[5]%5 )
                    {
                        i -= (d*c+c*l-l+d+e-f-e-
e*m)%100;
                        k += (m+a+d-m+m)%100;
                        j += (i+d)%100;
                        b += (e+h-h+m*c*a-k*f-k-h+a-
a+m+l)%100;
                        c += (f+l-m)%100;
                    }
                    do
                    {
                        a -= (f*a+a-f*k-b-b-e-m)%100;
                        d = (b-c+h-f+l*k+b+b-e+c*i+h-
n*a)%100;
                        c -= (n+a+a*i)%100;
                    } while( ++DOcnt[5]%5 );
                    for( ; ++FORcnt[7]%5 ; )
                    {
                        c += (b-j-k*a+i+d*l*h)%100;
                        b -= (k+d+g+e+b+c+a)%100;
                        l += (f+j)%100;
                        c += (n-f-i-d-m+k-k*f*i-
f+d)%100;
                        c += (h+d-j+b+m+i+e-m-
g*e+g)%100;
                    }
                    if( ++IFCnt[3]%10 )
                    {
                        d += (a+j-k-d+l*g-n+b+i-j-
n*j+f+f)%100;

```

```

        e += (b*j*j*m+d+e+a*g+g-
h)%100;
        c -= (i+g-e+h*e+j-k*h+a+b-e-i-
j)%100;
        d += (f-g+a-m-a*a*d+l*a-i+l*c-
j+l)%100;
    }
    } break;

    case 2:
    {
        for( ; ++FORcnt[8]%5 ; )
        {
            d += (b+h-c+d+i-i+n+i+c-a-
c+h+e-k)%100;
            b -= (a+g-n-m*g-h+e*g*j+g-
b+k*e+n)%100;
            e += (g-d*f*d*m-g)%100;
            l -= (l-b-k+l+a)%100;
            i += (f-a-a-n)%100;
        }
        if( ++IFCnt[6]%2 )
        {
            b = (l+m-n*b-b+k-g-e*n-
m)%100;
            e += (g+c-b-m*c*j*i-i-
a+a)%100;
        }
        else
        {
            j -= (l+g+e+g+n*j-n*e)%100;
        }
        while( ++WHILEcnt[6]%5 )
        {
            b -= (n-b*l-i-k-k+n-m-j+l-m+f-
a+k)%100;
            i -= (j-h-n+a*g+b-d-e)%100;
            e -= (a+j+m*m-e*d+h-j*f+l+n-
g)%100;
            j -= (i*b+i-b+n+m-l-l)%100;
            i += (f-a+c-l*m)%100;
        }
        do
        {
            f -= (e*c+k*k-m-f-l*d*i-
n)%100;
            g -= (n+g-d+f+c*n+d*m-
g)%100;
            m += (l-a-n*g*h)%100;
            g = (e-c-a-d-c+g+n+g-b-g*d-
a+b)%100;
        } while( ++DOcnt[6]%5 );
        f += (d*a-i-i-e-k-n-j*e)%100;
        h += (e-d-b)%100;
    }
    break;

    default:
    {
        a = (b-i-d-b+e+m-h-
g+m+e*h+l*a)%100;
        l -= (d-g)%100;
        n += (f+a+g+m)%100;
        g += (h-m)%100;
        l -= (i-b+n-l)%100;

```

```

        i -= (k*n+h-l+g-a)%100;
    }
}

i -= (a+h*j-l-e)%100;
j -= (m+j-e+c-i+j+e-n)%100;
i = (a-h-f*i)%100;
j -= (d+g+l+l)%100;
}
h -= (k-j-a+b-f+j-g-d+b-a+a-g)%100;
b += (f-i-h-a-m+g-m-g-g)%100;
n += (b*m*j-c+m-f-i-j+a-n-l)%100;
f -= (j+n*b+n+b+b-c-m+l)%100;
}
c -= (a+b+e-e*i*b-b*f*i-l-m-b+h-
h)%100;
g += (b-n-c*n-n*m-n)%100;
j -= (a+m*h+f+c+a*f*j)%100;
} while( ++DOcnt[4]%5 );
i += (l*j+b)%100;
j += (c-g+m+i)%100;
c += (e+d-f*i)%100;
e -= (b+l+l)%100;
l += (f-f-i+e-b+e+g*c*h+e)%100;
g -= (j-l)%100;
c -= (a-m+b+b-j+f*j+a)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F4(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    for( ; ++FORcnt[13]%5 ; )
    {

        switch( ++SWcnt[3]%3 )
        {

        case 1:
        {
            if( ++IFcnt[6]%10 )
            {
                if( ++IFEcnt[8]%2 )
                {
                    if( ++IFEcnt[7]%2 )
                    {
                        j -= (n+l-e+f+k+m+m-a)%100;
                        c += (m-d-i+g-h-l-d+m-b*h+a-
i-c)%100;
                        b += (j+k-n-b-e-n*f-c+j-f-
g)%100;
                    }
                }
            }
            else
            {
                k += (k-h-j)%100;
                j -= (f+g+n*k-f)%100;
                a += (b+c)%100;
                c -= (i-e-n-k+k+g-i)%100;
                k = (a+f-i-i)%100;
            }
            while( ++WHILEcnt[8]%5 )
            {
                j = (l-k+g-h-g*i+h-j)%100;
            }
        }
    }
}

```

```

        f = (i+a*k*n-j+g-j+f-c-a-k+d-c-
m)%100;
        a += (j+d+g)%100;
        f -= (k+c-k*c-a)%100;
        i += (a-g)%100;
    }
    do
    {
        k -= (e-e+k+h*i*h)%100;
        l -= (b-a-l)%100;
        j -= (m-d+k-a+e+c)%100;
        m -= (n+b-j+c-b*d)%100;
    } while( ++DOcnt[8]%5 );
    if( ++IFcnt[5]%10 )
    {
        d = (c-j)%100;
        l += (n-h+m-l+h*e+d+l)%100;
        b -= (j-i)%100;
        h -= (d-b-l+i*c*b-f-
g+d+i)%100;
        f -= (a-i-c*c-i*m+c+b)%100;
        l -= (e-f*d*h*i-e+f)%100;
    }
    i += (f*g-j+a-n)%100;
    for( ; ++FORcnt[12]%5 ; )
    {
        k += (a+f+a+f*i-a-
m*e+j*i)%100;
        f += (k+a-l*h*g-b-c-l)%100;
        f += (e*k)%100;
        k -= (k-c-b+l+e+d+c)%100;
        l -= (g-e+m+m+a)%100;
    }
    else
    {
        while( ++WHILEcnt[7]%5 )
        {
            f -= (m+c+n-k-n+j+f+c-a-n*a-
a*b)%100;
            g -= (i-j+c-a+a-c-d)%100;
            f -= (j+g+i-k+c-k+n+c+j-j)%100;
            d += (e*i+e-j+h-n-m)%100;
            f = (d-f+m*f-m-l+a+a+c-e-h-
m-i)%100;
            e += (l+e-c-a+l+j+f-m-j*i+j-e-
l)%100;
        }
        do
        {
            l += (j*g*g-g+g*k*e)%100;
            e += (c+n+i-d+n*n+j-n-
k+e)%100;
            d -= (a*d-d*n+i+c-g*k+i-
i)%100;
            l -= (b-g)%100;
            i += (b+l-i-c)%100;
        } while( ++DOcnt[7]%5 );
        for( ; ++FORcnt[10]%5 ; )
        {
            h += (j+k+a-e+j*n+n-n-f)%100;
            g += (k-m*j+c)%100;
            f -= (m-a+a*a-c)%100;
            c -= (e*j-a)%100;
            c += (e-e*a+h)%100;
        }
    }
}

```

```

    for( ; ++FORcnt[11]%5 ; )
    {
        j += (d-n-j+i-m+l-b-f*k)%100;
        c += (a*f)%100;
        n += (l+m-g-d*m)%100;
        j += (e+h*j-j)%100;
    }
    }
    h -= (f+i+f+m-j*b-m+b+b-
f+j)%100;
    i += (d+j)%100;
    d -= (f+g-l-i+n-i+i+g)%100;
    g += (m+i*d*m-l+b)%100;
    }
    h += (a+k)%100;
    c -= (k*d+f-j*j-n+j+b+b*k-k-c-d-
l)%100;
    }
    break;

    case 2:
    {
        n -= (d+i+f-g*n-d)%100;
    }
    break;

    default:
    {
        n += (c+h)%100;
        b += (h*f-k+k*m+h-n+n*c*i*d-i-f-
n)%100;
    }
    }

    c += (c-f-d-h+e+f+d*j*a)%100;
    e -= (c*i*f-d+k*g-m+b-d)%100;
    f += (c-j*g+h-g-h-j+g)%100;
    h -= (i-h-f+m-f+l-a-h)%100;
    }
    c -= (m+d)%100;
    k += (h-g)%100;
    n -= (f-e-c-g*g*j*c-l+a+b-n)%100;
    k += (h-l+l)%100;
    e += (j+c-k*f)%100;
    d -= (g+i+b+k+i+a+m*a)%100;
    b += (l+a-j)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F5(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;

    switch( ++SWcnt[4]%3 )
    {

    case 1:
    {
        if( ++IFEcnt[11]%2 )
        {
            h += (h+j)%100;
        }
        else
        {

```

```

while( ++WHILEcnt[10]%5 )
{
    do
    {
        for( ; ++FORcnt[14]%5 ; )
        {
            l -= (n+l+j+l*d)%100;
            l -= (d-i-b-n-l-m+g)%100;
            a = (n*b+h*i+k-m-d+k+m-
n+l+k-j+k)%100;
        }
        if( ++IFcnt[7]%10 )
        {
            g = (m+f-k+k+c+f)%100;
            h += (j-h*m+b)%100;
            k = (h*m)%100;
            l -= (c+l*n-g*i-b-d-k+e+n-
e+d)%100;
            e -= (i+m*b+d+l+d-f-l-
k+l+d)%100;
        }
        for( ; ++FORcnt[15]%5 ; )
        {
            e -= (k+b+m+b+d)%100;
            f += (n+m+l+d+b*i-l+d+b+n-
b)%100;
            k -= (j-d-a+i+n*h*d)%100;
            b -= (c+c+b+i-f-d-c-a-l+b*l+m-
k+d)%100;
        }
        if( ++IFcnt[9]%2 )
        {
            d -= (l-c*b*k*a-a*f+n+m-j+j-g-
j)%100;
            l -= (d*d-a+n*b*f)%100;
            i += (k-h+f+f*g*k-j+k-i)%100;
            f -= (k+k+k+a*e-k+i-j+k-c-j-
n+h)%100;
            e -= (i-k-k-m-c)%100;
        }
        else
        {
            l -= (b+h-d-d+f-d+e*h-k-
e)%100;
            a += (l-a-i+g+n+i-c-i-n*f-
g)%100;
            h += (g-e*k-l+f*d-l-d-m-i-k+i-
h)%100;
            n -= (j-g+l+a+i+d+c+l+d-
m+h)%100;
            n -= (f+i*d*n+m+n-i+i-h+c-
j)%100;
            l -= (j*g+m-c-c+j*g+i*f+l-n-
d+n-l)%100;
        }
        while( ++WHILEcnt[9]%5 )
        {
            k += (b-a*m-k-e-e+e+c-
b*h+k+f-d)%100;
            a += (e+d+b+e*k)%100;
            e = (d*d)%100;
            f += (m*m-f+a*l+h+j+i-
h)%100;
            f -= (f*i)%100;
        }
    } while( ++DOcnt[9]%5 );
}

```

```

do
{
    h -= (f-a+k-c+g-f-c-h-i-j)%100;
    for( ; ++FORcnt[16]%5 ; )
    {
        h += (n+h*h+b+f)%100;
        e = (f+a+c-f-f)%100;
        i -= (d+l-i*f+k-f)%100;
        b -= (e+j)%100;
        k -= (l-f+f+l+a+l)%100;
        d = (j+e*m*i-a-c+g-
c+j+i+m)%100;
    }

    switch( ++SWcnt[5]%3 )
    {
        case 1:
        {
            h += (j-f+l+c-a-h)%100;
            k -= (l*h)%100;
            k -= (l*f-d+d-b*g+h-j-
e*c+g)%100;
            n += (k+k-d+c+f+k+i+a+g-
h)%100;
            i -= (h-n*g*b+d+m+n+a-a-
m*e-i*e)%100;
        }
        break;

        case 2:
        {
            m += (a-f-f-e)%100;
            f -= (a+k*c+l+f*c-n)%100;
            f -= (i+i+a+n-d-c+b*b-c-
j+a+m)%100;
        }
        break;

        default:
        {
            a -= (c-c+i-g-l)%100;
            f += (f-d)%100;
            j -= (b-n*d-e-c-e-d-n*f)%100;
            h += (h*n+f-b+n)%100;
            i += (i*d)%100;
        }
    }

    if( ++IFcnt[8]%10 )
    {
        j += (j-f)%100;
        c -= (i*l-h-h*)%100;
        l = (g+k-c+g*b+k+a)%100;
        e -= (i-k+m+j-
n*f+f+d+g*d*h+n-a)%100;
    }
    while( ++DOcnt[10]%5 );
    if( ++IFcnt[10]%2 )
    {
        l -= (n-e+l-n+m+l+h+l-c)%100;
        f -= (h-i-n*d-l*i*a-
f*m+a+h)%100;
        m -= (j+b*m-h-e*i+b*i*c-
l)%100;
    }
}

```

```

    g += (h*e-i+g-i*h+h-k-
b+c)%100;
    k = (g*i*l+n-b)%100;
}
else
{
    j -= (n*g)%100;
    e += (b-c)%100;
    e -= (f*a-k*h-l*b-i-d)%100;
    f -= (b+j+d+g-d+a-k)%100;
    d = (d-g*k)%100;
    g -= (h-h*c-k-d-f-
e+b+i*g*i*h+l)%100;
}
g -= (k-i+d+d+j+g-m+e)%100;
f -= (m+m-k)%100;
h += (c-k+d+c+n+l+e-f-e+i-
m)%100;
}
d += (a-j-b+n)%100;
a += (a+e-i-b*j-d*k*e)%100;
e += (k+l+i-f)%100;
}
c += (g+i-g-e+m-j+f*l-a+a-d+f)%100;
n = (g+k)%100;
f = (e-j+k*l)%100;
a -= (a*f+e+i*j+b+c-j+a*f+m-f)%100;
}
break;

case 2:
{
    d -= (j-n+d*j-f+i-b+h)%100;
    k += (k+l+a+i*g*g*b+c+g-m+c-e-
n)%100;
}
break;

default:
{
    g -= (b-a-k+n*c*d)%100;
    n += (c-k-e-m*i)%100;
    k = (f+c-n-i*j+h+l*m+j*c+i-
n*m)%100;
    b += (n*a*f+m-j+g*b)%100;
    j -= (i-c-b+d+c-h+a+e+d+j)%100;
}
}

f = (g*l-i+m+i+m+g+e-e*k-n-g-
g+l)%100;
c += (h+n-l*f+i*d+f+c-l-h)%100;
g += (a-b+h-m+n+b+n)%100;
l = (n-i*d+n+g-k-n*n-i-n*c+g-g-
m)%100;
c -= (m-m-h+n-d-g+f+d-j)%100;
n -= (i-h+j-j-e-h)%100;
f -= (n+g-i-n)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F6(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
}

```

