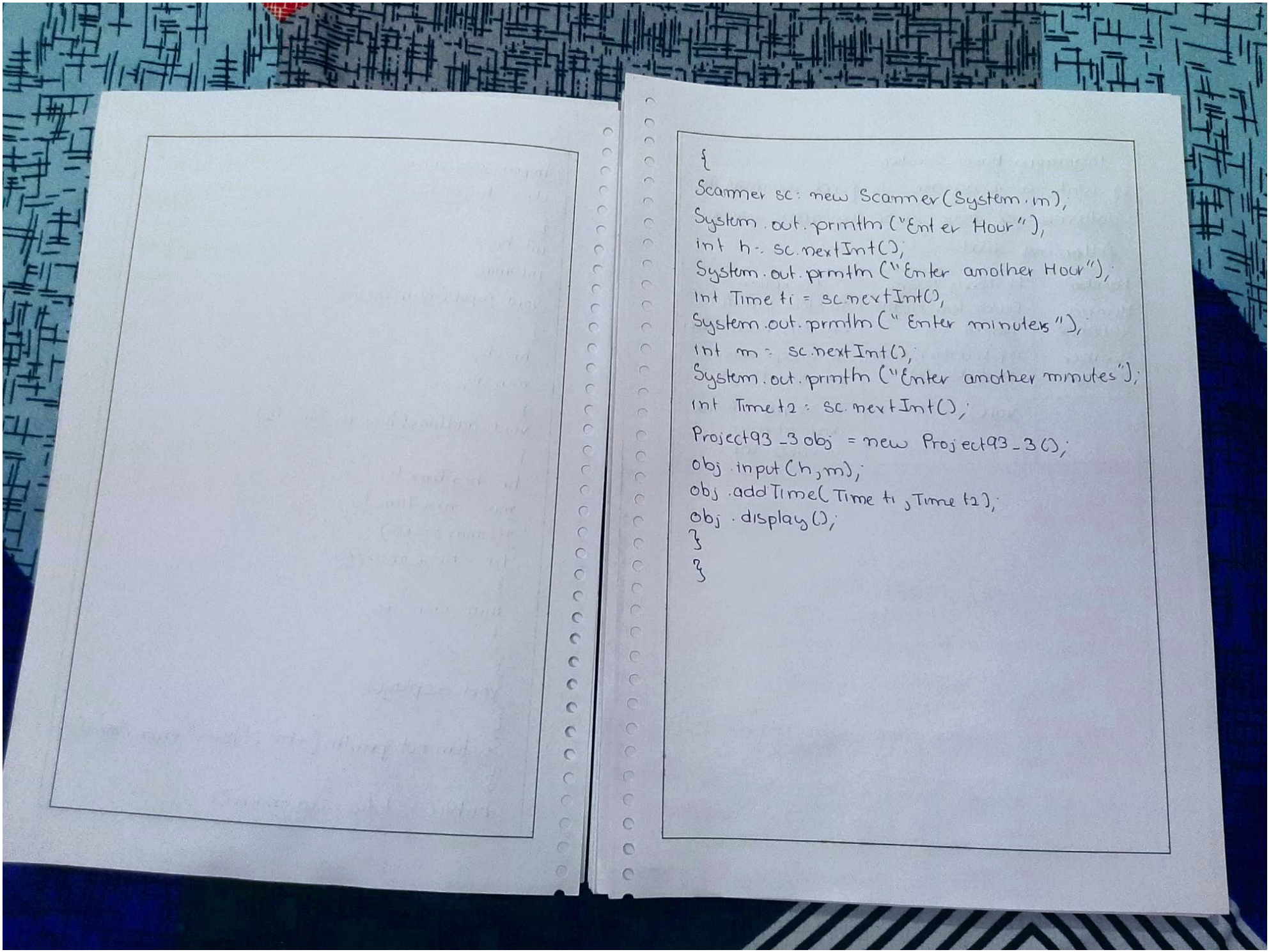
Programing Based Programs

1) Write a program to add two different time variables and display the output using following details in class time.

Members	Member Name	Description
member varibles	int hr, min	To store hour & minutes
member method	input(int h, int m) addTime(Time t1, Time t2)	To accept hours & minutes & add.
	display	To display time.

Eg 3hrs 40min + 3hrs 40min = 7hrs 20min

```
import java.util.*;
class Project93_3
{
int hr;
int min;
void input(int h, int m)
{
hr=h;
min=m;
}
void addTime(Time t1, Time t2)
{
hr=hr+Time t1;
min=m+Time t2;
if(min>=60)
hr=hr+min/60;
{
min=min%60;
}
}
void display()
{
System.out.println(+hr+"hours"+min+"minutes");
}
public static void main()
```



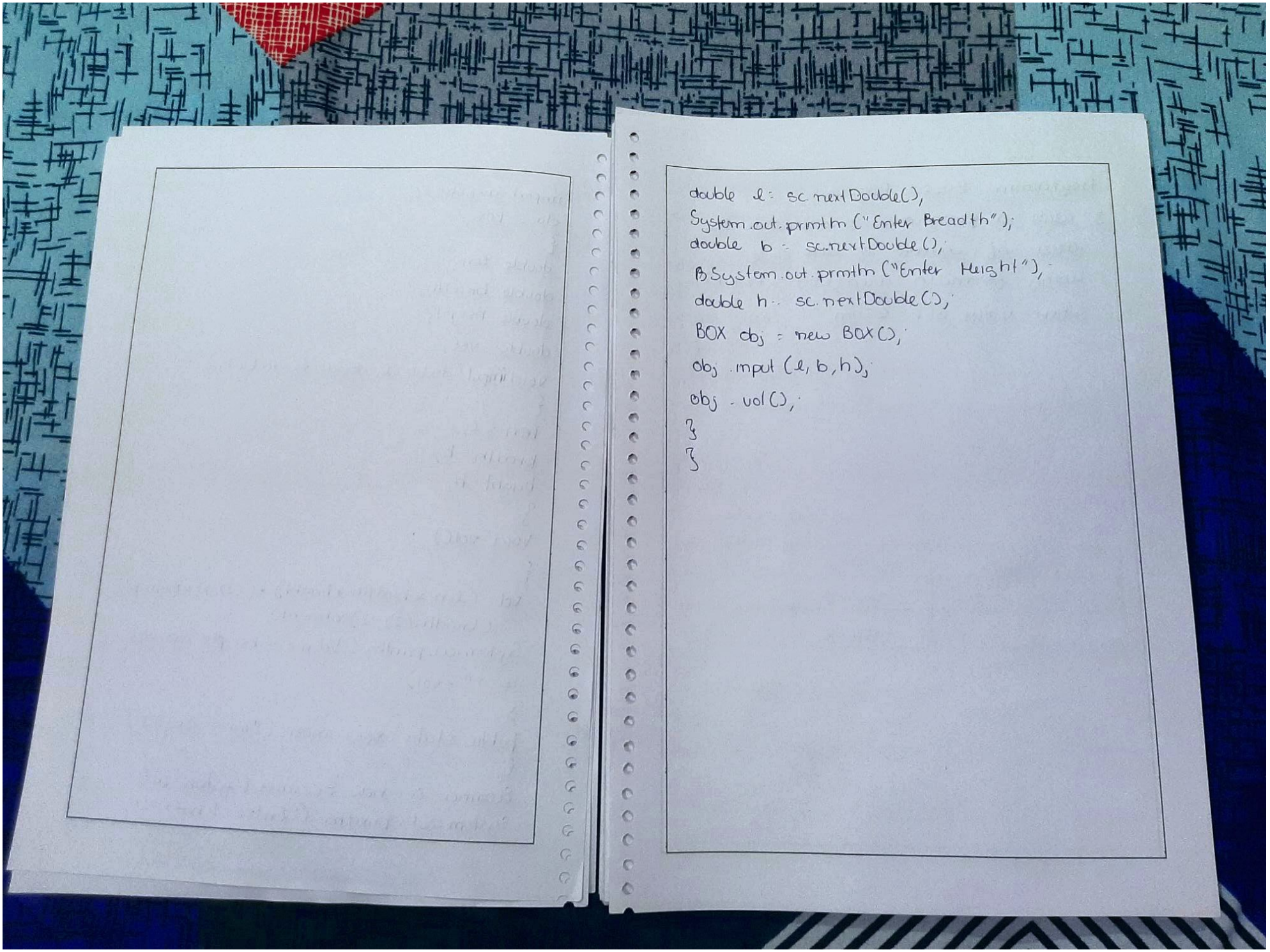
```
{
Scanner sc: new Scanner(System.in),
System.out.println("Enter Hour"),
int h:- sc.nextInt();
System.out.println("Enter another Hour");
Int Time t1 = sc.nextInt(),
System.out.println(" Enter minutes"),
Int m = sc.nextInt(),
System.out.println("Enter another minutes");
Int Time t2 = sc.nextInt();
Project93_3 obj = new Project93_3();
obj.input(h,m);
obj.addTime(Time t1, Time t2);
obj.display();
}
}
```