```

while( ++WHILEcnt[12]%5 )
{
    do
    {
        for( ; ++FORcnt[19]%5 ; )
        {
            for( ; ++FORcnt[17]%5 ; )
            {
                if( ++IFEcnt[12]%2 )
                {
                    m -= (d-j-f)%100;
                    f -= (l+d*k-d*d-b-d)%100;
                    d += (j*j-
g*b*h+g+g*m*g)%100;
                    h -= (e-b-a*i)%100;
                }
                else
                {
                    g -= (n-b+j+k-k-
n*m+l+g+h+d*b+m+b)%100;
                    g += (d-g-b+b-j*j*a+f)%100;
                    i -= (g+d)%100;
                    j += (g+f+b+i*k-k+m+k-
m*h+h-e*n)%100;
                    l = (n-j-j-i-j-e-a*h-k-c+f)%100;
                }
                while( ++WHILEcnt[11]%5 )
                {
                    n -= (k-g-d*k-m-a+e*i|-
m*i)%100;
                    b += (l-a+n+n+b-c-
c+j+k+j*i+d)%100;
                    m += (f+i*k-m)%100;
                    d -= (g+l-h*e-c-n-e)%100;
                    m -= (j-k+e-f-n-e+i)%100;
                    d += (h+a+h-d*k*b+i-l)%100;
                }
                do
                {
                    b -= (c-c+h+m+a*d+i-l*j-a-
c*i*c)%100;
                    e = (j-k*k+b-a-
d+l+c+b+b)%100;
                    a = (n-f*h+g-b+b+a-a+m-
l)%100;
                    e -= (h+a-m+c-b+k*b+j-
n)%100;
                    d += (a+f-a-j)%100;
                } while( ++DOcnt[12]%5 );
            }
            if( ++IFcnt[9]%10 )
            {
                h -= (c+e*h-d-i*j+d)%100;
                for( ; ++FORcnt[18]%5 ; )
                {
                    h = (i+d)%100;
                    n -= (e+i-g-e+g+f-h-m-c+k-a-e-
l)%100;
                    b -= (b+b*k)%100;
                    h += (j-j+f*c-k+k-n)%100;
                }

                switch( ++SWcnt[6]%3 )
                {
                    case 1:

```

```

{
    b += (b*k*b-f*e)%100;
    j -= (e+h)%100;
    c -= (m*a+e-n)%100;
    d -= (m-a*m-m-e-e*n+c+a+c-
d)%100;
    k -= (a-j+h-l-k+b-e-h+f-m+j-
b+d)%100;
    g -= (a*n+l-h-g+a+e-
h+i+h)%100;
    }
    break;

    case 2:
    {
        h -= (a+e*d-l+f*g+j-
d*j+e*g)%100;
        k += (c+b*f)%100;
        n += (d+l*b*e-c+i*a-
e+i*d+b+b+l+e)%100;
        n -= (m-k-d+b-k+e-c*n-
g)%100;
        b += (a-
c*f*d+h+j+c*h+a)%100;
    }
    break;

    default:
    {
        k += (i*|n+m-g)%100;
        e += (i+i-i-n-f-i*h+m)%100;
        e += (i*h*a*g-n+c-l+k+k-
f*|i)%100;
    }

    if( ++IFEcnt[13]%2 )
    {
        b = (f*j-h)%100;
        g += (f*e+n-e-m*c*m+m-
d)%100;
        m += (b+e+e+l+i-m*k+k-n-
b+j)%100;
        k += (m+k)%100;
    }
    else
    {
        a += (d+i*c*m+e+n+m-
h)%100;
        c += (e+k+h)%100;
        k += (e-n*b+e-h+n+f-g+a+d-k-
f+c)%100;
        l += (b-d+b+m-n-j-m)%100;
        c += (n*k-b*e*b)%100;
    }
    k += (j-b-a-e-e-c+g*d-i+d*n-
e)%100;
    }
    h += (e+j-l*c*k-k)%100;
    h = (i-c-c-b-c+h+i-i+i)%100;
    f -= (j+g+f)%100;
    }
    i += (a-d-i-e+f+n*n*b+a-k-f-
n+c*a)%100;
    c += (k*d+e-i*i+a)%100;

```

```

    f -= (i-n+n-l-g+f+f-h-m-b+l*c-
h)%100;
    n -= (c-g)%100;
    l += (i*|*k)%100;
    } while( ++DOcnt[11]%5 );
    l += (e+i)%100;
    g = (j-e-n-k-h)%100;
    g -= (f-g+g)%100;
    }
    l += (a-k)%100;
    e += (k-e-h+f-c*|*i)%100;
    j += (i+h*h-b-e-j*i-j*a-n+b-b-i)%100;
    h += (c-i-d+a-j+d)%100;
    j += (h-i-e*g+i*d+l-g-a)%100;
    l -= (e-h-b+b*h)%100;
    i = (j*c)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F7(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[15]%5 )
    {
        do
        {
            for( ; ++FORcnt[21]%5 ; )
            {
                if( ++IFcnt[10]%10 )
                {
                    for( ; ++FORcnt[20]%5 ; )
                    {
                        n += (b-a+f+n+k+j+n+j+e*d-n-
h)%100;
                        m += (e-i-n+m-f+e-a+e+k-c-
d+h-a-c)%100;
                        h += (g*f-k*b-g-m-g-c+i-
f+c+m+c)%100;
                        m -= (a*m+a-m+e-g-c-j)%100;
                        l += (c*g+a)%100;
                    }
                    if( ++IFEcnt[14]%2 )
                    {
                        m += (h*k*b+a+e*m-c-
a+e+e*h)%100;
                        m += (i+i-a-g+m-f+d-h-m-h-
m+e-m)%100;
                        g -= (c*m-l-c*i*|+m-b-j)%100;
                        n += (l+m-f+h-b*k-c-c+m*b-n-
c+l+e)%100;
                        l -= (l+f)%100;
                        j -= (e*j-n-h-g-j*d-
j+f+a+b)%100;
                    }
                    else
                    {
                        f -= (e-c+m-f)%100;
                        j = (k+i*i*j+c-h*c+e-h*c-
g+k)%100;
                        d += (i+g*d-i+g)%100;
                        g -= (l*b)%100;
                    }
                }
            }
        }
    }
}

```

```

while( ++WHILEcnt[14]%5 )
{
do
{
j -= (a-i+j*h)%100;
for( ; ++FORcnt[22]%5 ; )
{
n -= (f*k*|+e-a*m+h+d)%100;
g -= (g+j+k-a*b*k-a)%100;
f = (i+e)%100;
n += (n+k-m*d-g-a-m)%100;
h -= (d*f*f*e-f-a+d-a-a-k-h+c-n-
i)%100;
}

switch( ++SWcnt[7]%3 )
{

case 1:
{
l -= (l+d*n*f*d*a*m)%100;
i = (m*d+c+k-d-m-f*c-g-
l)%100;
k -= (c*h-a*d-i*n+b*k+b-d-
f)%100;
b -= (k+m-a*d-i-k-m-k+l+i-
i)%100;
}
break;

case 2:
{
a += (d*j+f-a-b+m-m+l+f-
g)%100;
b -= (n+i)%100;
m = (j+l+f*a+k+a+k-g*b-
a*k+c-h)%100;
a -= (a-m+l*f+b*a)%100;
n = (n+f+g)%100;
h += (k-i-m-e-l+a+g-
i+d+h)%100;
}
break;

default:
{
k = (b+c+b*|+d-b*n*k+n+i-
c)%100;
f -= (g+b)%100;
f += (m-h)%100;
i += (j+f*b-g+c-g*n+e-
a*n)%100;
k += (e*b)%100;
}
}

} while( ++DOcnt[14]%5 );
if( ++IFcnt[11]%10 )
{
if( ++IFEcnt[15]%2 )
{
b += (l-b*j*h-g-n*j+c*h-
j)%100;
k -= (f+e+n-b)%100;
h += (j*a+e+a+c-h-f-m)%100;
c += (e-d+j)%100;
}
}

```

```

f -= (m-b)%100;
d += (c+n-a+k*n+k-f-l-h-
n*a+l)%100;
}
else
{
d += (k-m)%100;
k += (i*j)%100;
d -= (c+m-k+g-c+c+d+e+k*k-
c+c*c)%100;
i += (k+e+i+a+k*h+n+g)%100;
}
while( ++WHILEcnt[13]%5 )
{
d += (f-a+e*b+f-n+l+h-n-
m)%100;
f += (f-l)%100;
a += (f+c+m*k+i)%100;
m += (a+d+h+f+j*a-m+f-h-
h*c)%100;
m -= (g*m*g+a-i+k-i*a+m-
i+d)%100;
}
d += (j+n+i-k*b-a+f)%100;
j -= (f+i-i-f-i)%100;
n += (h+c)%100;
}
m += (g+e*m+k*d-i-e-n)%100;
f += (h-c-j-n-f-l-l+n)%100;
f -= (j+g+a+b+i-a+l*j*j+g+i)%100;
}
m -= (j-k*m*c-j-c)%100;
j -= (n*a*d+f+j-d-h+d-i+j-
m+a)%100;
a += (g+m+n+l+k-d*c-i*n-f)%100;
} while( ++DOcnt[13]%5 );
g -= (i*d+b+f+n*d-d*f-f-c-a)%100;
h = (k*a+d-h-f)%100;
m -= (j+n+f+h+j)%100;
h -= (n-g+h+a*h+h+b-i-l-d*a)%100;
i += (l*f-g-e-d-k*a)%100;
}
a += (i-n+m*e+n-b-e*i-a-m)%100;
b += (c*j-d+d+b+j-k+n)%100;
a += (e*h*m*m*e+f-m)%100;
g -= (e-a)%100;
g = (i+e*c+a-m+f-l-h*g*j-m)%100;
k += (n*f-c-d+d*c*i*n+n)%100;
n += (n-a+c+d-m-i-m+n+g-n)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F8(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
do
{
for( ; ++FORcnt[26]%5 ; )
{
for( ; ++FORcnt[25]%5 ; )
{
if( ++IFEcnt[17]%2 )
{
if( ++IFEcnt[16]%2 )

```

```

{
n = (i*k-d-j-f+c-
m+g+k+b*n)%100;
f -= (g-i*g*|*l+f+j+b-
i+m*a)%100;
e += (l-l*a+a)%100;
e -= (m+n*m-l-m-m-
c+m+e)%100;
}
else
{
b += (d*j-k*i*c+c)%100;
c -= (c-j+d)%100;
j += (a+m+a-j*c+k-g+e-f-
i)%100;
l -= (a*g)%100;
c -= (c+f+a)%100;
}
while( ++WHILEcnt[17]%5 )
{
k += (m-h-d+j-a-h*m+i-m-
k*c+j-n*f)%100;
j += (i+i+h-a*|*l-m+k*m-
g+k)%100;
f += (a-b*i*b+a-
e+j*e+n)%100;
c += (i-f)%100;
f -= (d+k-k+l*b)%100;
a -= (n*g-k+n-
b*h*c+i+m*g+f*i*i)%100;
}
do
{
a -= (i+c-d-f+d*d-d+d+e+j-i-
k)%100;
d -= (l-d+j+n)%100;
i += (c*i-h+b-k-b*b*j+l+n-
c+k)%100;
b -= (g-n+i)%100;
g = (l-d+h*j)%100;
} while( ++DOcnt[17]%5 );
}
else
{
while( ++WHILEcnt[16]%5 )
{
d -= (b-f)%100;
g -= (b-n)%100;
f += (b*b-h-k*n*d-i*h-i-
d+c)%100;
e -= (f-b+l+g+j+l+b+j)%100;
e = (l+b-b-j)%100;
}
do
{
b -= (k+e*i)%100;
} while( ++DOcnt[16]%5 );
if( ++IFcnt[12]%10 )
{
j += (k+k-i+m-n-a)%100;
l -= (c*|)%100;
}
l += (m-d-a+d*h-l-g+h-i-b-
m*b+k-k)%100;
for( ; ++FORcnt[23]%5 ; )
{

```

```

        d -= (b+k-g)%100;
        j += (e*d-d-l+m-
c+e*g*b+l+e+h)%100;
        h += (n+b+j+e)%100;
        d =
(f*i*e+a*b+b*c+a+g+e*c+a+m)%100;
        e += (d+e-l+k*e*m-j)%100;
    }

    switch( ++SWcnt[8]%3 )
    {

        case 1:
        {
            e -= (d-m-e*k-j-d+f+n-
k+d)%100;
            l -= (a-m*l*e+f-d-c+f+l-
n)%100;
            h += (a-f*j+g+h-m-j+n*n+i-g-
j)%100;
            m = (m-l-m+c-n+g+g+n+i-c-
e+b+l)%100;
        }
        break;

        case 2:
        {
            d += (f-e-i*g-c)%100;
            k -= (f-d-b-h-e-i+i*h)%100;
            c -= (a-g+c-b-b-j+l+c+l)%100;
            c -= (h-n*l*a-j-d-a-g+b)%100;
            f -= (l-h*l*a+f+f+d+n-f-b-
k+c+a)%100;
            d -= (f+b-m)%100;
        }
        break;

        default:
        {
            l += (l-h-j+j)%100;
            l -= (j+i-a+b-a*l+c+g-m-d+m-
a)%100;
            e += (l+f+c+e-h)%100;
            e += (e+f-e-j+e*l+e+b)%100;
            a = (b+j+m+i+n+j+b+l-b)%100;
        }
    }

    for( ; ++FORcnt[24]%5 ; )
    {
        f -= (h+l)%100;
        g += (b-e*c*g+h+d*b)%100;
        l += (f+g+g+h-n*i+n-h)%100;
        l += (c-e)%100;
        n += (d-g+j+l*i+l+b*i-c)%100;
    }
    c += (l-k*k+j-d+k-j)%100;
    d -= (n-g*g+k+f-n-n-
i*b*g+g)%100;
    }
    h += (m+g-m-k)%100;
    c -= (g+b+j*j+f-l+h*j-j-l)%100;
    h -= (d-c)%100;
    a += (j-c*k-k)%100;
}

```

```

        c += (a-d-c)%100;
        n -= (e+c-f+n-e*a-g-d)%100;
    } while( ++DOcnt[15]%5 );
    g -= (g-l)%100;
    d += (h+n+h)%100;
    b += (h+h-f*j)%100;
    i -= (b+n-c+d-d*j-h-b-l-g+i+n)%100;
    n += (i*h+m*m*f-i-b)%100;
    n -= (g*b-k-e)%100;
    i -= (f+f+h)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F9(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[14]%10 )
    {
        for( ; ++FORcnt[30]%5 ; )
        {
            if( ++IFEcnt[20]%2 )
            {
                b += (l*d-a+n*i-l-h*k-c+g*l-
g)%100;
                g += (b+h-g)%100;
                k += (k+g-m+i-l+l+e-j*m+g-
k*d)%100;
                n -= (e+a-b*i-a)%100;
                e -= (k+j-f+i-d-h*b-h-l)%100;
            }
            else
            {
                while( ++WHILEcnt[18]%5 )
                {
                    do
                    {
                        c -= (d*f-e-i+m)%100;
                        f += (k-l-m+k*n+k*k-c+a+c-
n*g)%100;
                        g += (k+l+k+n+h+c-a-f-d)%100;
                        b -= (b+b*h-i*k-j*a+a+m-
m)%100;
                        k += (m-l+d-c+n+h+d-n-j*j+f-
a)%100;
                    } while( ++DOcnt[18]%5 );
                    g += (k*n*e*e)%100;
                    for( ; ++FORcnt[27]%5 ; )
                    {
                        d -= (i-d-a-c*n*b-h)%100;
                        g += (k-g+f-m-g-m)%100;
                        h += (m-d-b-k*n+c-h+k-d-d-
h+f+e)%100;
                        m = (b+f+j+m*g-a+d-k)%100;
                    }
                }
            }

            switch( ++SWcnt[9]%3 )
            {

                case 1:
                {
                    if( ++IFcnt[13]%10 )
                    {

```