Programming Based Questions

2) Write a program to find sum of the volumes of box and cylinder using following details

Members	Member's Name	Description
Member variables	Double len, breath, height, vol	To store length, breadth, height, volume
Member method	input (double l, double b, double h)	To accept length, breadth, height
	vol()	To compute volume and store in variable vol.

```
import java.util.*;
class BOX
{
double len;
double breadth;
double height;
double vol;
void input (double l, double b, double h)
{
len = l;
breadth = b;
height = h;
}
void vol()
{
vol = (len * breadth * height) + ((22/7) * Math.pow
    (breadth/2), 2) * height);
System.out.println ("Volume of box plus cylinder
is :" + vol);
}
public static void main (String arg[])
{
Scanner sc = new Scanner (System.in);
System.out.println ("Enter length");
```


double l = sc.nextDouble();
System.out.println("Enter Breadth");
double b = sc.nextDouble();
System.out.println("Enter Height");
double h = sc.nextDouble();
BOX obj = new BOX();
obj.input(l,b,h);
obj.vol();
}
}

Programing Based Programs

3) Write a program to calculate the area of ~~square~~ ~~of~~ ~~side~~ circle using formula area (ar) = 3.14 * r2 where value of r = 25cm

```
//import java.util.*;
class project 117_7
{
public static void main (String arg[])
{
double r = 25;
double ar;
double ar = 3.14 * r * r;
System.out.println ("Area of circle is" + ar + "cm²");
}
}
```

Variable description

Variable	datatype	description
r	double	to store radius value
ar	double	to store area of circle

Programing Based Programs

4) Write a java class date and define a constructor for the class for the following specifications.

Data members	Member methods
Day (dd)	default: Initialise date
Month (mm)	parameterized constructor: Initialise date of birth
year (yy)	yearplus(): to increase year of date by 1.
	display (): display date

```
import java.util.*;
class  Date
{
int dd1;
int mm1;
int yy1;
Date (int dd, int mm, int yy)
{
dd1 = dd;
mm1 = mm;
yy1 = yy;
}
void Yearplus()
{
System.out.println("Year Plus=" + dd1 + "/"
+ mm1 + "/" + (yy1+1));
}
public static void main(String args[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter Day");
int dd = sc.nextInt();
```

```
System.out.println("Enter month");
int mm = sc.nextInt();
System.out.println("Enter year");
int yy = sc.nextInt();
Date obj = new Date(dd, mm, yy)
obj.yearplus();
}
}
```

Programming Based Programs

5) Write a java program to define a class and perform the following from the given specification using constructor.

Data members	Member methods
Workernumber (wno)	calc() - to calculate
Hour worked (hwk)	init() - to initialise
Work rate (wrate)	display - to o/p result
Total wage (twage)	

```
import java.util.*;
class Worker
{
int wno;
int hwk;
int wrate;
int twage;
Worker()
{
wno = wno1;
hwk = hwk1;
}
void calc()
{
wrate = 100;
twage = twage + wrate * hwk;
}
void init()
{
Scanner sc = new Scanner (System.in);
System.out.println ("Enter worker number");
int wno1 = sc.nextInt();
System.out.println ("Enter hour worked");
int hwk1 = sc.nextInt();
```

```
}
void display()
{
System.out.println("Worker number:" +wno);
System.out.println("Hour worked:" +nwk);
System.out.println("Work rate:" +wrate);
System.out.println("Total wage:" +twage);
}
public static void main(string arg[])
{
Worker obj = new worker();
obj.calc();
obj.init();
obj.display;
}
}
```

Programming Based programs

6) Define a class Student with the following specifications:

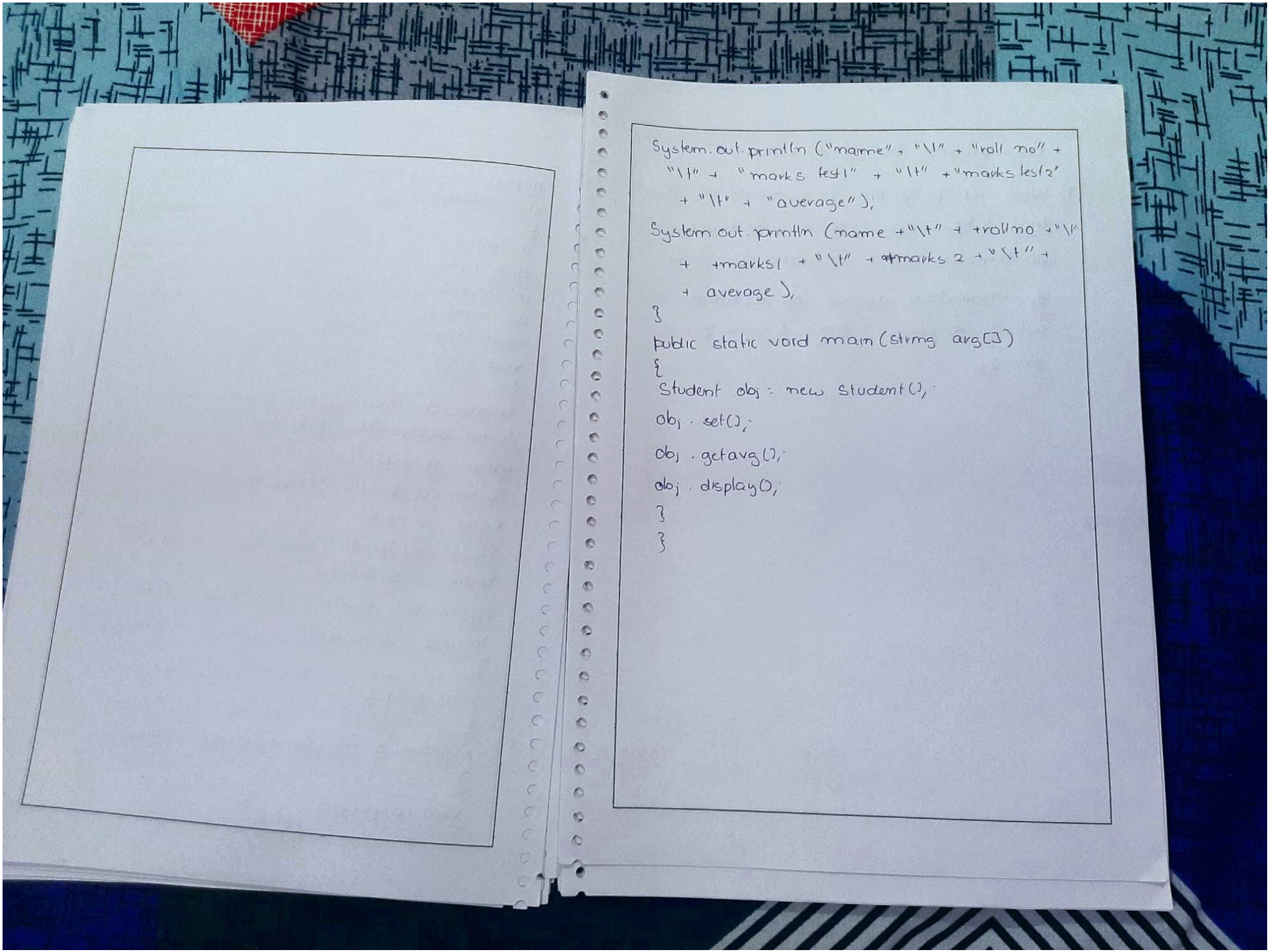
Private members

roll no	2 digit roll number
name	20 characters
marks1	total marks in test 1
marks2	total marks in test 2
average	average marks obtained