```

                        j -= (i-k-d-j-n-g+h-j*a+i-l-
a)%100;
                        g += (a-k+j-j+d-l)%100;
                        e = (h+k-j-g)%100;
                        f -= (f*l+n-h+e)%100;
                        m += (j*l+l-l*j)%100;
                    }
                }
                if( ++IFEcnt[18]%2 )
                {
                    d += (e*g+m-g+g+j)%100;
                    h += (b+j-e-k-i-a-c*d+m-f*f-
e)%100;
                }
                else
                {
                    m -= (j*c+c+j*b)%100;
                }
                while( ++WHILEcnt[19]%5 )
                {
                    e -= (b+h-a+a)%100;
                    m -= (i+c+c+h+m-g-b-
a+d*a)%100;
                    j += (n-e-i+c-c-j+c+c-m-c-
e)%100;
                    g -= (h+j-e*b-f-h+e-f+e-g+i-
j+i*h)%100;
                    b += (k-m+d+d-h)%100;
                }
                do
                {
                    l += (d+e-j-k-l+j)%100;
                    l -= (k-c+g+m+k*c-e*i)%100;
                    c += (j+l+c+a-i+k+f*b+d-k-
d)%100;
                    a += (l*n-e-h-k+f*f*b-k)%100;
                } while( ++DOcnt[19]%5 );
                for( ; ++FORcnt[28]%5 ; )
                {
                    h -= (f*f-c+e+h+l-f+k+j+g-
d)%100;
                    l -= (i-m)%100;
                    l -= (a-c+l-f-d-i-a+m+m-
c)%100;
                    e += (h+f+l+e+a-d*j)%100;
                    a -= (d-i-h-c-c)%100;
                    i += (b-n+a+g)%100;
                }
                for( ; ++FORcnt[29]%5 ; )
                {
                    n += (h+k-f+m+j+e+m-e)%100;
                    b += (d+m-l-c-k+a-b)%100;
                    c += (g+g-a-g+g-j+h+d-
e+n)%100;
                    h = (f-e)%100;
                    l = (j+n+b*c+m+j)%100;
                }
            }
            break;

            case 2:
            {
                if( ++IFEcnt[19]%2 )
                {
                    g -= (n-n+m+i-c-n)%100;
                    l += (n-m-c-h-m)%100;
                    h -= (a+l+a+k*d*a*g)%100;

```

```

        d += (f*h)%100;
    }
    else
    {
        b -= (n+d+b-n+e-c)%100;
        m += (b-i+g+i+l+f*k-k)%100;
        k += (f+c-d+d+m+g-d*b)%100;
        d += (c-m-j+e+h*a)%100;
        m += (j-n-d+e*m+e-d*a-g-a*a-
l*c-m)%100;
    }
    while( ++WHILEcnt[20]%5 )
    {
        h += (j-k-d)%100;
        j = (k+c*j+b)%100;
        n -= (l-e+n*f+h)%100;
        n += (c*l*b+k-c)%100;
        h -= (h-b-h+g-c*a*m-k-
c)%100;
        e -= (g+i+k+n*j-n-k)%100;
    }
    f -= (h+c)%100;
}
break;

default:
{
    a += (c+m-n)%100;
    m = (n+k+f-h-j-j)%100;
    a += (d+b*c-j*d+i+j-k+k*l+g+b-
b+a)%100;
    f -= (c+n+j-b-a*e-k)%100;
    f -= (n-h-i-b-h-i-a*a-d+f+i)%100;
}
}

f -= (h*m*g+n+l+k-l+f)%100;
b += (b-n-c-m*f)%100;
n += (l+j+f+d-k-f-b-n+j)%100;
}
i = (m-f-i-e-e+h-j+c+b*j+h-
k+e+d)%100;
j += (d+c-i-d+a*m-l-c+a)%100;
f -= (h-c+n+k+f+f+g)%100;
g -= (g+j+l*n-f+n-j*j+l-c)%100;
b += (g-l+f+f)%100;
}
b -= (j+c-l+h)%100;
g = (n+n+j*n*a-i-b)%100;
d += (g-k-d+f-j+b+l)%100;
}
d -= (i+c+b*c-h*a+c)%100;
d += (n-m+e)%100;
m -= (m+l+c-h*m+h-f-c+m*c)%100;
f -= (m+d)%100;
k += (d+c*m)%100;
j = (e*i-m+k+a-j-k)%100;
b += (h-c)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F10(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;

```

```

do
{
    if( ++IFcnt[16]%10 )
    {
        i += (c+e*c-a+j+e-c+f)%100;
        for( ; ++FORcnt[33]%5 ; )
        {

            switch( ++SWcnt[10]%3 )
            {

                case 1:
                {
                    if( ++IFEcnt[21]%2 )
                    {
                        k += (f*e+i+h)%100;
                        h -= (i+h+l+b-k-b+j*a+g-
g*c+k)%100;
                        d = (j-j+c+j)%100;
                        e += (f-j*g-e-f-n-i+d+h+i+l-
d+c)%100;
                    }
                    else
                    {
                        n -= (m+d-i-i+j+i-n-h-
n*e+i+g)%100;
                        l -= (h*a*m-e-h+h*i+j)%100;
                        e += (e-j-h*k-i*e+e)%100;
                        k += (a-e+k-j-m-f+i+j+j)%100;
                        j -= (g*j-a-g*i-j-h*f)%100;
                    }
                    while( ++WHILEcnt[21]%5 )
                    {
                        j -= (n-h+b*l-e)%100;
                        f -= (c-k-d*g-a+e*f)%100;
                        h -= (m-e-m+a-l*h+m+k+k-
l*d)%100;
                        e -= (g-n+a-i-j-g-m-k-e*a-d-
m)%100;
                        c -= (c*m+h-f*h-
e*m+j*d+b+a+b)%100;
                        h -= (g-f+b-l-c)%100;
                    }
                    do
                    {
                        b += (b-m+g+h+j-h+b-a)%100;
                        j -= (c-h+m+d-f*f+k-j+c+e-
k*a)%100;
                        g += (d+j-h+e+d-d-l*b-f)%100;
                        b -= (h+b-h*j+n+i*d*a-m+k+b-
a-l)%100;
                        k -= (g+g+h-e+j-i-h+d-
e+l+l)%100;
                    } while( ++DOcnt[21]%5 );
                }
                break;

                case 2:
                {
                    for( ; ++FORcnt[31]%5 ; )
                    {
                        e -= (j*i*f+g-k-d+b-m+e)%100;
                        g += (n*h+j-m)%100;
                    }
                }
                break;

```

```

            default:
            {
                if( ++IFcnt[15]%10 )
                {
                    n -= (f*i*d-f)%100;
                    b += (d-b-n-b)%100;
                    i += (i-b-j-h*i)%100;
                    m += (m+j*n-c-n*e+d*c-g-g-
c+n+e)%100;
                }
                for( ; ++FORcnt[32]%5 ; )
                {
                    l -= (d+l-h*k)%100;
                    e += (n+f*j*g+j+l)%100;
                    l -= (b*j-k+n*k)%100;
                    n -= (k-d)%100;
                    c -= (f*c+c-m*n-a)%100;
                    m += (c+e+d+l+a+l+g)%100;
                }
                if( ++IFEcnt[22]%2 )
                {
                    i -= (c-e*d-i*j)%100;
                    k -= (l+a+h+c*c-c-a)%100;
                    m += (k-f*l-b-g+h*c+k-i-
c+c+l)%100;
                }
                else
                {
                    d += (j-a*h*d+d)%100;
                    a = (d*e+c+j)%100;
                    f += (m-j*h)%100;
                    h -= (e-d+l+i)%100;
                    n = (i+i-i-g*n*k-l*f+e)%100;
                }
                while( ++WHILEcnt[22]%5 )
                {
                    h += (b*i-d*e*c)%100;
                    n -= (a+n+g-a-m)%100;
                    f += (c*g+m+k-i*g+h-m)%100;
                    g += (a+c*h-h-c+j-b-e-
n+b*c+l)%100;
                    m -= (f*m)%100;
                }
                f += (f*i*h+l+c-l*d-k)%100;
            }
        }
        g += (e+i*b-g+c)%100;
        n -= (j*d-h-a+a)%100;
        c += (n-a+k-c+f)%100;
        m -= (i+c-k)%100;
    }
    d -= (k+g+e+l+e+e+d-e+d-
g*e+g+h)%100;
    b = (n-a+m+f+a*m+c*j-f+l+l)%100;
    c += (a-i+j-m+m+f+a-h+a+f*k+c-
d)%100;
    m -= (f+m*j*l*n)%100;
}
f -= (d+c-a+h*a-k-j*g)%100;
b = (c+a-m+l-a)%100;
n = (h-h+f-c*e*n-n+a+g-e)%100;
} while( ++DOcnt[20]%5 );
h += (n-a-m+h-b+k-b*g)%100;
i += (k+h)%100;

```



```

m = (d+k+d+i+i-d+k*i-m)%100;
m -= (l+h)%100;
i += (i-k-n-l-e+b+k+n+j)%100;
g -= (j+f*c*c*k+b-l-i*j+c+h-
g*h+e)%100;
e -= (d-f*n+m+i-g-g*f-
n+m*d*d+j)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F11(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
do
{
h += (h+c*a-a-d*n)%100;
for( ; ++FORcnt[37]%5 ; )
{

switch( ++SWcnt[11]%3 )
{

case 1:
{
if( ++IFcnt[17]%10 )
{
if( ++IFEcnt[23]%2 )
{
e += (g-k+d+b-h-h-d-a+f+n-
h+m)%100;
i += (j-m-j+k+n+f+n+n)%100;
j += (f-j-b)%100;
c += (k-f-l+m+i-d-h+f-k)%100;
b = (g+h-l+e*b*i)%100;
b -= (g-e+m)%100;
}
else
{
f -= (g-c+c-a-g+g-g+j-i)%100;
e += (e*f-m+f+k*h-c+e*i-l-n-h-l-
e)%100;
k += (l+k)%100;
f -= (h-d*a-h-k-f+n*a-h)%100;
}
while( ++WHILEcnt[23]%5 )
{
i += (a+e-c*m+l-c-g-g-f+h-
f)%100;
l -= (n+a-l)%100;
f += (b+e*c)%100;
h -= (i*a-c)%100;
k -= (d+i-l*k-e+i-g+g+a-j+g-
g)%100;
}
do
{
k += (f+b)%100;
e += (n-f-g-n*j-m+j+d-j)%100;
i -= (b*k-m+l*a+b-a-c*k*e-
a)%100;
} while( ++DOcnt[23]%5 );
for( ; ++FORcnt[34]%5 ; )
{

```

```

m += (f*c*c-g-b+d-f+m-
h)%100;
g += (k+b*g+g)%100;
m -= (g-k+d+e-i*j-
m+c+l*f+a+f+e)%100;
i += (c*c+k*d*k-
d+g*c+m)%100;
c -= (j*m-l-c-f+g)%100;
}
for( ; ++FORcnt[35]%5 ; )
{
m = (e*m-i)%100;
e += (c-a-m*i+m+l+j-a-g-
c+c)%100;
e -= (k-k)%100;
i -= (e-b+g+e+d+m-n*f*m-
h+h-n)%100;
}
if( ++IFEcnt[24]%2 )
{
k -= (m-i+e+b-l+b-f)%100;
c += (m*f*h*m*h)%100;
m = (f*e*k+e-m-g*n-c-a*f-h-
m)%100;
}
else
{
while( ++WHILEcnt[24]%5 )
{
d += (k+a*k*d-c-n*h-g)%100;
f += (c-e-c-a-g+f*m-a-
l+d)%100;
e -= (d-j-j+e+b+f-k-k+h+d-
i*f)%100;
n -= (n+n-b-d-h-m)%100;
l -= (h+g-k+h)%100;
}
do
{
m -= (g-n+a)%100;
} while( ++DOcnt[24]%5 );
if( ++IFcnt[18]%10 )
{
m = (h-a+m-g+b+f+h-f)%100;
d -= (n+d*m)%100;
}
d += (b+l)%100;
for( ; ++FORcnt[36]%5 ; )
{
g -= (g+i+j)%100;
k -= (a-i+l+k+j+k-h+n-
h+i+g+c)%100;
e += (k+l-h+n-d-e)%100;
f -= (k*n+i-j+m*h+n-b+m-h-
l+n-k*j)%100;
f += (i-c)%100;
}

switch( ++SWcnt[12]%3 )
{

case 1:
{
g -= (j+k+j*k+a-h*n+a+d-
g+d*i)%100;

```

```

l -= (f*f+n+a+g*h-b-m+d-
i)%100;
f += (h+j-k*h*g)%100;
n -= (d+l+d-i*c-a-k+e+f-n+l-l-n-
l)%100;
}
break;

case 2:
{
c -= (e*i*d-g+j+f+e)%100;
c += (i+g+k-m*f-m*i+l-k-d-
h*a+m)%100;
l += (b-c*b+e*i*n)%100;
a += (n*i-i+g+b*f)%100;
e -= (m-n)%100;
k += (e-b+i*d+d+e*d*j+m*i-
c)%100;
}
break;

default:
{
c = (g-c+b+k+f-j-n+k-m-
d*a+f*d-i)%100;
d -= (e-g-b*n+b+m)%100;
j += (l*b-c+g*n+l+e+g-c-j-d-
a+d)%100;
k += (e+n+g+b*a-k-n-b-m+g-c-
d+n+e)%100;
c -= (n-j+l+a-i+e-c-d-
e+e)%100;
}
}

k -= (g+m-j+b-l*g+a-h*j+b)%100;
c += (j+m)%100;
a += (h+m+d)%100;
}
break;

case 2:
{
b += (i+j-i-k-c-m-g-n-b-
f*e*m)%100;
j += (c*d)%100;
h -= (k-j+c+j-l+m*k+m-d+b)%100;
k -= (d-j+a+n+h+b-e-m*i)%100;
e -= (c-i-n*g)%100;
}
break;

default:
{
c -= (m+l-g+m+m-i-j+n)%100;
f += (k-h)%100;
f -= (k+b-h-d*n+e-a+a+k-
e*j*k)%100;
g -= (l-h+g-b+a-d*g-c+e+f*n-
h)%100;
}
}

m = (m-l+n+l-m*f)%100;
g += (b-d+b+c+b-h-k+e+l*i-e)%100;

```

```

a += (c+g+i+n+e*a*g+a+m-
j+b+b)%100;
c += (g-b-i-b-g+h+m)%100;
i += (m-h-d)%100;
}
i -= (n*c+g-i+c+g+m-e-e-f*e-k)%100;
e -= (j-n-c*f+a+i*m*h+e)%100;
} while( ++DOcnt[22]%5 );
f += (g-f-j+f-h+g)%100;
d -= (e*l-j-g-m*k-g-n)%100;
m += (n+k-l)%100;
k += (b-n*m-c-n+i*a)%100;
n -= (m-e*j-j-g+c+k*j-c)%100;
e -= (d+e+j-
j+a*b+c+n+h+g+n*d+e)%100;
h += (f-f+g+k)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F12(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
if( ++IFcnt[26]%2 )
{
h += (b-f-h-n-f-d)%100;
h += (h+a+a-l-h-b+e)%100;
i -= (k-d+c-d-j*f+n+e*l+g+a*j)%100;
n -= (b-k-g-a*b+n-f-n+d)%100;
f -= (h+j*n+c+f*g+h+f)%100;
c += (d-c-e+i-g-d)%100;
}
else
{
while( ++WHILEcnt[26]%5 )
{
do
{
for( ; ++FORcnt[39]%5 ; )
{
if( ++IFcnt[19]%10 )
{
l -= (i-h+c-f+g+f+d)%100;
i -= (g-c+d+n+k+k)%100;
d += (l-h-n*c-a*e+m-g)%100;
m += (c-h)%100;
i -= (n-m+b-g-m*l*e)%100;
n += (a-n-l-d+g+h-h*g*k+a+f-
d+i-k)%100;
}
}
for( ; ++FORcnt[38]%5 ; )
{
c -= (f+l+e*i-e*a+h*m*i+n-g-
d+d)%100;
l -= (e+g)%100;
j += (g+a)%100;
k -= (c*f-h-m+m+j-n-c-n-c+k-
h)%100;
k -= (g+c*k-b-j*h*e+a-i-
l*c+m)%100;
}
if( ++IFcnt[25]%2 )
{
g += (c-j+m-k-e+j-g+g+e)%100;
l += (m+j-n*d*f-f+b-n*k)%100;

```