Public members

set()	method to accept values for data members & to invoke getavg()
getavg()	method to compute average marks obtained in 2 tests
display()	method to display all data members

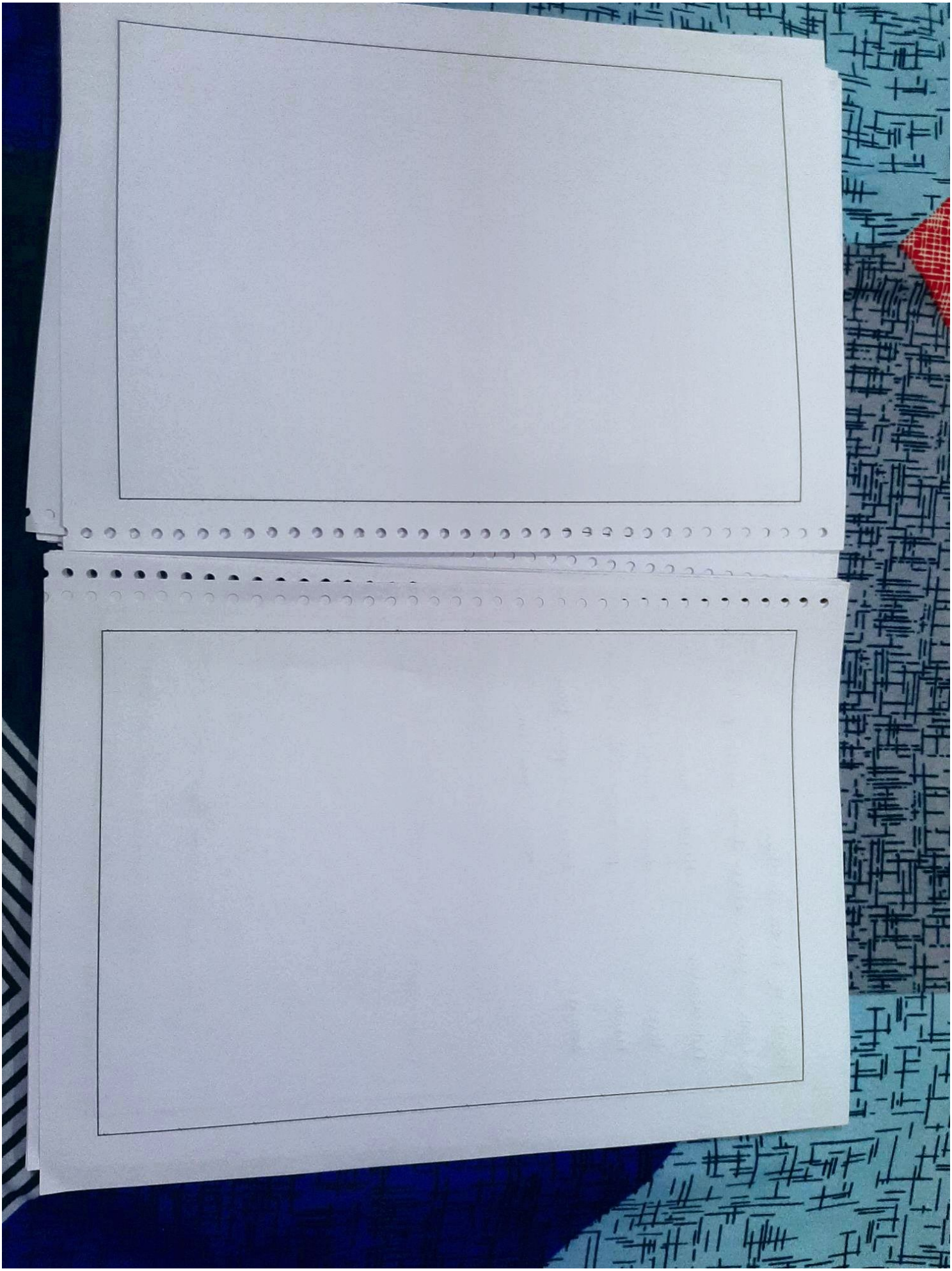
```
import java.util.Scanner;
class Student
{
int rollno;
String name;
int marks1;
int marks2;
double average
Void set()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter Roll no.");
rollno = sc.nextInt();
System.out.println("Enter name");
name = sc.next();
System.out.println("Enter marks in test 1");
marks1 = sc.nextInt();
System.out.println("Enter marks in test2");
marks2 = sc.nextInt();
}
void getavg()
{
 System.o average = (marks1+marks2)/2;
}
void display()
{
```



```
System.out.println ("name" + "\t" + "roll no" +
 "\t" + "marks test1" + "\t" +"marks test2"
   + "\t" + "average");
System.out.println (name +"\t" + +rollno +"\t"
   + +marks1 + "\t" + +marks 2 +"\t" +
   + average );
}
public static void main (String arg[])
{
Student obj = new Student();
obj.set();
obj.getavg();
obj.display();
}
}
```

Programing Based Questions

7) Write a program to ~~generate~~ 20 Java that takes an English word as parameter and returns the same of number value of alphabets which A starts from 1 and zends on 26.

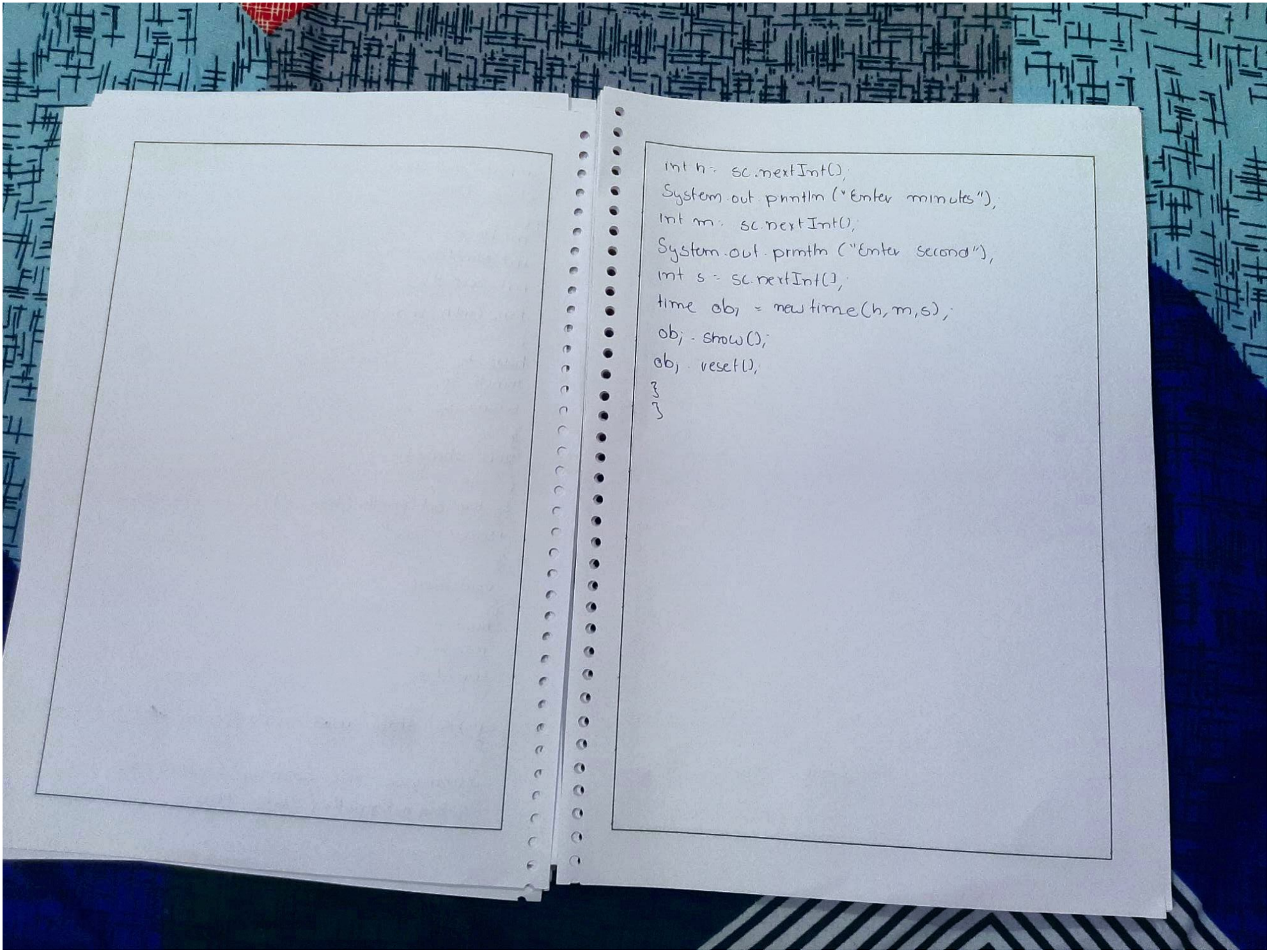


Programming Based Questions

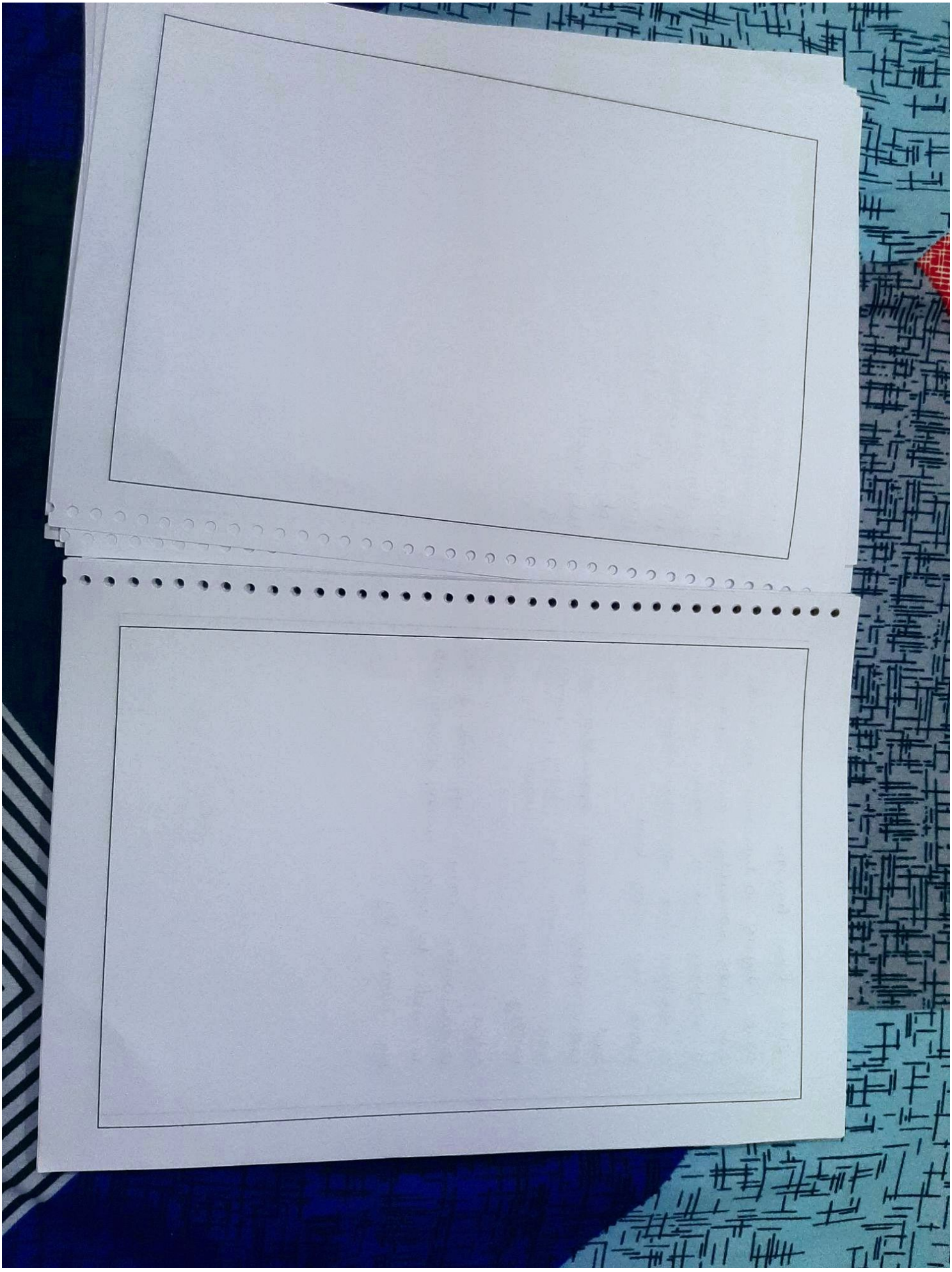
8) Write a java class time with following

Datamember	Member function
Hours	default: initialise time
Minutes	parameterised cons: initialise Time
Seconds	show(): display time
	reset(): reset time

```
import java.util.*;
class time
{
int Hours;
int Minutes,
int Seconds;
time (int h, int m, int s)
{
hours = h;
minutes = m;
second = s;
}
void show()
{
System.out.println (hours + "hrs:" + minutes + "min:"
+ second + "sec:");
}
void reset()
{
hours = 0;
minutes = 0;
second = 0;
}
public static void main (String arg[])
{
Scanner sc = new Scanner (System.in);
System.out.println ("Enter hours");
```



```
int h = sc.nextInt();
System.out.println("Enter minutes");
int m = sc.nextInt();
System.out.println("Enter second");
int s = sc.nextInt();
time ob1 = new time(h,m,s);
ob1.show();
ob1.reset();
}
}
```



String Based Programs

1) While typing a typewriter has made 2 or more consecutive blank spaces in a sentence. write a program in java to replace two or more consecutive blank by a single blank.