```

e += (a-d+g-g-g-d-m+a*g*k-
i+j-j+a)%100;
c = (i-e-h)%100;
f += (h+a-k+e*a+k-b+e-
c+h+e)%100;
}
else
{
k +=
(g*k*m+c+d+g+f*g+h+g)%100;
l += (b*m+f*a-l+l-e+c+e)%100;
a -= (c+c-d-i-j*c+g-g+a-e-
a)%100;
}
while( ++WHILEcnt[25]%5 )
{
h -= (j+m)%100;
f += (n-j+i*h*l*h-e-h+m-
n+f)%100;
i -= (j+i)%100;
k -= (i-e+c-c-k*h-d+c)%100;
}
do
{
j -= (b*f*a*j+n+f)%100;
for( ; ++FORcnt[40]%5 ; )
{
j += (e+c)%100;
k -= (b-a+k*g)%100;
d -= (a-n-i-g-d)%100;
a += (d+b-d+d-k-d-l+m-a*a-
k*h*d+a)%100;
c += (n-j-g+c+a-l+h+n*k-i-
k*i+j+i)%100;
}
switch( ++SWcnt[13]%3 )
{
case 1:
{
k += (b+j+c-d)%100;
}
break;
case 2:
{
b -= (c+n+n+c+f-f+h+k-b-d-
c+m-i+c)%100;
c -= (a-m-h+i+c-m+k)%100;
}
break;
default:
{
g += (c+g-h+f*a)%100;
h -= (m+e*g+i-c-j*e+n-
b*d)%100;
h = (c-m+m-f-h-k+i+h-b-n-i-
d*e)%100;
d -= (d-d*k+g-e*h*b)%100;
h -= (l-n+l)%100;
}
}

```

```

if( ++IFcnt[20]%10 )
{
i = (f+g+j-f-f-a-a-c-g-h+n*c*c-
c)%100;
i -= (l-i-b+l)%100;
e -= (l*f-i-k)%100;
m = (n+l+c-i+g+k*f-n*h+a-e-
d-k-f)%100;
}
e -= (g+j+l-c+k+f+c+b*e-e-
h)%100;
m += (e+b-i-d+m*f-a)%100;
} while( ++DOcnt[26]%5 );
e += (m-h)%100;
d += (c*f-e-g-a*j)%100;
g += (h+e+b-l)%100;
} while( ++DOcnt[25]%5 );
c += (d-f+l-e*a-i+l)%100;
d -= (m-j+i-n+f+b-j-c-
i*m*h*i)%100;
j = (d*i)%100;
g -= (c*f+n-l+f)%100;
}
c += (i+i+f-g+g-n*j)%100;
f -= (h+m-b-e-b-j+h*l+f-e+e+e)%100;
m += (h+n-b*j-e)%100;
c += (h*j*a-m-g*b-c-
d*g*l+l+d)%100;
k += (h-a+f)%100;
}
c += (a+b-a-b-k*m*l*d+g+f*n-b)%100;
f += (e+g-n*n)%100;
f -= (n*f-c+m-j*d+n*h+e-k)%100;
i -= (h-b)%100;
m += (h+d+n-a+b)%100;
b -= (b+e+a*m+c-c-f+e)%100;
g -= (e-a)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F13(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
if( ++IFcnt[28]%2 )
{
n -= (d-j+j*b-e-g-c+h)%100;
}
else
{
while( ++WHILEcnt[28]%5 )
{
do
{
for( ; ++FORcnt[42]%5 ; )
{
for( ; ++FORcnt[41]%5 ; )
{
i += (n-h*k-n+c-f-i+b+j-m-c-
b+l*d)%100;
a += (g*k+j-e-n)%100;
m -= (b*c)%100;
j -= (n-j*m+n-c*d+g)%100;
a += (m-c-n*d)%100;
j += (l-b*h+k)%100;

```

```

    }
    if( ++IFEcnt[27]%2 )
    {
        e -= (l-a-c-l-i)%100;
        n += (j+d-l+j*c+k-b-l+f)%100;
        c = (h*m-e+h-g)%100;
        l -= (l-c)%100;
        d += (b+l+n+m)%100;
    }
    else
    {
        d = (i+m+h*f+a+n+k-
j*i+e)%100;
        l -= (a*j-n)%100;
        n += (b+h-k+b*l-m+b-
j+i)%100;
        f -= (d*n)%100;
        e -= (e-d)%100;
    }
    while( ++WHILEcnt[27]%5 )
    {
        e -= (n+d)%100;
        d += (b+j*i)%100;
        n = (f+a*l-n+e+g+n-b+k+n-
i)%100;
        a -= (k+a+e)%100;
    }
    do
    {
        e = (e+b*i-i+g-a-g-l+f-
b*m)%100;
        j += (k+l*d+a+m-d*k+j*h-i-
c)%100;
        j -= (f+l-a-i*h-d+i+e-k-e)%100;
        a += (b-h+g)%100;
        f += (c-f+n+m+m+l)%100;
        f -= (l+c+i-n-i-
d+d+b+n+a)%100;
    } while( ++DOcnt[28]%5 );
    }
    if( ++IFcnt[21]%10 )
    {
        g -= (k+n*c+h*n-m+l+h+f*e*b-
e+l)%100;
        for( ; ++FORcnt[43]%5 ; )
        {
            h += (b-a-l+m)%100;
            l -= (d+m+g*d-h+m+g-
j+j+h+d)%100;
            k += (h-l+e*j-j-c-n-a)%100;
        }

        switch( ++SWcnt[14]%4 )
        {

            case 1:
            {
                k = (m+g-b-b+m+j-j-e+m+m-
k+j)%100;
                i = (g+c)%100;
                a += (c+g+h+e+c-f+e)%100;
                a += (k-l+c+m-n*h+g-m-
h+m+k*h)%100;
                k -= (a*d-l-a+j+a-
b+e*f*f+j+g)%100;
            }

```

```

            break;

            case 2:
            {
                e += (b-c)%100;
                m += (f+h+f+i-c+i+d*m-l+a-a-j-
h)%100;
                k = (a-j-n-e*i+i)%100;
                d -= (b+c-a-a-i*i-n-a+a-n-
j*g+e)%100;
            }
            break;

            case 3:
            {
                h += (d+b)%100;
                k -= (l-m*n*g*l-c*a-c-c)%100;
                n -= (j-l+n*n-e-e*b+g*k)%100;
                b += (a-d*e-a*d-n+c+a+f*m-l-
c)%100;
                g -= (e-n-j-n*k+d+a+f+l)%100;
                l += (d-h-a-m+l-l-c+f-d-n-
h)%100;
            }
            break;

            default:
            {
                c += (g*a-n-m*a)%100;
                m = (n+l+e+k-
c+l+l*k+h+f)%100;
                l -= (g-c+h*j-l-k+b*j+d-e-
i)%100;
                b -= (a-g-k-j+l+a+h-i-
m+j+i)%100;
                k += (i-k+g)%100;
            }

            b += (b+f-a-c+d)%100;
            j -= (h+f+c+d)%100;
            m = (e+k)%100;
            h -= (f-m+d*k+b+a)%100;
            l += (l*f+e-k-j+k*d)%100;
        } while( ++DOcnt[27]%5 );
        j += (e+i+a+g+i*b-c-i+d+k+h+m-e-
h)%100;
        e += (k+m)%100;
    }
    e += (i*j-l+a-e-g+b*b*d*c*j)%100;
    f -= (c+a+c+i-n+n+m)%100;
    n += (l-g-e*f-n-n+k+n)%100;
    i += (d-f*k+c+a-i-e-i*a)%100;
    }
    g -= (f*h*a)%100;
    j += (n-c-i*b)%100;
    h -= (d+e+k+f+k+j)%100;
    f += (d+b+e-e-k*f)%100;
    j += (h+h+j-c*k+d-j+h+b-c-k)%100;
    i -= (h-n)%100;
    l -= (h*m+l+g*d)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F14(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFEcnt[31]%2 )
    {
        k += (i+k-k+j+e-g-l-a)%100;
        c -= (n-n)%100;
        k -= (l-f*c-e+n*b+f)%100;
        m -= (a-c-l+f*k+i-h*a+e-a+m-m-
m)%100;
        h += (g*j+j-d-n+b-f-c-a)%100;
        g += (k+n+a*c+n+d-l)%100;
    }
    else
    {
        while( ++WHILEcnt[30]%5 )
        {
            do
            {
                for( ; ++FORcnt[45]%5 ; )
                {
                    if( ++IFcnt[22]%10 )
                    {
                        d += (l-g+d-i-g)%100;
                        d -= (l+b+f+l-d+f*a*b-c*n-
f+i)%100;
                        k += (l+e+k-c*i*n-j-d-
b*n)%100;
                        b += (d-h)%100;
                        b += (h+l-e-c+b+a+i+a-i-
k*h*i+i+h)%100;
                    }
                    for( ; ++FORcnt[44]%5 ; )
                    {
                        i += (f*j+h-n+l+f+h-m+c-e-k-
b)%100;
                        j -= (n-h-c+l+i+b+i*g-
c+h)%100;
                        e -= (l*l-n-j+i+c-h-m)%100;
                    }
                    if( ++IFEcnt[29]%2 )
                    {
                        c -= (m-i-n)%100;
                        l -= (j-f+b-f-g*h+d-a)%100;
                        m -= (h+g)%100;
                        f -= (g-e)%100;
                    }
                    else
                    {
                        h += (e-d-n+a-j+j*e)%100;
                        k += (g-c-a-l*l-a-d+e-k-b*n-
n+j+h)%100;
                        h -= (j*j-f*k-b)%100;
                        c -= (j+h+c*b+f-c*m+j)%100;
                        i += (b+j+l)%100;
                    }
                    while( ++WHILEcnt[29]%5 )
                    {
                        n = (a-k*f+n+c-k-k+a-a)%100;
                        j -= (g+j+m-m-b)%100;
                        n -= (c*b-j-c+d-h*f-h)%100;
                        i +=
(b+b+j+n*n+l+b+e*i*m+k+d)%100;
                        c -= (n*a)%100;

```

```

n = (g+j-f+i-n+l+i+g+g+e-
c)%100;
}
do
{
j += (l-m)%100;
h -= (f-n-n*i+h-g-g-j+m)%100;
m += (f+e+c-h*n-b*c*d*e+g-
k+e)%100;
e -= (a-f-d+j)%100;
m += (e-l-h-j-d*a+f)%100;
} while( ++DOcnt[30]%5 );
i -= (i-i*n*e*a+n-k-h*m-f+d-
l)%100;
}
for( ; ++FORcnt[46]%5 ; )
{

switch( ++SWcnt[15]%3 )
{

case 1:
{
m -= (c+c)%100;
j -= (l*i+g-h-d+h*b+j+f+g+k-
g+k)%100;
e -= (l*n-j*b-c-l-b-h+m+h-h-
j*a-f)%100;
k = (a-i-a+j+a-d-d-l+h-c)%100;
}
break;

case 2:
{
g += (e*n-f+c-g+m*f-k*e-
b)%100;
c += (i*j-e-d-m-g*e+b+n-
g)%100;
l += (i+e-m+b*m-f-
n+h+n*k+l+m*j)%100;
a -= (i-e-g+n)%100;
k += (a+i-g-b-a+i+g-n+l)%100;
b -= (m*g)%100;
}
break;

default:
{
c -= (j-j*k-d*i+e)%100;
e += (m-m-b+e+l+e-n)%100;
f = (e+c+d-m+b-
m+m+e+i+n+a-h)%100;
a += (g+a*j-i-g+a-d)%100;
k -= (d-l-
c*g*k+i+i*f*g*j+j*a)%100;
}
}

if( ++IFcnt[23]%10 )
{
g -= (b+m+k+e+c-l+f)%100;
b -= (f-l+k+f-h)%100;
i = (m+j-j)%100;
}
if( ++IFcnt[30]%2 )
{

```

```

e -= (d*f+f-i*d-c+d+l+k-e+c-
l)%100;
l -= (n-j+d-l-
k*c+h+f+b+c+a)%100;
m += (b-m-m)%100;
n += (c+k-g+i+g-h-
m+b+k*n+i)%100;
}
else
{
m += (c-k*n-i*d-g+f)%100;
i -= (j*m-g-
m+j*j+j+n+n*n*h+i)%100;
k += (e+d+k-b-e*k+d*d*g-
a)%100;
i = (d+l-b-l+f*j+g+m+h*n-l-i-
a*h)%100;
n += (m-l-e*l)%100;
}
d -= (i-l+j-d*m+k+c)%100;
j -= (i+h+h-c-k+e+a+e+g*h+k+c-
h)%100;
}
k = (m-i-e+k-m-j)%100;
m -= (j-n-a-g+h+h+e+a)%100;
} while( ++DOcnt[29]%5 );
b -= (f+l*j-n-h+a*e)%100;
n -= (c-i-d+m+e-i)%100;
n -= (b+e+b*e+g*a+c+d-a-g*h-
m)%100;
i += (m+d-j-d-k+h*d+j-h-b-f-
m*k*h)%100;
}
a += (j+d*i-n+g-e)%100;
}
c += (m-f+h*j-c-h*m+b)%100;
c += (b-c*m-d+j-h+b-c-i-l+e+f)%100;
j -= (i*d*m-h-j-j-a-c+i*f-g+i)%100;
c -= (b-b-g*i+h+e*i+b+j-g)%100;
l -= (a*j-j*j-e*b+j+c+f*f-m+g)%100;
f += (h+e-d*l-h+h+b)%100;
h = (k+a+e-m-n*m+k-e*a+h*b)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F15(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
while( ++WHILEcnt[32]%5 )
{
do
{
for( ; ++FORcnt[49]%5 ; )
{
for( ; ++FORcnt[47]%5 ; )
{
if( ++IFcnt[32]%2 )
{
m -= (g+d-g+k+h-k+a)%100;
k -= (i+n+k-h-m)%100;
i += (a+h+h-i+b*j-l)%100;
d += (c+f)%100;
j = (i+n+j+n-g+n+k-e*k-
b)%100;

```

```

e += (e+g-i+h)%100;
}
else
{
e += (m*g+b+h-h-i*c-a*k+f-
h)%100;
g += (h-b+a-d*d-i+g+d+h+d+j-
l)%100;
e -= (g-c-j-l-a+l+f-d)%100;
g += (j+i+k*e-e*g-h+b)%100;
}
while( ++WHILEcnt[31]%5 )
{
j += (g-l-b+d*j*l)%100;
l -= (l-n-m)%100;
j -= (j+f+n+h+k-k+c-i*k)%100;
n -= (b*k-c-e+i+b-g-d)%100;
b += (j*j)%100;
}
do
{
i += (n+e-j-g-b)%100;
g -= (g-c+k+m+k-j+a+a-g-
l)%100;
m -= (m+c-n+b-a-d-i+b+h*a-
i)%100;
} while( ++DOcnt[32]%5 );
if( ++IFcnt[24]%10 )
{
k += (e+e+l+i+d+l-k-
k+c+d)%100;
a += (d*m+c+h+k+i*f-c*g-
g+l*b+m*a)%100;
a = (a-l-k+m+c*l)%100;
d = (l-m-j+b-j+e*h+i-
m*c)%100;
l += (k*l-c)%100;
}
c += (l-d*m+f*a-b-a-d)%100;
}
for( ; ++FORcnt[48]%5 ; )
{

switch( ++SWcnt[16]%4 )
{

case 1:
{
h -= (m+j-f+b*k-n+l+l-m-a-
b*c)%100;
g += (j-f+g+n*k-d+a-n-f)%100;
i -= (m-e*a-g+m*k+c-m+a-
n+n-j)%100;
f -= (m+b-f-j-m+m+a*e-a+n-g-
a)%100;
f = (j-l+d-k)%100;
j -= (m+e+k-b+h-k)%100;
}
break;

case 2:
{
e -= (d-g-d-i-k-j+m*n)%100;
n += (a+e-a+d-
d+k+a+m*a*l+m+k+d)%100;

```