Input:
Understanding computer applications for ICSE is written by Vijay Kumar pandey and Dilip Kumar Dey

Output:
Understanding computer applications for ICSE is written by vijay Kumar pandey and Dilip Kumar Dey.

```
import java.util.*;
class blanks
{
public static void main(String arg[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter a Sentence");
String sen = sc.nextLine();
sen = sen.replaceAll("\\s+", " ");
System.out.println(sen);
}
}
```

Variable Description

Variable	Datatype	Description
sen	String	to stor user input

String Based Programs

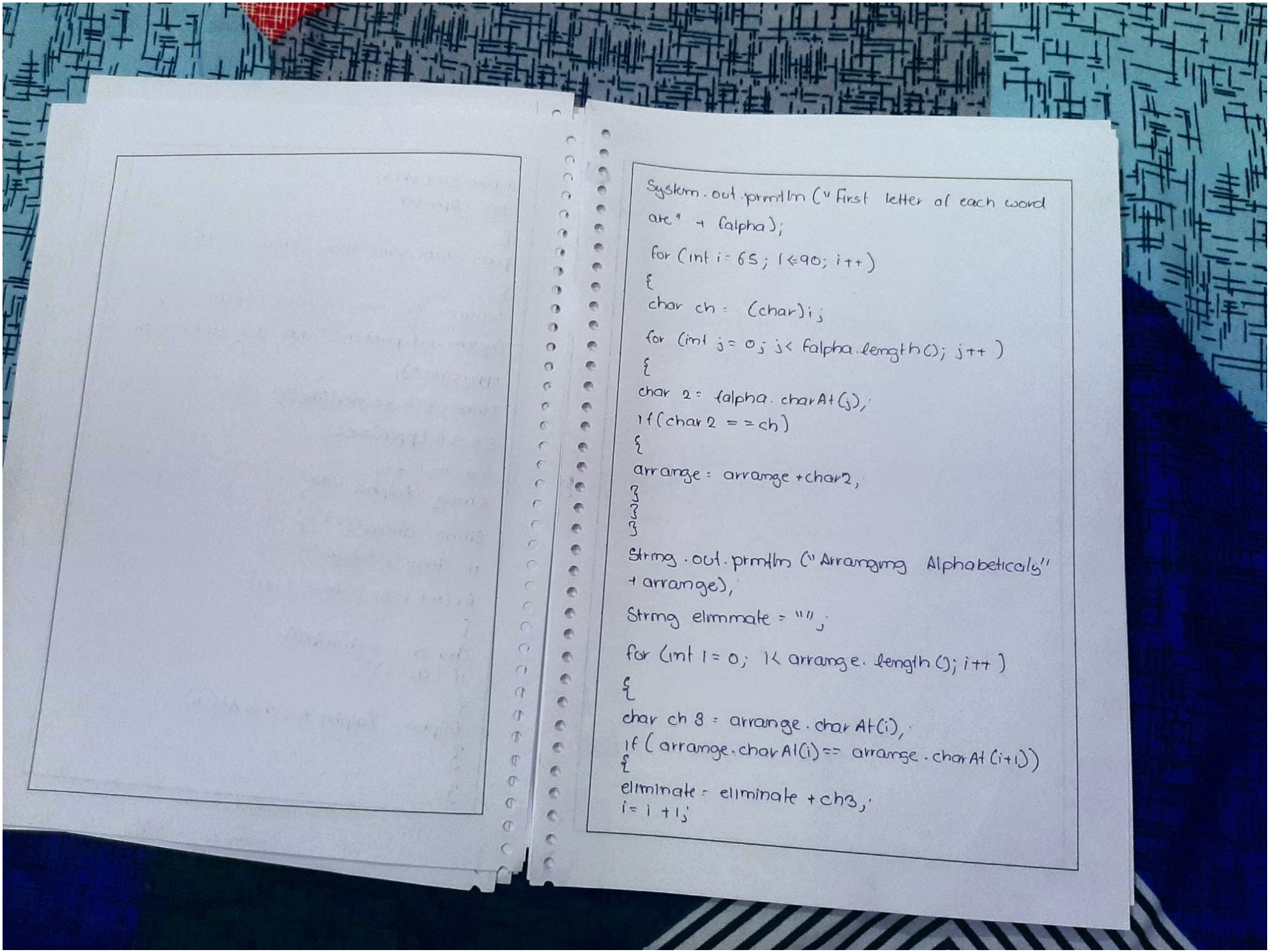
2) Write a program to input a sentence in uppercase and frame a word by adding the first alphabet of each word & also arrange the alphabet of the new word in an alphabetical order also remove the duplicating alphabets.

Input:
NARENDRA MODI IS THE PRIME MINISTER OF INDIA

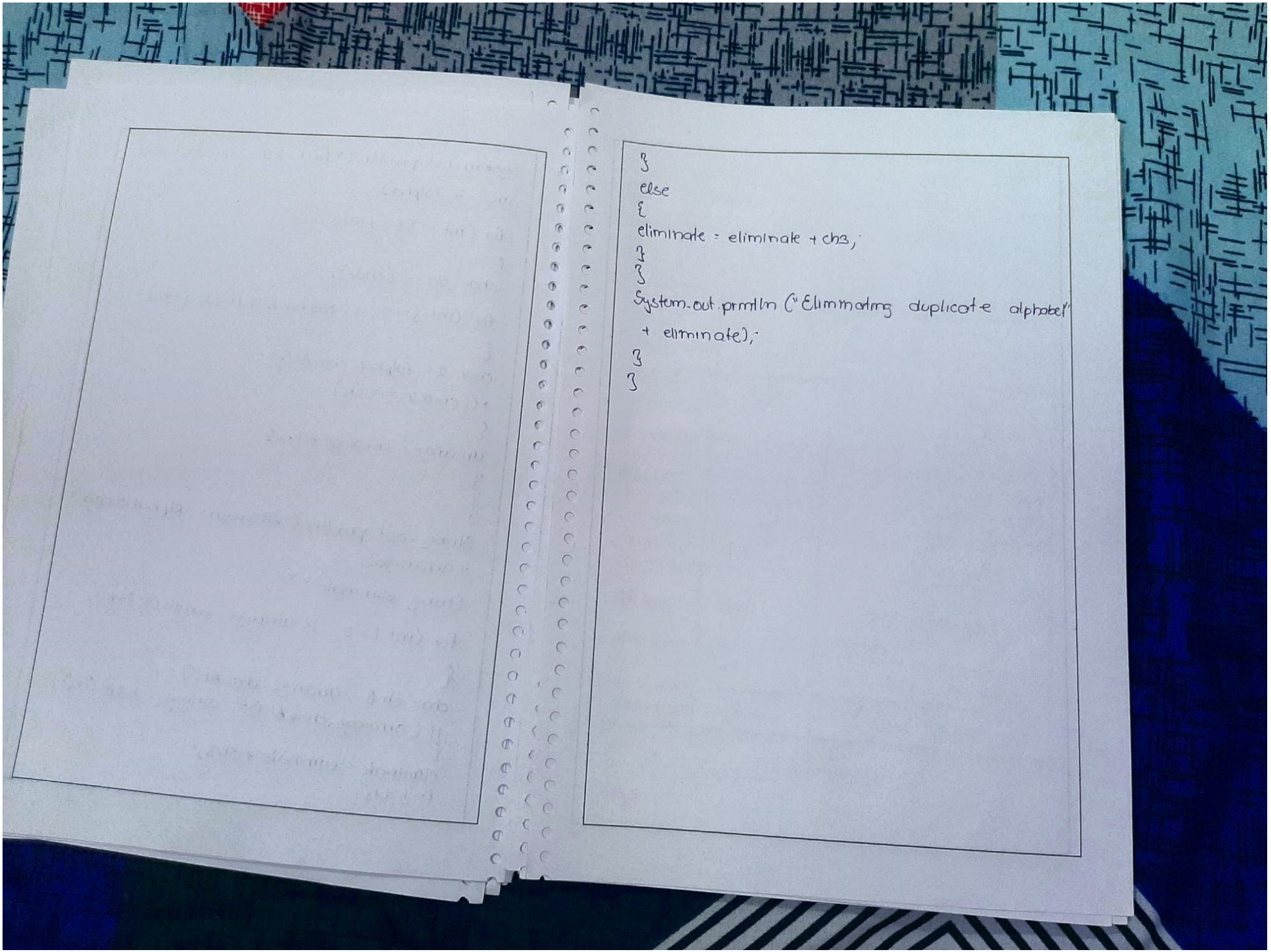
Outpd:
NMI TPMOI

The word in alphabetical order is IIMNOPT
After elimmating duplicate alphabet IMNOPT