```

f += (e-f*b-f*m+h-
e+c+g*n+n+f)%100;
b = (e*m+c-g-a-g-i)%100;
n += (e*g+c-f)%100;
}
break;

case 3:
{
b -= (j*f*a)%100;
b += (f+n+n+b*d+l)%100;
j = (g-l+h+g+c+i*h+a-d)%100;
d -= (c+h*h*d-m*e-c*i+d-
n)%100;
d -= (g*c+i+h+l-b+b-a-n+i-
m)%100;
}
break;

default:
{
e += (c*f+g*f+f+a+i*b+i*n-
g*k)%100;
b -= (a+f+i+l+e*i-b+j+m-j+e-
c*d+i)%100;
l = (g*g+c+f+d-e+b+k-e*h+k-
c)%100;
l = (a-m+h-e-c)%100;
}
}

if( ++HFEcnt[33]%2 )
{
h -= (k-i-c+j+g+f)%100;
e += (e-m+m+j+e+h+h-
g+i+f*n+h+b)%100;
h -= (c-d-l-d)%100;
c = (k+d-j+i*e)%100;
a -= (n+i-n-c+m-j+m+g+k+k-
g+d-h)%100;
}
else
{
h += (n-h+g-f)%100;
l -= (l-l*a+l+m*g-i+h+i-e-
j*m)%100;
c += (d-c-c-l*e-b-i-e+k*m-i-e-
b)%100;
f -= (j-i-a*j)%100;
a += (e-h-i*e+a+a-k-d+a+i*d-
b)%100;
a += (d-h-e-l-h+b-c-k-
j*g)%100;
}
n -= (n-k+l-g)%100;
c += (g-a*f)%100;
}
} while( ++DOcnt[31]%5 );
g += (e*l-n*k-j-h+g-e)%100;
j -= (f-b+a*j+e-c-g*g+e-c-h)%100;
n -= (c-n*l+e-i+d)%100;
m += (l+i+l-f+i*j+f-m*c)%100;
}
i -= (a-a-l+n+b)%100;
m += (l*l-k*d+b-l-e-g-h-e+j)%100;

```

```

d += (l*e-n*b+g*n-k+b+c)%100;
m -= (g-g-b*c)%100;
j -= (d+f-b+f-h+c+e+c-e-l*m+i)%100;
g -= (b*g*g*a-j+m-c+c-g)%100;
e += (a*h*i+g*b*i)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F16(void)
{
int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
while( ++WHILEcnt[35]%5 )
{
do
{
for( ; ++FORcnt[54]%5 ; )
{
if( ++IFcnt[25]%10 )
{
for( ; ++FORcnt[50]%5 ; )
{
d = (n-i*j*d+c+g+m-l-i-g-f+i-
i+k)%100;
h = (i+k+k-b-a-a+k*j)%100;
h += (e+c+c-k+c)%100;
a -= (l-c-i-k*n-l*g+j+b-h*j-k-b-
n)%100;
g = (n-h*j-c+a-
g*k*f+l+f)%100;
}
if( ++HFEcnt[34]%2 )
{
j = (n+f-i+k+n+d+m+d*n+c-
f)%100;
j -= (k*f-d-h-h+i*b-b-b)%100;
}
else
{
b += (a+h+g*e-e+g*j+n-a*h-n-
g+i)%100;
}
while( ++WHILEcnt[33]%5 )
{
d += (b+m+l-a-j)%100;
n -= (e*m+b-h*d-m+f*i+m-
f)%100;
m += (n*m+b-j-h-m+b+i-
l*i)%100;
i += (n-d+b)%100;
b += (j-n-a*e+a*i)%100;
}
do
{
f = (f+l+g)%100;
e -= (g+m+b-l*c*m+j-b-
c+e*d+m)%100;
m += (i+i)%100;
a -= (j+e)%100;
} while( ++DOcnt[34]%5 );
i -= (j-m*d-k-k*l)%100;
for( ; ++FORcnt[51]%5 ; )
{
h += (i+h)%100;
a += (n+b+d*k+j+f+i+j)%100;

```

```

d = (a*h-b-j-a+l)%100;
d += (i-c*g-a+h-n+l*n)%100;
a -= (b+a+i+l+h*a-e+n-k+j+l-
j)%100;
l -= (m-b+h-e+l+a*h+g)%100;
}
}

switch( ++SWcnt[17]%3 )
{
case 1:
{
if( ++IFcnt[26]%10 )
{
d -= (j*e+a+d*l-j-h)%100;
l -= (a*l)%100;
g += (h+h)%100;
}
if( ++HFEcnt[35]%2 )
{
j += (n+e-d-g+j)%100;
n -= (n+n*n-l-i*f+m-a)%100;
m -= (h*a+m-m-f-e*c-a)%100;
f += (d+h-d-f+h+h*h)%100;
}
else
{
g -= (m-d-j-a-c-b+h-l-a+h-m-
e+k+c)%100;
g += (d-d+b-i+h-a-j-h*i*l*e-
k-e)%100;
f += (e+i+c)%100;
m = (m-c+j+m)%100;
k += (g-n+g+f+a+n-a+i+b-
f+c+m+n*h)%100;
}
while( ++WHILEcnt[34]%5 )
{
a -= (a*b-j-i+e*j+i)%100;
l -= (k*h-h*k+c-k-d*g-l*b-d-
h+n-n-c)%100;
f += (l-m)%100;
f -= (h-k*d+b-c+a+m)%100;
c -= (i-f-f-h+m-f-a-
m*a*f+m)%100;
i -= (l*d-h-m-f*g)%100;
}
do
{
j -= (n+g)%100;
c -= (g*k-k*f+c-e*f-c)%100;
h -= (m+l+h+b*l-e+j-
j+j+f)%100;
i += (g+h+g*k+g+b+h+k+b-
a+c+g+e)%100;
k -= (n+a+c+d-d-l)%100;
} while( ++DOcnt[35]%5 );
for( ; ++FORcnt[52]%5 ; )
{
k -= (c+e)%100;
m -= (n+h-e)%100;
b = (i*b)%100;
n -= (k*n*i*g-b-l-m+b-f+i*j-
l+k+a)%100;
b -= (i-m*a+i)%100;

```

```

    }
}
break;

case 2:
{
    for( ; ++FORcnt[53]%5 ; )
    {
        f -= (l+e+i-i-g-d-f+i+l)%100;
        l -= (m-g-m+d-j)%100;
        k -= (m+d*c*b)%100;
        n -= (l*d-f*n)%100;
        c = (m+a*e-n-n+i-k-k-f-
f+m*h)%100;
        a += (m+h-d+a*j+j+d*a)%100;
    }
    k -= (f*k)%100;
    d = (g-f+i-l*n-i+h-g+i+m)%100;
    m -= (n+i-j*j-a-m-b+h+d)%100;
}
break;

default:
{
    f -= (g*i+d+g-j+g-g)%100;
    e -= (n-d+l+g-k+l-j-m+b-k-k-
k+n)%100;
    a -= (e-d-d+h)%100;
    h -= (m-l-k+n*c+b*j-a*c-
e*g)%100;
    g -= (k*h*a+b+h-j+d+k-m-
c*l)%100;
}

    e -= (e+b-h)%100;
    l -= (b-d)%100;
}
k += (l-f-i*f-m*a+m-g-j)%100;
m += (n+b+f+a-f+c*a+f-c-
l+i+h)%100;
a += (f+f)%100;
m -= (j-a+e+n-a*e-g+e-a+c)%100;
} while( ++DOcnt[33]%5 );
h += (f+d+a)%100;
a += (m*a-k+g-h+c-i+c-n*b*l)%100;
}
b += (b+d+a*l+g*j+d+l)%100;
i += (g*e+e*a+h)%100;
m -= (f-i*a+k+h*m+l+b+c+h+e+j)%100;
h -= (j*m*h-h-k+b-e*c-f+k)%100;
e -= (m+a-f+f-j-d-n-g-c)%100;
j += (g+k*f-k-m)%100;
b = (e-e-e)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F17(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFEcnt[38]%2 )
    {
        c += (d-k)%100;
        c -= (j-l*m-n*e-i*c+l+k)%100;

```

```

        l -= (g-n+k+b-f-e-c-h+b+g*l)%100;
        g -= (a+e)%100;
    }
    else
    {
        while( ++WHILEcnt[37]%5 )
        {
            do
            {
                if( ++IFCnt[27]%10 )
                {
                    e += (n*f*e+d*n+j-d-g+i)%100;
                    for( ; ++FORcnt[55]%5 ; )
                    {
                        c -= (i-k)%100;
                        k -= (f*c+k-b+n*a-m)%100;
                        n = (m-j-d+a*m*k-g)%100;
                        g += (a-e-d+l-m-i*a+l*k-
a+g+g)%100;
                        g += (g-f*g+c+d*i*g+a*j-k+j-
c+f-n)%100;
                    }

                    switch( ++SWcnt[18]%3 )
                    {

                        case 1:
                        {
                            n -= (i*h)%100;
                        }
                        break;

                        case 2:
                        {
                            l -= (m*d+m-i+d-c-l+d-
k+n)%100;
                            n += (a*i+g+a-g+b+b-
k*f+c+e)%100;
                        }
                        break;

                        default:
                        {
                            k -= (e-f+d-l*n*h-h+m+c-n-j-
k*a)%100;
                            d += (g+g+e+a+i*i-e-d)%100;
                            c -= (m-n-k-h-h*b-e+n)%100;
                            n -= (k+j)%100;
                            n += (d+g-e*n-k*n)%100;
                        }
                    }

                    if( ++IFEcnt[36]%2 )
                    {
                        f += (h-c)%100;
                        g = (f-i*a-a*i+l)%100;
                        e += (i+l+h+k*i+j-n+e-m-
e*e+a)%100;
                        f += (d+k-m-f*n*g)%100;
                        m += (a*h+j-e+j-i-a+h-i-g-
c)%100;
                        j -= (k-a+e-c+a*h*e*b-c+h-
e*m+n+d)%100;
                    }
                    else
                    {

```

```

                        c += (l+g)%100;
                        h -= (j+f*k*h)%100;
                        m = (k-k+n+g+e+m+b+d+e-d-
i)%100;
                        m += (k*i-a-k+g)%100;
                    }
                } while( ++WHILEcnt[36]%5 )
            {
                m += (n*c)%100;
                f -= (l+j*h-a-c+d-a)%100;
                d -= (g+f+e+f+g-h+b-n+l*i-
e)%100;
                e -= (a-a)%100;
                c -= (h+k+m)%100;
            }
            do
            {
                g += (n*e)%100;
                d -= (i-g+l+k-i-e+j+m-a+n-i-f-
m*n)%100;
                d += (k+l*g*g-j+h-c*i*i*g-g-j-
g)%100;
            } while( ++DOcnt[37]%5 );
        }
        for( ; ++FORcnt[57]%5 ; )
        {
            if( ++IFCnt[28]%10 )
            {
                e += (e+a+m-k+h+g+e*k+c-
j+f)%100;
                f = (h+i+e-k)%100;
                f += (n-h*l)%100;
                h += (i*n-i)%100;
            }
            for( ; ++FORcnt[56]%5 ; )
            {
                g -= (f-f+i+e-e-j-g+j+h+h)%100;
                f = (c*b*e-
n+h+n+k+g+i+a)%100;
                i += (a+n)%100;
                i += (e+e-l*a+m+i+h-e-
d)%100;
                j -= (c+b+f+d-i)%100;
                e += (b-m-m*b)%100;
            }
            if( ++IFEcnt[37]%2 )
            {
                d -= (m+d-d*b*d-m*a-a*b+d-
c-g*j)%100;
                d += (j-b)%100;
                j -= (f+c+h+m-k+b-n)%100;
                f += (e*h-n-i+g)%100;
                j += (a+m+j+m+e*h-g+l-h*l-b-
n+f)%100;
            }
            else
            {
                f += (h-a-h+c+j-k)%100;
                a += (j-a-g+f*d+f+n-
h+m)%100;
                h += (k*a+i+e-k)%100;
                c += (g-n+f-m*a)%100;
                j = (a-d-h+h-b+j+j+j+f+m+n-n-
f-g)%100;
            }
            k += (g+e+b+d-d)%100;

```

```

        b += (m-k)%100;
    }
    l = (e-c-j-j+f+n+n-j+l*m)%100;
    n += (n+c*i-i+n*g)%100;
} while( ++DOcnt[36]%5 );
j -= (g+h+g-g-k+e+i-e-k+f*g)%100;
a -= (n+f-k-j)%100;
m -= (d-e+a-b*j-m+k+k+c+k)%100;
i -= (k-a+j+l)%100;
}
m -= (b*h*n-l-l+f)%100;
i += (i-g-j+n-m+j-f+c-n-k)%100;
}
j -= (l+g+g-i-a*c-a+d+i)%100;
h -= (c-f-f)%100;
c -= (d-a-i-a-k-m-n+f)%100;
c += (b*m*e+g-b)%100;
b += (l-b)%100;
l = (b-f-j+g-j*m*l+a-b+g)%100;
k += (f-f*i+h+j-e+l+n-h+n+l)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F18(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[40]%5 )
    {
        do
        {
            i -= (j*d-d+k+n)%100;
            for( ; ++FORcnt[60]%5 ; )
            {

                switch( ++SWcnt[19]%3 )
                {

                    case 1:
                    {
                        if( ++IFcnt[29]%10 )
                        {
                            f += (a*b*f+n)%100;
                            b -= (b+c*h*i)%100;
                            c -= (e+l-i+c*k-h+i)%100;
                            f += (k+k+h-a+l+a-e-
h+k+j*b+l*m+m)%100;
                        }
                        if( ++IFcnt[39]%2 )
                        {
                            c += (i+d)%100;
                            g = (e-f+j+d-d-
i+b*j*b+b+f+l+a+d)%100;
                            a -= (b-i-f-h)%100;
                            k -= (j-i-c-f+m+e+e-j)%100;
                            c -= (a+f+g)%100;
                        }
                        else
                        {
                            e -= (f-j*n-l-d+j)%100;
                            h += (d*a+a*l-c-
b+j*m+a+c*h+e+f-n)%100;
                            k += (l-l-e+l-k)%100;
                            h += (i+g+i-l-j*n-l-b)%100;

```

```

n)%100;
        g += (m+c-e+n-h-e+n-i-i-l+c-i-
n)%100;
        a = (g+b-h-d-k-j*e+l-
i+n+e*i*m)%100;
    }
    while( ++WHILEcnt[38]%5 )
    {
        h -= (a*k*)%100;
    }
    do
    {
        d += (c*j-i+n-c*f-d-c*e-
d+f+a)%100;
        f += (b+m*f+h*f+g)%100;
    } while( ++DOcnt[39]%5 );
    for( ; ++FORcnt[58]%5 ; )
    {
        n += (f-i+b+c-b+l-d+i*n-k-
i)%100;
        b -= (a*n*l+i*c+n)%100;
        d += (d+b*d+c+f-g-
c*b+j+l)%100;
        a -= (i-k)%100;
        j -= (j*m)%100;
    }
    break;

    case 2:
    {
        for( ; ++FORcnt[59]%5 ; )
        {
            h -= (f-d-d-m-b-c-n+a-c)%100;
            h += (f-j-j)%100;
            l += (j-c*h*h+i*b+j-d-
m*f)%100;
            e -= (g+h*l-j-f-m-n-f-l+m*d+g-
k+a)%100;
            e += (f-h)%100;
            d = (a-n-b*a+n+k+h-l-
n*c+e*d+h-i)%100;
        }
        if( ++IFcnt[40]%2 )
        {
            n += (g+d+d+e-e+j)%100;
            l -= (j-h)%100;
            c += (a*m*n-i-k*k+e-a)%100;
        }
        else
        {
            m += (n-m)%100;
            c -= (e-c-m-f)%100;
            c -= (m+l+d+i-h-n+m*d*m-
j+i)%100;
            d += (m-j*f-l+n-b+b+m)%100;
            g -= (n-l)%100;
        }
        while( ++WHILEcnt[39]%5 )
        {
            h += (l-g+n*l+j+k+f)%100;
            g += (h-k)%100;
            a += (h-g*f-d-n+a)%100;
            e += (e-k*l-d+e-h-c)%100;
            f += (e-f-a-l*n-k+h*c+n)%100;
        }
    }
    do

```