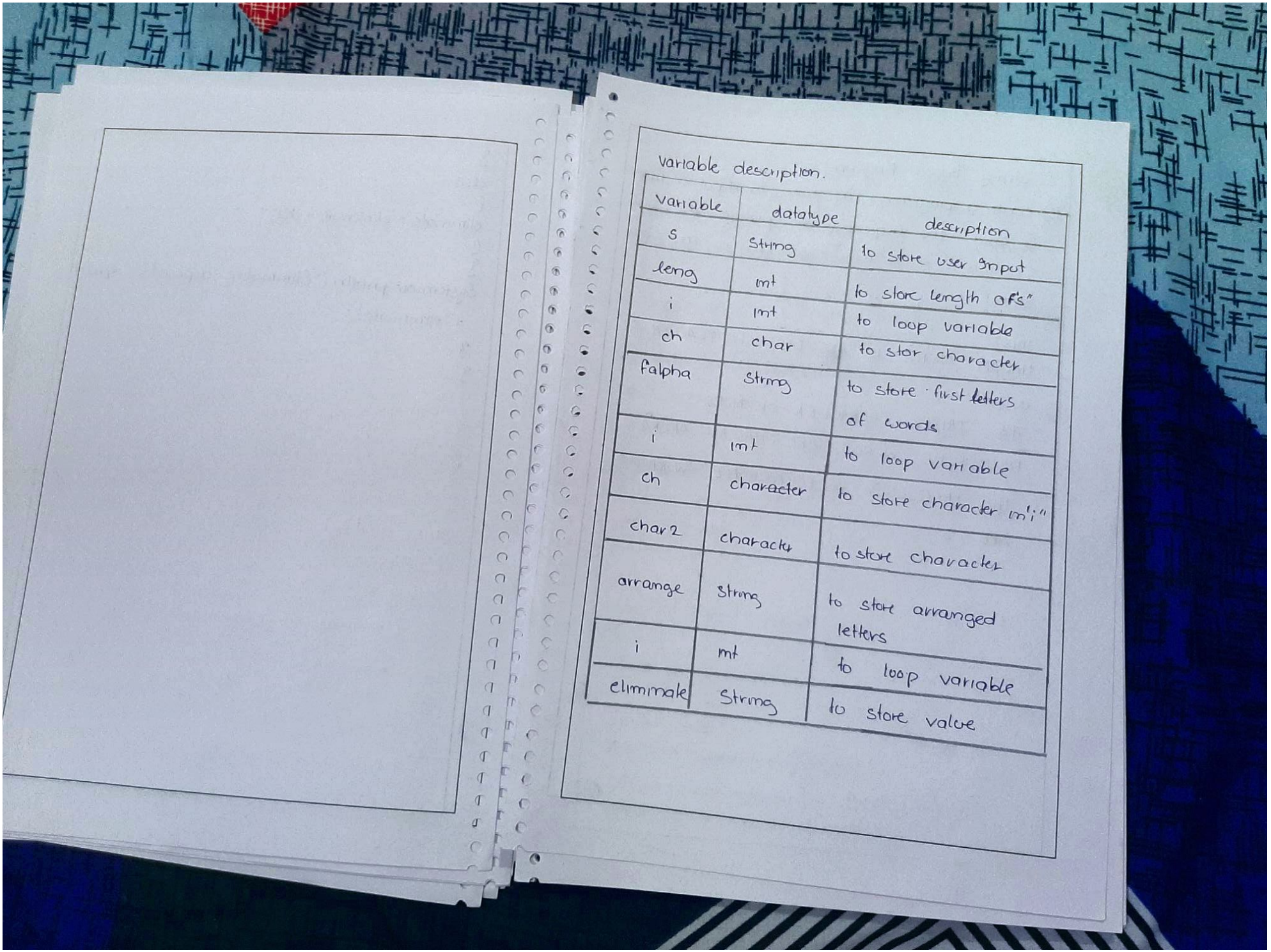
```
import java.util.*;
class Alphabet
{
public static void main(String arg[])
{
Scanner sc = new Scanner (System.in);
System.out.println("Enter any sentence in
uppercase");
String s = sc.nextLine();
s = s.toUpperCase();
s = " " + s;
String falpha = "";
String arrange = "";
int leng = s.length();
for(int i=0; i<leng; i++)
{
char ch = s.charAt(i);
if (ch == ' ')
{
falpha = falpha + s.charAt(i+1);
}
}
```



```
System.out.println("First letter of each word
are" + falpha);
for (int i=65; i<=90; i++)
{
char ch = (char)i;
for (int j=0; j< falpha.length(); j++)
{
char 2 = falpha.charAt(j);
if(char2 == ch)
{
arrange = arrange + char2;
}}}
String.out.println("Arranging Alphabetically"
+ arrange);
String eliminate = "";
for (int i=0; i< arrange.length(); i++)
{
char ch3 = arrange.charAt(i);
if (arrange.charAt(i) == arrange.charAt(i+1))
{
eliminate = eliminate + ch3;
i = i + 1;
```



```
}
else
{
eliminate = eliminate + ch3;
}
}
System.out.println("Eliminating duplicate alphabet"
+ eliminate);
}
}
```



variable description.

variable	datatype	description
s	string	to store user input
leng	int	to store length of "s"
i	int	to loop variable
ch	char	to stor character
falpha	string	to store first letters of words
i	int	to loop variable
ch	character	to store character in "s"
char2	character	to store character
arrange	string	to store arranged letters
i	int	to loop variable
elimmate	string	to store value

String Based Program

3) write a program in java to input a string & check the frequency of palindrome words (containing 3 or more characters) in the given string.

input:

MADAM ARORA IS AN ENGLISH TEACHER

output

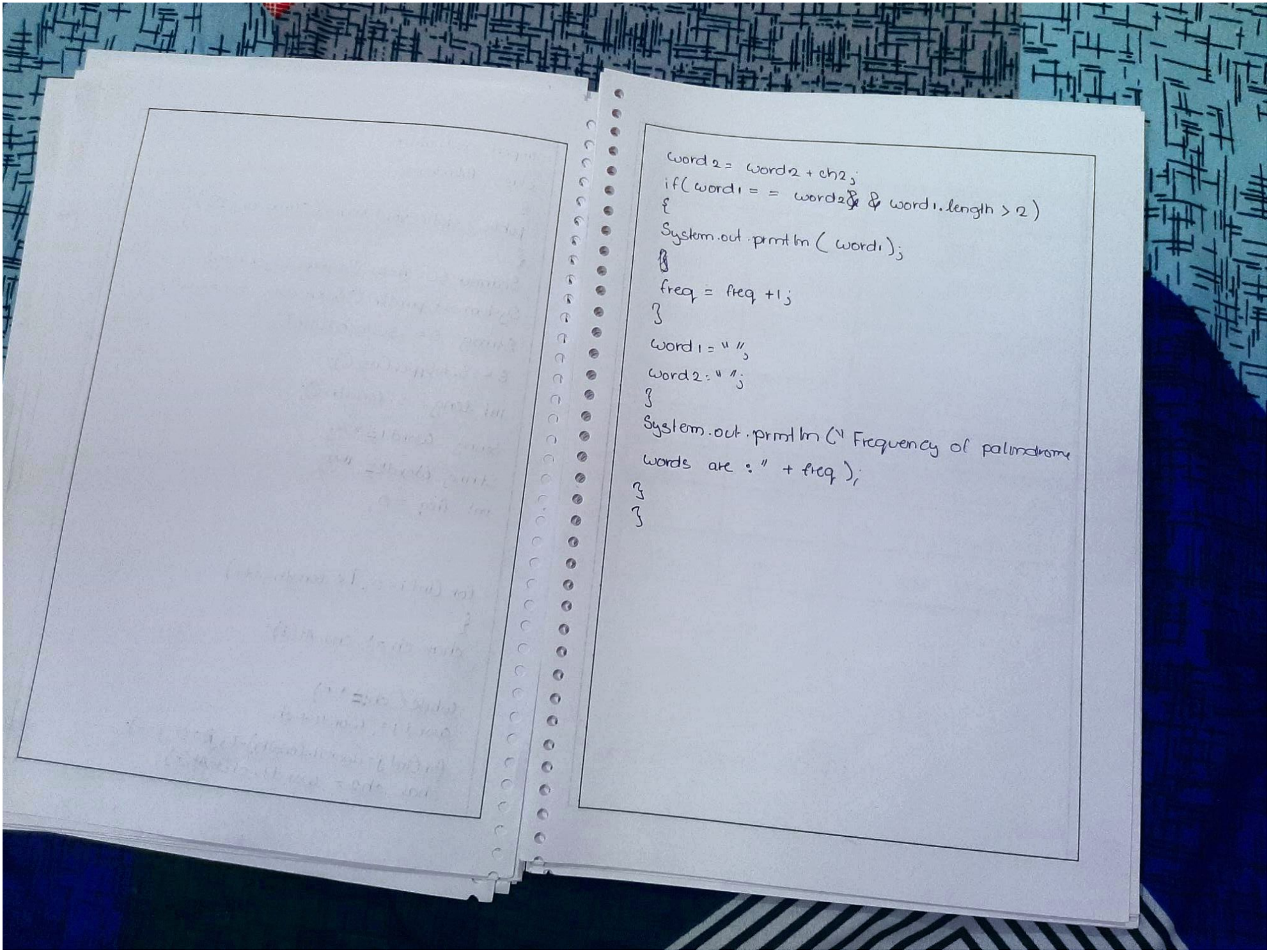
THE THREE CHARACTER OR MORE PALINDROME WORDS ARE: MADAM ARORA

THE FREQUENCY OF PALINDROME WORDS ARE: 2

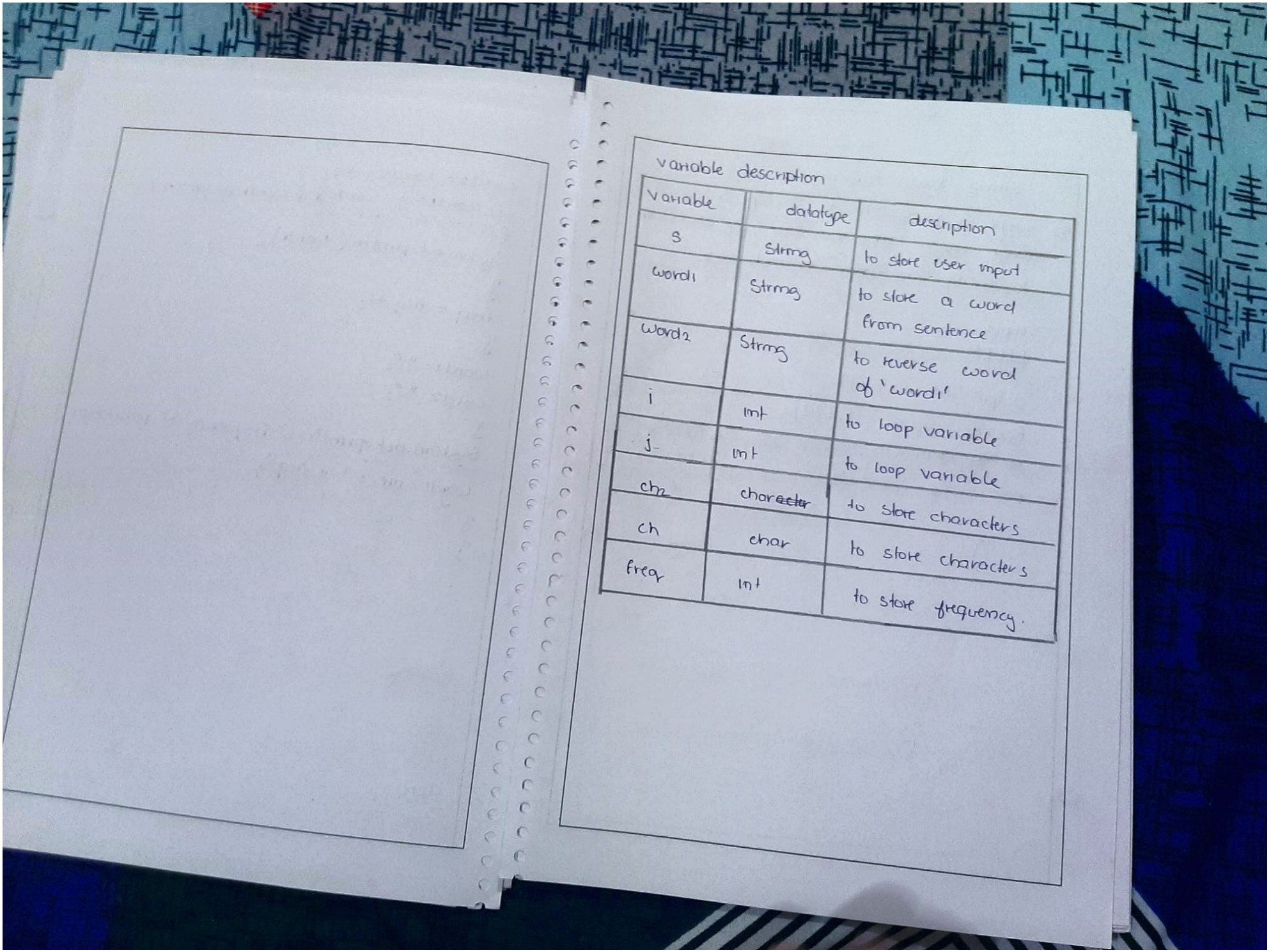
```
import java.util.*;
class Palindrome
{
public static void main(String arg[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter any Sentence")
String S= sc.nextLine();
S = S.toUpperCase();
int leng = S.length();
String word1="";
String word2="";
int freq = 0;


for(int i=0; i< length; i++)
{
char ch = S.charAt(i);

while(ch!=' ')
word1 = word1 + ch;
for(int j=(word1.length)-1; j>=0; j--)
char ch2 = word1.charAt(j);
```



```
word2 = word2 + ch2;
if(word1 == word2 && word1.length > 2)
{
System.out.println(word1);
}
freq = freq + 1;
}
word1 = "";
word2 = "";
}
System.out.println("Frequency of palindrome
words are :" + freq);
}
}
```



variable description

variable	datatype	description
s	String	to store user input
word1	String	to store a word from sentence
word2	String	to reverse word of 'word1'
i	int	to loop variable
j	int	to loop variable
ch2	character	to store characters
ch	char	to store characters
freq	int	to store frequency.

String Based Program

4) Write a program to accept a word and count the number of character to make the word palindrome