```

    {
        d += (h-e+l+h)%100;
        a = (l+a-j*i-l-e+c-j-
i*j*i+g)%100;
        b += (g+b-g-k)%100;
        c = (c+l+g*j-b-a)%100;
    } while( ++DOcnt[40]%5 );
    }
    break;

    default:
    {
        h -= (n+k)%100;
        c += (a*h+c+a)%100;
        h += (j+e+c+d+h+i-m+c-f*j*j-
e+n*i)%100;
        e = (a+g*f+b)%100;
        e += (h*m+d-g+b-f*l*c+j-k-
c)%100;
        a += (f+b)%100;
    }
    }

    n -= (g-f*n*b+d-k-n-n)%100;
    n += (b-h-g-e*g)%100;
    }
    k += (j+e+m+f*c)%100;
    j -= (g+l)%100;
    i -= (i-i*c-m+c+l-n+g-n+f+i-i-
d)%100;
    } while( ++DOcnt[38]%5 );
    m -= (b+l-a)%100;
    b += (g+g)%100;
    h -= (m+f+m-c+l+e-c)%100;
    c -= (i*l-k)%100;
    j -= (k*l+b)%100;
    }
    e += (g*j+d-c-i+h-j+g)%100;
    l += (g-n+b-d-j*d*f-c-f*f*h+a+f)%100;
    g += (a+f)%100;
    g -= (e*i)%100;
    j += (g+l+h*m)%100;
    e += (e-a-l*j)%100;
    d -= (c*j+m+h)%100;
    return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

```

```

int F19(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[31]%10 )
    {
        k += (n*j*j*d+a-l+h-e+e*e-
g*l+f+i)%100;
        for( ; ++FORcnt[64]%5 ; )
        {

            switch( ++SWcnt[20]%4 )
            {

                case 1:
                {
                    if( ++IFcnt[42]%2 )
                    {

```

```

while( ++WHILEcnt[42]%5 )
{
    m = (f+h+i+k+j*b+n-g+b+j-
i)%100;
    g += (c-b+m+n+c-e+c-b-
c)%100;
    l -= (k-i+e)%100;
    k += (g+l+f+n+e-
c+k*d+e)%100;
    f -= (l+d*n+f+c+n+k*l-g-c-a-n-
h-e)%100;
}
do
{
    i -= (i-h-a+m+d-d)%100;
    f -= (k*c-i-e+m-j*n-c-
d+l*e)%100;
    f += (l*n-d*c+d+b*k-
a+b*d+k+k-j)%100;
    k -= (i*g*m+n-c*f-l-b+n+e+l-
k)%100;
} while( ++DOcnt[42]%5 );
}
else
{
    while( ++WHILEcnt[41]%5 )
    {
        l -= (i+d+e+l-b-g-
c*g+k+k+c+g*m-i)%100;
        n = (i-i-l-c-e)%100;
        c += (m-h-c*g-g-c)%100;
        c -= (m+c)%100;
        i += (m-d)%100;
    }
    do
    {
        f -= (g+e+a-a+j-j+f-e)%100;
        n -= (i-h+c+k+n+j*c)%100;
        m += (h+m+b-b*n+a+m-
i*g*h+l-g*h)%100;
    } while( ++DOcnt[41]%5 );
    for( ; ++FORcnt[61]%5 ; )
    {
        g += (a+l+h+a+l+g+f-m+j-
e*l)%100;
        k -= (m*i)%100;
        g += (b-c-m-e+k*l+c)%100;
        d -= (f+n-b+h+g)%100;
        f += (d-m)%100;
    }
    if( ++IFcnt[30]%10 )
    {
        a = (b-i-b+b+b-l*i+h-c-
c+e)%100;
        k -= (b-g-h)%100;
        l -= (e*l+j+g+c)%100;
        i += (m*n*h+k-n*c)%100;
    }
    for( ; ++FORcnt[62]%5 ; )
    {
        b = (l+g+g+i-k+g+i-e)%100;
        k -= (l*e-h+j+c-l*a)%100;
        n += (m+h*e-c)%100;
        n += (f+k+d+c-c*j*n)%100;
        e -= (l+d*g+b+f*n+h)%100;
        h -= (d+d+h+e-l)%100;

```

```

}
if( ++IFEcnt[41]%2 )
{
    j -= (b-n+h+i-a-n+f+i+e-
e)%100;
}
else
{
    i += (j-m*f+e)%100;
    m = (f+k-h-i-h)%100;
    c -= (i+n*k*i+e-d+k+j-b)%100;
    f -= (c-n-m-g-b-m+f+n+h-
m+l*d+n)%100;
    b += (g-k*f*l*c-g-g+l-a-h-
a)%100;
}
i += (g-m-l*n-m-d-b+e)%100;
for( ; ++FORcnt[63]%5 ; )
{
    switch( ++SWcnt[21]%3 )
    {
        case 1:
        {
            h -= (a+n*g-i*e+c-f*g)%100;
            k = (b+d+m-b+h-h-k-j+f+i-
d*j)%100;
            n += (l+k+a+c)%100;
            n -= (k-i*b*b*m*e+i-n)%100;
            a -= (g+f-g-i+n*d-g*l-f-g)%100;
        }
        break;
        case 2:
        {
            b += (k+d-a+e+l+k)%100;
            d -= (b-c-g+h+d)%100;
            i += (g*a+j+d)%100;
        }
        break;
        default:
        {
            k -= (c+m+i*b-j-l)%100;
            i = (e-m+e+h*f-f-l+d)%100;
            k += (h+c-h*j-d-
g*i*n*g+l)%100;
            g += (d*h*l-k+d+c*h-h*k-k-i-
d)%100;
            a -= (e+b+m+f+b+g+j-k)%100;
        }
        e -= (g+g)%100;
        l += (h-k*i)%100;
        i += (b*a-n+b+l-n+f-e*f-
m)%100;
        j -= (m+c)%100;
        a = (j-c-l-i*k)%100;
    }
    a += (m-d+e+g-g*m+k*b-
i*j+j+f)%100;
}
break;

```

```

case 2:
{
    l += (a+f-d-b*l+c+n*l+c*b)%100;
    f -= (m-m+l-d-d+l)%100;
    e += (n-k+e*f*l-c-d+a+m)%100;
    f += (a*d*e+f+a-g*j+i-m)%100;
}
break;

default:
{
    m -= (n+b-i+j+j+m*d-g+k*g-
h)%100;
    a -= (g+k-j+f+c-
n*m+n+g*k+k)%100;
    e += (c-c+n-k+l-d+l-m-c*n-
k*i)%100;
    l -= (f*n-l+l+a-a-j-l*n)%100;
    c += (b-m+n+k-k-f+m)%100;
    m -= (l-a+a+m+c-l*i+e+d*j-
j+a)%100;
}
}

k += (h+i*l+a+d-m+f+h-e)%100;
c -= (n+i+i*l)%100;
h += (n+d-c)%100;
b += (b-b+l*d+b-d+m+c+c-i-b*k*n-
i)%100;
}
h += (f-m+a*l+e-d+b-d+g+c+h)%100;
n += (g+i-c*l-l+i-
j*k+e+g+d*g*b)%100;
a -= (m*c)%100;
}
g -= (c*e*m+e+k-a-k)%100;
b += (n-d+f+k*k+b-a-k-e-e)%100;
c -= (h+n*g*m+c+f+g+h+j+g-
a+a*c)%100;
i -= (c+f-b-c-k+h*a+n-m+i-a)%100;
a += (l*b+n+c*c)%100;
d += (d+n+i*b+g-c-c)%100;
n -= (l+k-d-d+m+d-h*i*b+c-
a+n*d)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F20(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[33]%10 )
    {
        if( ++IFEcnt[44]%2 )
        {
            f += (n-b)%100;
            g += (g-f+j+g-k-e-b+a*e)%100;
            g -= (i*c-g*k*k+m-a+d-g*k)%100;
        }
        else
        {
            while( ++WHILEcnt[44]%5 )
            {
                do

```





```

}
while( ++WHILEcnt[45]%5 )
{
    n += (j+l*k*n)%100;
    j -= (c+a+a*l)%100;
    n += (m-j-m-n-f+g*i-n-f)%100;
}
do
{
    l += (c*i*g+f*n)%100;
    m -= (k*k-m-n+l)%100;
    a -= (b+c-i+i*k*c+k*b+i-
d*m)%100;
    i = (g-e*j-l-l-a+n+d+i)%100;
    b += (k-d-i)%100;
} while( ++DOcnt[46]%5 );
}
d += (l-b)%100;
for( ; ++FORcnt[70]%5 ; )
{
    switch( ++SWcnt[23]%3 )
    {
        case 1:
        {
            c -= (c*k*k-f-j+k+l*j)%100;
            d = (e-l-m-c-a*m*m+f+i-
a)%100;
            a -= (f-d-l+j+a*h+m)%100;
            k -= (d-h+a-f-j-k-k+j+k+d-
c)%100;
            h -= (b*e-m-a-j-d)%100;
            m -= (b+e-n-j+l-g-a-l*j+i)%100;
        }
        break;

        case 2:
        {
            g -= (n*g+l+h)%100;
            g -= (a+k-h+d-k*k*c)%100;
            a -= (f*b-c-n+n-c*n-k*b-c*k-
j)%100;
            f -= (i*d+c-d)%100;
            k = (h-i-k)%100;
        }
        break;

        default:
        {
            n -= (a+l+a+d*n+l-l*a-
m)%100;
        }
    }

    if( ++IFcnt[35]%10 )
    {
        i += (k*d+i)%100;
        i -= (n-l*c+a)%100;
    }
    if( ++IFcnt[46]%2 )
    {
        m -= (a+l-
m+k+c*g*c+c+h)%100;
        f += (m-c-i+g+l*a+l*m-
e+l*h)%100;

```

```

    a += (g-n+c-c)%100;
    c -= (k-b-a-i*k+n-l-d*l+m+l*g-
b+b)%100;
    }
    else
    {
        a -= (f+f)%100;
        j += (i-c+c+c-g+e+a+g)%100;
        e -= (l+g-l+a+i+g+c-l-
k+l+b)%100;
        d -= (b+i-h+n)%100;
        f += (j+e-n*c+k-i+n-b+j-k-l-
g)%100;
    }
    g += (n-k)%100;
}
f += (g+b-e+b*f+h*j*a-m+f+a-
j+f*a)%100;
b -= (a*b-f-e+g*j-n*i+g-n)%100;
j -= (h-d+j+b-c-f*f*g-f+h-a)%100;
} while( ++DOcnt[45]%5 );
f -= (c-e-b)%100;
j -= (h+i-h)%100;
g = (i+m-c+k*m-b-c+b*c)%100;
}
i += (g*l+f+l+c)%100;
m += (f+k+i+a+e-g-b-j*e-g+c)%100;
d -= (b+h*c+d-l-m-j-
d+m+c+c+d+g)%100;
l += (l-m*l*f-c)%100;
}
h -= (i-m)%100;
f = (i-l*h+m+b-c+j-j-m+k+g-l-j-i)%100;
e += (c*g*k-j+g+l+a-m+k-i)%100;
b -= (g+d+m+g)%100;
m -= (d+j+i+h+h*n-f)%100;
n = (c+a)%100;
k += (b*i-d+i-m-j-l-g)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F22(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    while( ++WHILEcnt[48]%5 )
    {
        do
        {
            for( ; ++FORcnt[73]%5 ; )
            {
                for( ; ++FORcnt[71]%5 ; )
                {
                    if( ++IFcnt[48]%2 )
                    {
                        j -= (b+n+c-
d+k+e*g+n*j*m)%100;
                        g += (g-l+a-a+e-f*c+b-
j+a)%100;
                        d -= (i+m+i-g-
d+a*a+g*c+k)%100;
                        i -= (d+j)%100;
                        j += (e-m-n+e-e*a+g)%100;
                    }
                    else

```

```

{
    f -= (c*l+i+a+i-a)%100;
    k -= (h+i+k+f+c-m+b*b-
l+d+a)%100;
    i += (a+j)%100;
    j += (n-m-b-j*l-g)%100;
    n += (m-f*n+j+j+h*k-e+e-h+g-
l+i+d)%100;
    n += (g*f+h-m*f-f-a-c-f-
i)%100;
}
} while( ++WHILEcnt[47]%5 )
{
    d += (b-j*c*m+h+l+i-
i*c+c)%100;
    k += (c+l-n)%100;
    n -= (a+i+a-g+b+j-a+k+d-
g)%100;
    e = (c-g-m*e-m-j)%100;
    g += (e+n-j+l-k-g*g+n*a+f*i-
d+e)%100;
}
do
{
    d += (d-a-g*f+i*n)%100;
    f += (j+b*m-f-a-k-d*j-h+e*i-
g)%100;
    n += (b+b-c+g-g*b*h-d*k-
j*b*c+h)%100;
    h += (f+m+m-h-h-b-n+j*n-l-
n)%100;
} while( ++DOcnt[48]%5 );
if( ++IFcnt[36]%10 )
{
    n += (c-i)%100;
    h -= (i+h-f-h-e*f-f+i+f-a)%100;
    c -= (d-k*g-k*e*d)%100;
    a += (g+c*g)%100;
    n -= (g+d+i*e-a-j*h*m)%100;
    e -= (l+l)%100;
}
}
k = (l-d*k-n*c)%100;
for( ; ++FORcnt[72]%5 ; )
{
    switch( ++SWcnt[24]%4 )
    {
        case 1:
        {
            d += (l-l-h+a+n+f+a-b+l*f-
c)%100;
            e += (n-e*k)%100;
            j -= (k-n+l+c*e+a*m)%100;
        }
        break;

        case 2:
        {
            b -= (j+l)%100;
            l -= (k-e+k*g+i+b-h+d)%100;
            g += (c+a+e*d-f+j+i*d)%100;
            g = (g*k-c+e-g-k-k-l+d+e-
i)%100;
            i -= (d-g)%100;

```

```

    }
    break;

    case 3:
    {
        g += (b*f+e+f+f-c-i*e+n-c+h-
j)%100;
        b += (a-d+g-n*m+f-m)%100;
        c += (a+j+f+b+c-j)%100;
        c += (f+n-m-l+m+d*h+b-
e*i)%100;
    }
    break;

    default:
    {
        j = (m*n+g-n+e+l)%100;
        m -= (i*c*l+m*l-j+l+b-
j+j)%100;
        i += (h+f*c-d+h-g-k*i)%100;
        d -= (g+g-e*d)%100;
        n -= (f-i*n-f)%100;
        b += (h-e-i+h-n+e*c-k)%100;
    }
}

    b += (g+n+b-k-i+m+a-d-
f+a+n)%100;
    l += (i+b+l-i)%100;
    k += (j+j-c*j-k-b-d+h-
b+l+i+f)%100;
    i += (e+n*l+b+k+n-f-j+d)%100;
}
a -= (b-g*e)%100;
d -= (e*l-j-e+k*n-l-l*k-m)%100;
}
n += (g-m*h+f*c+a+f-l-d-c+b)%100;
e = (d+l-d-l)%100;
} while( ++DOcnt[47]%5 );
j += (b-i+m-d-b-i-j*g+k-i*e+h)%100;
j += (l*e+l+e-m+j+b+b*m+c)%100;
k += (j-n+d)%100;
j -= (a*a)%100;
}
d += (m-h+e-f+b-c)%100;
i -= (f*l-h+a+k-i)%100;
b += (l*b-j-f-l-f-b+b-n+h)%100;
f += (d-m*b*d*d+l-l-c+b+f+e+l)%100;
a = (k*e*h-k)%100;
m -= (m+i+m*n+c*a-i+d+e-j-
k+m+n)%100;
d -= (h+m+h+j+j)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int F23(void)
{
    int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
    a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
    if( ++IFcnt[51]%2 )
    {
        f = (g+d+a*h-a+m+j+g*d-d+g-l-m-
a)%100;
        c += (i*f-e+c-j*j*k+a)%100;
    }
}

```

```

else
{
    while( ++WHILEcnt[50]%5 )
    {
        do
        {
            for( ; ++FORcnt[75]%5 ; )
            {
                if( ++IFcnt[37]%10 )
                {
                    l += (e+j)%100;
                    l += (e-l+n+l+h+e-b*i*a-
k+j)%100;
                    i += (i-a-f*n-b-l+f+e-
g*e+n+a*l+e)%100;
                    n = (h+m-l+e-k)%100;
                }
                for( ; ++FORcnt[74]%5 ; )
                {
                    a = (f-g-e+j+c+l-a*d*b)%100;
                    e -= (n+d-g*e*l-m-
f+a*g+j)%100;
                    f -= (f+e+b-f-f-k*d*m*d)%100;
                    i = (d-i+k+f-k+c+j*l-m*b+a-c-
h*m)%100;
                    i += (j+e+m-a-m*h*m+l+f*i+i-
a+c)%100;
                    j = (e+i*g+f*c*a+b)%100;
                }
                if( ++IFcnt[49]%2 )
                {
                    k -= (d-f)%100;
                    a = (j*e-h*e+m)%100;
                    h += (c-h-l-c*f-
m*m+h+d+n)%100;
                }
            }
            else
            {
                k -= (m-g+g-c)%100;
                h += (h-a-g*k+k-i-g+a-g-l-
d*k+m+n)%100;
                c += (h+c+g-e-j-c*c-e+g+k+l-
m-l)%100;
                l -= (f-j+n-h-f*f+f+n*d+h+h-e-
l+m)%100;
                e += (e+m+b)%100;
            }
        } while( ++WHILEcnt[49]%5 )
        {
            f -= (f-n-d-m*g-n*d)%100;
            g -= (d-g+n-f-h*e+f+d-j)%100;
            n += (e*h-k-d+l+j-l+a*h*h-f-c-
i)%100;
            i += (g+e*m+g-m-l+n+b+k-b-
g)%100;
            m += (l-b*c-e-l+h-m-c-
c+k+k+d)%100;
        }
        do
        {
            m += (n*k-a*n+e+j+l-n-
i)%100;
            h -= (g-c+l+n*m*d-
c+i+g*n+b+f)%100;
            b += (l*d+c*b+m+n)%100;

```

```

        d -= (e+k-m+g+f+b-g*b-
c*n*n)%100;
        } while( ++DOcnt[50]%5 );
    }
    h -= (b+g+c+c-d-h+n-m-
i+f+k*i+e)%100;
    } while( ++DOcnt[49]%5 );
}
for( ; ++FORcnt[78]%5 ; )
{
    switch( ++SWcnt[25]%4 )
    {
        case 1:
        {
            if( ++IFcnt[38]%10 )
            {
                if( ++IFcnt[50]%2 )
                {
                    l += (l+k-n+e-g-a+j+l-k-j)%100;
                    e = (j-m+e-l*e-j-i-l+e*l-
l)%100;
                    f += (k-d)%100;
                    e += (j-m+j+k*a+k-
n*f+a*m+b-d)%100;
                    a += (k-a*g*d-l+j+k)%100;
                    i -= (n-d-j-k*m*c+j)%100;
                }
            }
            else
            {
                h += (g*e+i+h+l)%100;
                g += (i-j*g+l-a-j)%100;
                n += (i*g-d-i-i)%100;
                b -= (n-k*b-j+g+i)%100;
            }
        } while( ++WHILEcnt[51]%5 )
        {
            b -= (g+k+k-m*a*h-
k+b+h)%100;
            b += (i-b-a-l+c)%100;
            h += (f-g-f+l-k-m-l+j-f-c-e-
m+a)%100;
            e += (f*k)%100;
            e += (c+e*n+h*b+h+a-f-
a)%100;
        }
        do
        {
            g = (h+i+n*j*l-c-m-d-i)%100;
            k += (f+g+i*n+i+n-
a+c+f+n)%100;
            i -= (j+g+e+d+g*l-
f*k+i+d)%100;
        } while( ++DOcnt[51]%5 );
        for( ; ++FORcnt[76]%5 ; )
        {
            d -= (b*c*l*g-n-d*n*c*j-
i+m*b-j+h)%100;
            h -= (n+n-m-m)%100;
            d -= (n+i)%100;
            a = (f*h*e)%100;
            n -= (n-f+f+h-g+g-e)%100;
        }
        for( ; ++FORcnt[77]%5 ; )
        {

```

```

        j -= (d+a+g-k-h+a)%100;
        h += (c-n-k-k*m+n)%100;
        d -= (l+i-a+j+a+j-m+a)%100;
        m -= (g+i)%100;
    }
}
c -= (m+j-j-g-f)%100;
i -= (b-k+j-f-j)%100;
f += (f+f-m-k-e+l)%100;
k -= (k-d+f*i-c-d+e-b+c+f*k)%100;
}
break;

case 2:
{
    l -= (d*n-k-k-g+n-c-c*j-m-g)%100;
    g += (c-m+d+j-h+e*n*d-d)%100;
    k += (a-d-i*d+c)%100;
    d += (c*n-e-f-h+m*f*k+a+l-
i*g+l)%100;
    d -= (c+n-m*j+e+c-c-n-b+k-d-
i)%100;
    a += (l+g-g*h-j+n*a*k+i-k-
b)%100;
}
break;

case 3:
{
    e += (j+n-g-f-j-c+l+f+e)%100;
    l -= (i*n*n+g+a)%100;
    f -= (i+m*e-e-g*d-b*c+e*n*l+g+j-
f)%100;
    l -= (c+g*f-a*n-e-m+f+g+m)%100;
    j -= (m+m*k+b+l-b)%100;
}
break;

default:
{
    m -= (n-g)%100;
}
}

b -= (a*m+l*g+c+n*n+j-
g+e+h+n)%100;
j -= (e*h-c-f+j+h+h+c)%100;
j -= (a*f+j*e+h-n+f-a*e)%100;
k += (c*c)%100;
e = (e-k-b-c)%100;
}
c += (f+f+g+m+h*l+l-m-i)%100;
k += (b+k-a+l+e*e-a+i+h+n+a-b-j-
b)%100;
b -= (e+f)%100;
}
g += (g-j-k-e-m-f*b*d+l)%100;
k += (b*n+a+h-n-h-n)%100;
b += (m-j-d-k+i+m*d+j-l)%100;
j += (b-f-a-g*m+b+g+h*m-d-h-
d+d*j)%100;
i -= (e-i*d+i+b+b-j+j)%100;

```

```

j += (n+e-c+k+d+k+d+m)%100;
k -= (i*c+k-d-d-j)%100;
return
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

int main(void)
{
    int l;
    clock_t StartTick = clock();
    for(l=0; l<57; l++) IFcnt[l] =0;
    for(l=0; l<76; l++) IFECnt[l] =0;
    for(l=0; l<38; l++) SWCnt[l] =0;
    for(l=0; l<76; l++) WHILECnt[l]=0;
    for(l=0; l<76; l++) DOcnt[l] =0;
    for(l=0; l<115; l++) FORcnt[l] =0;
    long int sum=0;

    sum += F1( ) ;
    sum += F2( ) ;
    sum += F3( ) ;
    sum += F4( ) ;
    sum += F5( ) ;
    sum += F6( ) ;
    sum += F7( ) ;
    sum += F8( ) ;
    sum += F9( ) ;
    sum += F10( ) ;
    sum += F11( ) ;
    sum += F12( ) ;
    sum += F13( ) ;
    sum += F14( ) ;
    sum += F15( ) ;
    sum += F16( ) ;
    sum += F17( ) ;
    sum += F18( ) ;
    sum += F19( ) ;
    sum += F20( ) ;
    sum += F21( ) ;
    sum += F22( ) ;
    sum += F23( ) ;

    {
        int a,b,c,d,e,f,g,h,i,j,k,l,m,n;
        a=b=c=d=e=f=g=h=i=j=k=l=m=n=1;
        if( ++IFECnt[52]%2 )
        {
            l += (f*b+h)%100;
            e -= (c*l+n+k+n-h*m+b+k-n+i+l-
d+i)%100;
            j = (a*g+e)%100;
            l += (l*e+k+n+n-k-n*n*h*k)%100;
            m += (f*j-c-k)%100;
        }
        else
        {
            while( ++WHILECnt[52]%5 )
            {
                do
                {
                    m += (c-b*k+j)%100;

```

```

b += (e+b+f-j-a+b*n*j*n-a-f-
b)%100;
l = (l*b+b)%100;
h -= (a-n*c+h+d+c-k*b+i-
n)%100;
m -= (a*m*g*n-n*k+l)%100;
b -= (l+i-j+n-l-a*k+n-f*i-l-j*e-
d)%100;
} while( ++DOcnt[52]%5 );
m -= (n+m+b-d-l*k*m-f+c*a-
g)%100;
d -= (g*b+g+j+n*j+b)%100;
e = (g+j+m*h*g+k-n)%100;
}
n -= (b+k-l-l-m-n-c-k+h+g)%100;
i += (d-d-g+m+m-n-k-m+k+h)%100;
j -= (c+m+i-d+l+b)%100;
h -= (h-j+c*l*b)%100;
}
e -= (k-n-g-e-j*i-e+j)%100;
g -= (n-i-l-c+f*i-i-l+f-c-g*d-i*b)%100;
f += (f+b-j-c-d-n+d-e*l*i*j-
a+i+k)%100;
i -= (n-h*n-e-m+i)%100;
f = (n-a-e-m+d-g-d)%100;
k += (h+j-g-c*i-k-i-m)%100;
n += (c*f-a)%100;
sum +=
(a+b+c+d+e+f+g+h+i+j+k+l+m+n)%100 ;
}

cout << "\nChecksum = " << sum;
for(l=sum=0; l<39; l++) sum += IFcnt[l];
cout << "\nIF frequency: Static = "
<< 39 << " Dynamic = " << sum ;
for(l=sum=0; l<53; l++) sum +=
IFECnt[l];
cout << "\nIF-ELSE frequency: Static =
" << 53 << " Dynamic = " << sum ;
for(l=sum=0; l<26; l++) sum +=
SWCnt[l];
cout << "\nSWITCH frequency: Static
= " << 26 << " Dynamic = " << sum ;
for(l=sum=0; l<53; l++) sum +=
WHILECnt[l];
cout << "\nWHILE frequency: Static =
" << 53 << " Dynamic = " << sum ;
for(l=sum=0; l<53; l++) sum +=
DOcnt[l];
cout << "\nDO frequency: Static = "
<< 53 << " Dynamic = " << sum ;
for(l=sum=0; l<79; l++) sum +=
FORcnt[l];
cout << "\nFOR frequency: Static = "
<< 79 << " Dynamic = " << sum ;
cout << "\nRun Time = " <<
double(clock()-
StartTick)/CLOCKS_PER_SEC << "
sec\n\n";

return 0;
}

```

## e.) Matrix Multiplication Preprocessor

```
/*
 * Class: CSC 840
 * By: Matthew Wishoff
 * Date: 3/4/2016
 * Description: This program multiplies two matrices 48 different ways.
 *              It also steps the matrix size by 20 on each iteration through the 48 ways.
 */
#include <iostream>
#include <stdio.h>
#include <chrono>

using namespace std;

#define DO_MULTIPLY_1 c[i][j] += a[i][k]*b[k][j]
#define DO_MULTIPLY_2 c[i][j] += a[i][k]*b[j][k]
#define DO_MULTIPLY_3 c[i][j] += a[k][i]*b[k][j]
#define DO_MULTIPLY_4 c[i][j] += a[k][i]*b[j][k]
#define DO_MULTIPLY_5 c[j][i] += a[i][k]*b[k][j]
#define DO_MULTIPLY_6 c[j][i] += a[i][k]*b[j][k]
#define DO_MULTIPLY_7 c[j][i] += a[k][i]*b[k][j]
#define DO_MULTIPLY_8 c[j][i] += a[k][i]*b[j][k]
#define DO_MULTIPLY_i "Hello Hari"

#define DO_MULTIPLY_IJK(method) \
    for(int m=0; m<M; m++) \
        for(int i=0; i<n; i++) \
            for(int j=0; j<n; j++) \
                for(int k=0; k<n; k++) \
                    DO_MULTIPLY_##method;

#define DO_MULTIPLY_IKJ(method) \
    for(int m=0; m<M; m++) \
        for(int i=0; i<n; i++) \
            for(int k=0; k<n; k++) \
                for(int j=0; j<n; j++) \
                    DO_MULTIPLY_##method;

#define DO_MULTIPLY_KJI(method) \
    for(int m=0; m<M; m++) \
        for(int k=0; k<n; k++) \
            for(int j=0; j<n; j++) \
                for(int i=0; i<n; i++) \
                    DO_MULTIPLY_##method;

#define DO_MULTIPLY_KIJ(method) \
    for(int m=0; m<M; m++) \
        for(int k=0; k<n; k++) \
            for(int i=0; i<n; i++) \
                for(int j=0; j<n; j++) \
                    DO_MULTIPLY_##method;

#define DO_MULTIPLY_JKI(method) \
    for(int m=0; m<M; m++) \
        for(int j=0; j<n; j++) \
            for(int k=0; k<n; k++) \
                for(int i=0; i<n; i++) \
                    DO_MULTIPLY_##method;

#define DO_MULTIPLY_JIK(method) \
    for(int m=0; m<M; m++) \
        for(int j=0; j<n; j++) \
            for(int i=0; i<n; i++) \
                \
```

```

                                for(int k=0; k<n; k++) \
                                    DO_MULTIPLY_##method;

#define MULTIPLY(method)

                                \
                                startTime = chrono::steady_clock::now();

                                \
                                DO_MULTIPLY_IJK(method);

                                \
                                stopTime = chrono::steady_clock::now();

                                \
                                RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M; \

                                \
                                startTime = chrono::steady_clock::now(); \

                                \
                                DO_MULTIPLY_IKJ(method);

                                \
                                stopTime = chrono::steady_clock::now();

                                \
                                RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M; \

                                \
                                startTime = chrono::steady_clock::now(); \

                                \
                                DO_MULTIPLY_KJI(method);

                                \
                                stopTime = chrono::steady_clock::now();

                                \
                                RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M; \

                                \
                                startTime = chrono::steady_clock::now(); \

                                \
                                DO_MULTIPLY_KIJ(method);

                                \
                                stopTime = chrono::steady_clock::now();

                                \
                                RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M; \

                                \
                                startTime = chrono::steady_clock::now(); \

                                \

```

```

DO_MULTIPLY_JKI(method);

\
stopTime = chrono::steady_clock::now();

\
RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M; \

\
startTime = chrono::steady_clock::now();

\
DO_MULTIPLY_JIK(method);

\
stopTime = chrono::steady_clock::now();

\
RUN_TIME[TEST_NUM++ % NUM_OF_TESTS][col] = chrono::duration_cast<chrono::duration<double>>(stopTime -
startTime).count() / M;

#define NUM_OF_TESTS (8*6)
#define NMAX 500
#define NMIN 100
#define STEP 20

int TEST_NUM = 0;
double RUN_TIME[NUM_OF_TESTS][((NMAX - NMIN) / STEP) + 1];

int main(int argc, char**argv)
{
    chrono::steady_clock::time_point startTime, stopTime;
    int col = 0;
    int M ;

    // Initialize
    for(int n=NMIN; n <= NMAX; n+=STEP, col++) //increase size of matrix
    {
        M=(NMAX*NMAX*NMAX)/(n*n*n);
        if(M < 4)
        {
            M = 4;
        }
        double** a = new double*[n]; //Initialize to the min, the next iteration will step by 20
        double** b = new double*[n]; //Initialize to the min, the next iteration will step by 20
        double** c = new double*[n]; //Initialize to the min, the next iteration will step by 20

        cout << "before init" << endl;
        for(int i = 0; i < n; i++)
        {
            a[i] = new double[n];
            b[i] = new double[n];
            c[i] = new double[n];
        }

        // Matrix a[ ][ ], b[ ][ ], and c[ ][ ] initialization
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
            {
                //Initialize diagonal to 2.0002, and everywhere else to 1.0001
                if(i == j)
                {
                    a[i][j] = 2.0002;
                    b[i][j] = 2.0002;

```

```

        }
        else
        {
            a[i][j] = 1.0001;
            b[i][j] = 1.0001;
        }
        c[i][j]=0.0;
    }

    for(int i = 0; i < 7; i ++)
    {
        MULTIPLY(i);

        MULTIPLY(1); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(2); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(3); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(4); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(5); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(6); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(7); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array
        MULTIPLY(8); //multiply 6 different matrices and save their run times to the RUN_TIME 2d array

        //Delete dynamic array of pointers.
        for(int i = 0; i < n; i++)
        {
            delete[] a[i];
            delete[] b[i];
            delete[] c[i];
        }
        delete[] a;
        delete[] b;
        delete[] c;
    }

    //Pipe the RUN_TIME 2D array out into a text file.
    freopen ("MatrixMultiplicationWithBatteryPower5Iterations.txt","w",stdout);

    for(int i = 0; i < NUM_OF_TESTS; i++) // this is correct
    {
        for(int j = 0; j <= (NMAX - NMIN) / STEP; j++)
        {
            if(j == (NMAX - NMIN) / STEP)
            {
                cout << RUN_TIME[i][j];
            }
            else
            {
                cout << RUN_TIME[i][j] << ", "; //Print a row of data
            }
        }
        cout << endl;
    }
    cout << endl;
    fclose (stdout);
}

```



## f.) Matrix Multiplication Standard

```
/*
 * Class: CSC 840
 * By: Matthew Wishoff
 * Date: 3/4/2016
 * Description: Standard way of multiplying matrices
 */

#include <iostream>
#include <stdio.h>
#include <chrono>

#define NUM_OF_TESTS 48
#define NMAX 500
#define NMIN 100
#define STEP 20

//Create 2d Array RUN_TIME and store values then you might be done.

int TEST_NUM = 0;
double RUN_TIME[NUM_OF_TESTS][((NMAX - NMIN) / STEP) + 1];

using namespace std;

int main(int argc, char**argv)
{
    chrono::steady_clock::time_point startTime, stopTime;
    int col = 0;
    int M;

    // Initialize
    for(int n=NMIN; n <= NMAX; n+=STEP, col++) //increase size of matrix
    {
        M=(NMAX*NMAX*NMAX)/(n*n*n);
        if(M < 4)
        {
            M = 4;
        }
        double** a = new double*[n]; //Initialize to the min, the next iteration will step by 20
        double** b = new double*[n]; //Initialize to the min, the next iteration will step by 20
        double** c = new double*[n]; //Initialize to the min, the next iteration will step by 20

        cout << "before init" << endl;
        for(int i = 0; i < n; i++)
        {
            a[i] = new double[n];
            b[i] = new double[n];
            c[i] = new double[n];
        }

        // Matrix a[ ][ ], b[ ][ ], and c[ ][ ] initialization
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
            {
                //Initialize diagonal to 2.0002, and everywhere else to 1.0001
                if(i == j)
                {
                    a[i][j] = 2.0002;
                    b[i][j] = 2.0002;
                }
                else
                {

```

```

        a[i][j] = 1.0001;
        b[i][j] = 1.0001;
    }
    c[i][j]=0.0;
}

//*****
//*****IJK LOOPS
START*****
    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                for(int k=0; k<n; k++)
                {
                    c[i][j] += a[i][k]*b[k][j];
                }
    stopTime = chrono::steady_clock::now();
    RUN_TIME[0][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                for(int k=0; k<n; k++)
                {
                    c[i][j] += a[i][k]*b[j][k];
                }
    stopTime = chrono::steady_clock::now();
    RUN_TIME[1][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                for(int k=0; k<n; k++)
                {
                    c[i][j] += a[k][i]*b[k][j];
                }
    stopTime = chrono::steady_clock::now();
    RUN_TIME[2][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                for(int k=0; k<n; k++)
                {
                    c[i][j] += a[k][i]*b[j][k];
                }
    stopTime = chrono::steady_clock::now();
    RUN_TIME[3][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                for(int k=0; k<n; k++)
                {
                    c[j][i] += a[i][k]*b[k][j];
                }
    stopTime = chrono::steady_clock::now();
    RUN_TIME[4][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();

```

```

for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            for(int k=0; k<n; k++)
                {
                    c[j][i] += a[i][k]*b[j][k];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[5][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            for(int k=0; k<n; k++)
                {
                    c[j][i] += a[k][i]*b[k][j];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[6][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int j=0; j<n; j++)
            for(int k=0; k<n; k++)
                {
                    c[j][i] += a[k][i]*b[j][k];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[7][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

//*****
//*****IJK LOOPS
END*****

//*****
//*****IKJ LOOPS
START*****

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                {
                    c[i][j] += a[i][k]*b[k][j];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[8][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                {
                    c[i][j] += a[i][k]*b[j][k];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[9][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)

```

```

        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                for(int j=0; j<n; j++)
                {
                    c[i][j] += a[k][i]*b[k][j];
                }
stopTime = chrono::steady_clock::now();
RUN_TIME[10][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
            {
                c[i][j] += a[k][i]*b[j][k];
            }
stopTime = chrono::steady_clock::now();
RUN_TIME[11][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
            {
                c[j][i] += a[i][k]*b[k][j];
            }
stopTime = chrono::steady_clock::now();
RUN_TIME[12][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
            {
                c[j][i] += a[i][k]*b[j][k];
            }
stopTime = chrono::steady_clock::now();
RUN_TIME[13][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
            {
                c[j][i] += a[k][i]*b[k][j];
            }
stopTime = chrono::steady_clock::now();
RUN_TIME[14][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int i=0; i<n; i++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
            {
                c[j][i] += a[k][i]*b[j][k];
            }
stopTime = chrono::steady_clock::now();
RUN_TIME[15][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

```

```
//*****
```

```

//*****IKJ LOOPS
END*****

//*****
//*****KJI LOOPS
START*****

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[i][j] += a[i][k]*b[k][j];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[16][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[i][j] += a[i][k]*b[j][k];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[17][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[i][j] += a[k][i]*b[k][j];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[18][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[i][j] += a[k][i]*b[j][k];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[19][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[j][i] += a[i][k]*b[k][j];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[20][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)
                    c[j][i] += a[i][k]*b[j][k];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[21][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int j=0; j<n; j++)
                for(int i=0; i<n; i++)

```

```

                                c[j][i] += a[k][i]*b[k][j];
stopTime = chrono::steady_clock::now();
RUN_TIME[22][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int k=0; k<n; k++)
        for(int j=0; j<n; j++)
            for(int i=0; i<n; i++)
                c[j][i] += a[k][i]*b[j][k];
stopTime = chrono::steady_clock::now();
RUN_TIME[23][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

//*****
//*****KJI LOOPS
END*****

//*****
//*****KIJ LOOPS
START*****

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                for(int j=0; j<n; j++)
                    c[i][j] += a[i][k]*b[k][j];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[24][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                for(int j=0; j<n; j++)
                    c[i][j] += a[i][k]*b[j][k];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[25][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                for(int j=0; j<n; j++)
                    c[i][j] += a[k][i]*b[k][j];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[26][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                for(int j=0; j<n; j++)
                    c[i][j] += a[k][i]*b[j][k];

    stopTime = chrono::steady_clock::now();
    RUN_TIME[27][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

    startTime = chrono::steady_clock::now();
    for(int m=0; m<M; m++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                for(int j=0; j<n; j++)
                    c[j][i] += a[i][k]*b[k][j];

```

```

stopTime = chrono::steady_clock::now();
RUN_TIME[28][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int k=0; k<n; k++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                c[j][i] += a[i][k]*b[j][k];
stopTime = chrono::steady_clock::now();
RUN_TIME[29][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int k=0; k<n; k++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                c[j][i] += a[k][i]*b[k][j];
stopTime = chrono::steady_clock::now();
RUN_TIME[30][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int k=0; k<n; k++)
        for(int i=0; i<n; i++)
            for(int j=0; j<n; j++)
                c[j][i] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[31][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

//*****
//*****KIJ LOOPS
START*****

//*****
//*****JKI LOOPS
START*****

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[i][j] += a[i][k]*b[k][j];
stopTime = chrono::steady_clock::now();
RUN_TIME[32][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[i][j] += a[i][k]*b[j][k];
stopTime = chrono::steady_clock::now();
RUN_TIME[33][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[i][j] += a[k][i]*b[k][j];
stopTime = chrono::steady_clock::now();

```

```

RUN_TIME[34][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[i][j] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[35][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[j][i] += a[i][k]*b[k][j];

stopTime = chrono::steady_clock::now();
RUN_TIME[36][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[j][i] += a[i][k]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[37][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[j][i] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[38][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int k=0; k<n; k++)
            for(int i=0; i<n; i++)
                c[j][i] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[39][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

//*****
//*****JKI LOOPS
END*****

//*****
//*****JIK LOOPS
START*****

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[i][j] += a[i][k]*b[k][j];

stopTime = chrono::steady_clock::now();
RUN_TIME[40][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

```



```

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[i][j] += a[i][k]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[41][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[i][j] += a[k][i]*b[k][j];

stopTime = chrono::steady_clock::now();
RUN_TIME[42][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[i][j] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[43][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[j][i] += a[i][k]*b[k][j];

stopTime = chrono::steady_clock::now();
RUN_TIME[44][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[j][i] += a[i][k]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[45][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[j][i] += a[k][i]*b[k][j];

stopTime = chrono::steady_clock::now();
RUN_TIME[46][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

startTime = chrono::steady_clock::now();
for(int m=0; m<M; m++)
    for(int j=0; j<n; j++)
        for(int i=0; i<n; i++)
            for(int k=0; k<n; k++)
                c[j][i] += a[k][i]*b[j][k];

stopTime = chrono::steady_clock::now();
RUN_TIME[47][col] = chrono::duration_cast<chrono::duration<double>>(stopTime - startTime).count() / M;

```

```
//*****
```

```

//*****JIK LOOPS
END*****

```

```

        for(int i = 0; i < n; i++)
        {
            delete[] a[i];
            delete[] b[i];
            delete[] c[i];
        }
        delete[] a;
        delete[] b;
        delete[] c;
    }

    //Pipe the RUN_TIME 2D array out into a text file.
    freopen ("MatrixMultiplicationStandardBatteryPower5Iterations.txt", "w", stdout);

    for(int i = 0; i < NUM_OF_TESTS; i++) // this is correct
    {
        for(int j = 0; j <= (NMAX - NMIN) / STEP; j++)
        {
            if(j == (NMAX - NMIN) / STEP)
            {
                cout << RUN_TIME[i][j];
            }
            else
            {
                cout << RUN_TIME[i][j] << ", "; //Print a row of data
            }
        }
        cout << endl;
    }
    cout << endl;
    fclose (stdout);
}

```

G.)

```
// By: Matthew Wishoff
// Date: 3/21/16
// Class: CSC 840
// Description:
```

```
#include <iostream>
#include <fstream>
#include <stdio.h>
#include <chrono>
#include <regex>
```

```
bool oneLiners(char s[]);
```

```
using namespace std;
```

```
// Counting LLOC?
```

```
// NSC = Number of semi colons
```

```
// NEQ = Number of Equals signs
```

```
// LLOC += (NSC + NEQ); // NSC = 1
```

```
// 1. We count logical lines of code and
denote their number as LLOC.
```

```
// 2. Each logical statement counts as
one, regardless its complexity or level of
nesting.
```

```
// 3. All comments are ignored: in C++,
both single line [//] and multiple line
[/...*/].
```

```
// 4. The logical statement counting
formula (based on token analysis) is the
following:
```

```
        // LLOC = NPU // main
program plus all functions
        // + NSC // in the whole
program, except comments
        // + NEQ // data definitions
only (constructor-initializer)
        // + Nif // all if statements
        // + Nswitch // all switch
statements
        // + Nwhile // all while
statements
        // + Nfor // all for statements
        bool operatorCheck(string
line);
```

```
        regex And("&&");
        regex Or("||");
        regex colon(":".);
```

```
int main(int argc, char**argv)
{
```

```
    string fileName;
```

```
    int PLOC = 0;
    int LLOC = 0;
    int ifCnt = 0;
    int whileCnt = 0;
    int switchCnt = 0;
    int forCnt = 0;
    int semiColonCnt = 0;
    int equalSignCnt = 0;
    int comment = 0;
```

```
    int function = 0;
    int v = 0;
    string currentLine;

    // Read in C++ file
    cout << "Enter the file name:
";
    cin >> fileName;
    ifstream C_File;
    C_File.open(fileName);
```

```
    regex Function("int(.*)");
    // regex
    Function("^\\s*[\\w_][\\w\\d]*\\s*.*\\s*[\\w_][\\w\\d]*\\s*{.*}\\s*");
    // regex
    oneLiner("([{}]*)([{}]*)([{}]*);");
    // regex
    oneLiner("^([{}];){3}.*$");
    // regex
    oneLiner(".*for(.*,*,*,+;+);");
    // regex
    oneLiner("for{([{}]*;){3};[{}]*}");
    ;{3}");
    // regex
    oneLiner("for*{([{}]*;){3};[{}]*}");
    [{}];"); // Regex for a forloop
    regex If("if"); //Need to
make the definition more robust!
    regex For("for"); //Need to
make the definition more robust!
    regex While("while"); //Need
to make the definition more robust!
    regex Switch("switch");
//Need to make the definition more
robust!

    //Currently ignores +=, -=,
*=, /= expressions.
    regex Equals("[^+*-/]=");
//Need to make the definition more
robust!
    //(";|;.*|;+)" <--- Accepts
everything that is passed in???
    regex SemiColon(";",); //Need
to make the definition more robust!
```

```
    //Do this last, has least
impact on the program.
    regex Comments("//");
//Need to make the definition more
robust!
    regex
MultiCommentStart("/(*)*");
// regex
MultiCommentEnd("(*)/"); //Breaks
regex whyyyy?

    // Go line by line counting
lines of code.
    while(!C_File.eof())
    {
```

```
        // cout <<
        "*****Next
line*****"
        << endl;

        //take a line from
the program
        getline(C_File,
currentLine);
        ++PLOC;
        cout <<
currentLine << endl;

        // cout <<
        "*****current
line*****" <<
        endl;

        //
        *****
        *****
        *****
        *****
        //
        if we see a /* we
see that a multi line comment has
started.

        if(regex_match(currentLine,
MultiCommentStart))
        {
            //we
continue to go through the file until we
see a */ signifying the comment has
ended.
            //or
the end of file is reached

            while(regex_match(currentLi
ne, MultiCommentEnd) &&
!C_File.eof())
            {
                getline(C_File, currentLine);
            }
        }

        // Probably need to make sure
that if a multi line comment is at the
end of a file this logic does not break it.

        //
        *****
        *****
        *****
        *****
        //wont work if
comments are on the same line?

        if(regex_match(currentLine,
Comments))
        {
            //Do
nothing
        }
```

```

else //increment
appropriate counters.
{
    if(regex_search(currentLine,
If))
    {
        LLOC++;
        ifCnt++;
        v++;
        if(operatorCheck(currentLine
))
        {
            v++;
            if(regex_search(currentLine, Switch))
            {
                //
                cout << "If" << endl;
            }
            else
            if(regex_search(currentLine, For))
            {
                LLOC++;
                forCnt++;
                v++;
                if(operatorCheck(currentLine
))
                {
                    v++;
                }
            }
            //
            cout << "For" << endl;
        }
        else
        if(regex_search(currentLine, While))
        {
            //
            LLOC++;
            whileCnt++;
            v++;
            if(operatorCheck(currentLine
))
            {
                v++;
                cout << "While" << endl;
            }
            else
            if(regex_search(currentLine, Switch))
            {
                LLOC++;
                switchCnt++;
                if(operatorCheck(currentLine
))
                {
                    v++;
                }
                cout << "Switch" << endl;
            }
        }
        C_File.close();
        //
        freopen
        ("BMProgram25.txt", "w", stdout);
        cout <<
        "=====
        =" << endl;
    }
}

cout << fileName << endl;
cout << "PLOC Count: " <<
PLOC << endl;
cout << "LLOC Count: " <<
LLOC << endl;
cout << "Cyclomatic
Complexity: " << v << endl;
cout << "If statement Count:
" << ifCnt << endl;
cout << "for loop Count: " <<
forCnt << endl;
cout << "while loop Count: "
<< whileCnt << endl;
cout << "switch statement
Count: " << switchCnt << endl;
cout << "function statement
Count: " << function << endl;
cout << "Semi Colon Count: "
<< semiColonCnt << endl;
cout << "Equal sign Count "
<< equalSignCnt << " - - (Data def only)"
<< endl;
cout <<
"=====
=" << endl;
//
fclose (stdout);

return 0;
}

bool operatorCheck(string line)
{
    if(regex_search(line, And))
    {
        return true;
    }
    else if(regex_search(line, Or))
    {
        return true;
    }
    else if(regex_search(line,
colon))
    {
        return true;
    }
    return false;
}

```