Input

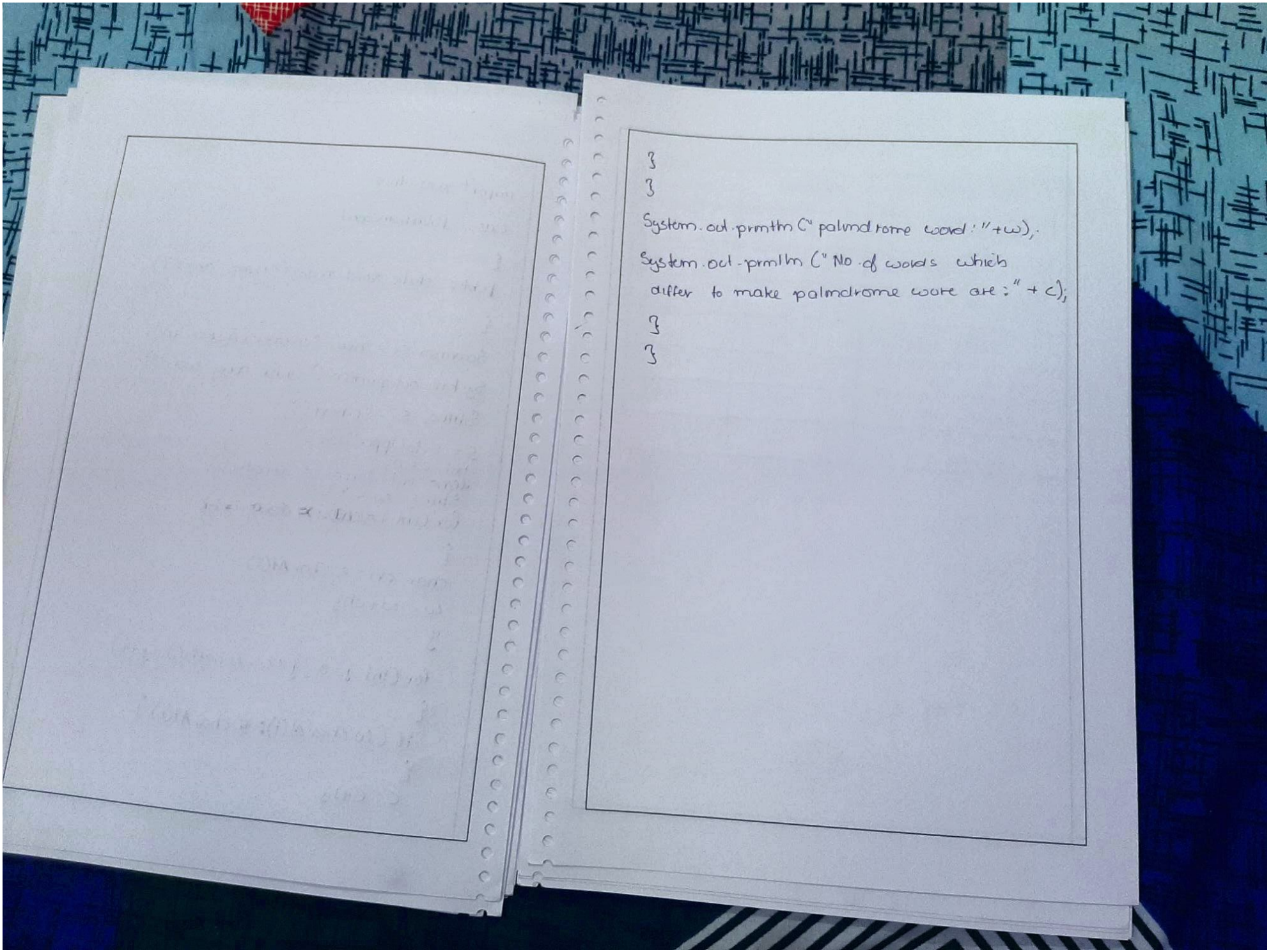
DILIP

Output

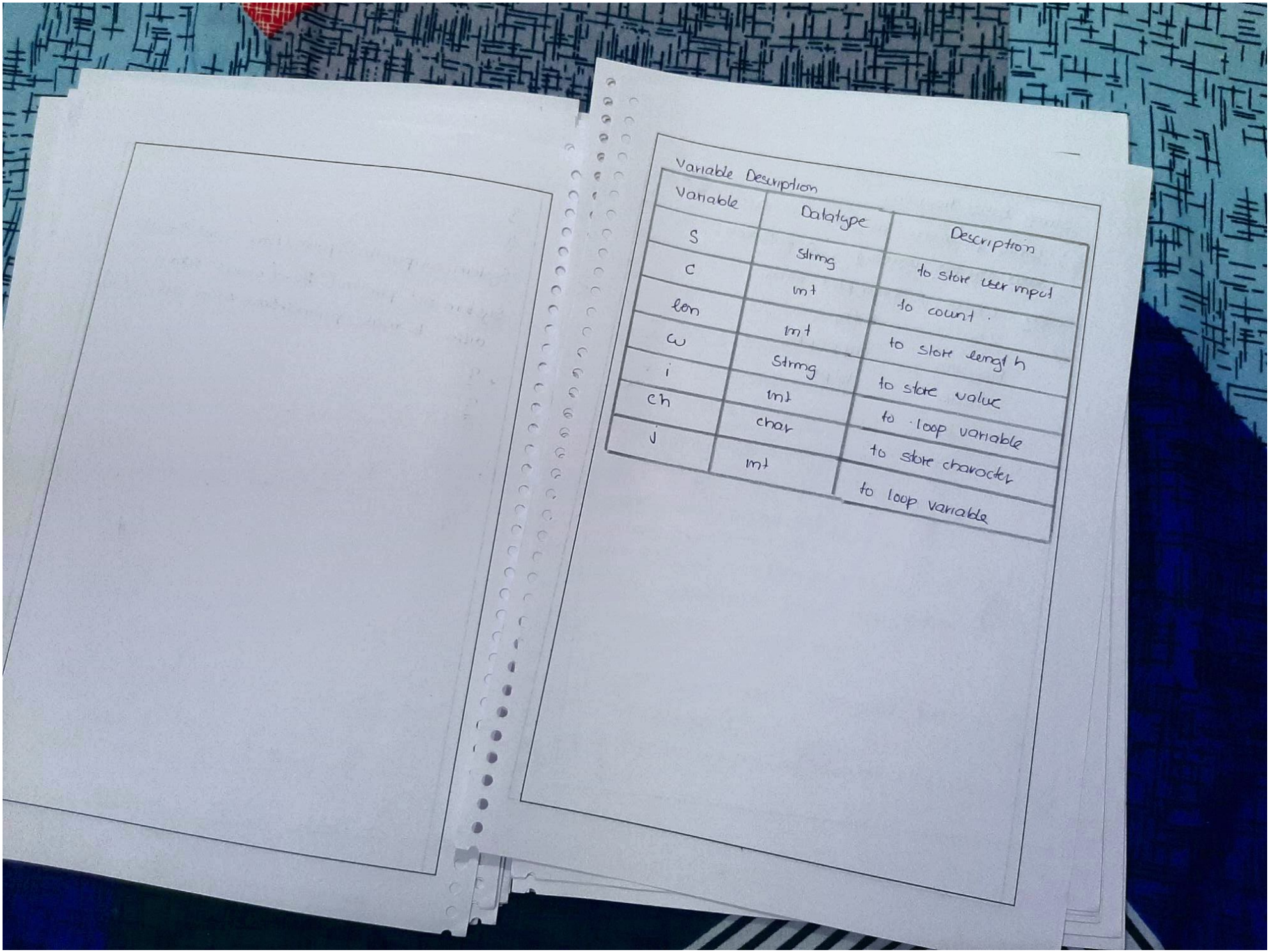
Reversed word: PILID

no of character which differ to make palindrome word: 2

```
import java.util.*;
class PalindromeWord
{
public static void main(String arg[])
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter any word");
String s = sc.next();
S = s.toUpperCase();
int c = 0;
int len = s.length();
String w = "";
for(int i=(len-1); i>=0; i=i-1)
{
char ch = s.charAt(i);
w = w+ch;
}
for(int j=0; j<w.length(); j++)
{
if(w.charAt(i)!=s.charAt(i))
{
c = c+1;
```



```
}
}
System.out.println("palindrome word:"+w);
System.out.println("No of words which
differ to make palindrome word are:" + c);
}
}
```


Variable Description
Variable | Datatype | Description
S | string | to store user input
c | int | to count
len | int | to store length
w | string | to store value
i | int | to loop variable
ch | char | to store character
j | int | to loop variable

Strmg Based Program

5) write program m java to accept a word and encrypt a follows:

eg word	encrypted form
Zeal	x fb,
Program	nppcpbk
anode	blpbf

```
import java.util.*;
class encryption
{
public static void mam (strmg arg[])
{
Scanner sc: new Scanner (System.m),
System.out.prmtln ("Enter word"),
Strmg s: sc.next();
$ Int leng = s.length();
  Strmg nencrypt = "";
for (mt i=0; i<leng; i++)
{
char ch = s.charAt(i);
char main = ((int)ch)+1;
//nencrypt = nencrypt + mam;
char actualmain = (char)main;
nencrypt = actualmain;
}
System.out.prmtln ("Encrypted form is:"),
//+(nencrypt);
Sy stem.out.prmtln("word" + "\t" + "encrypted");
}
```

```
System.out.println(s + "\t" + mencrypt);
}
}
```

Variable description

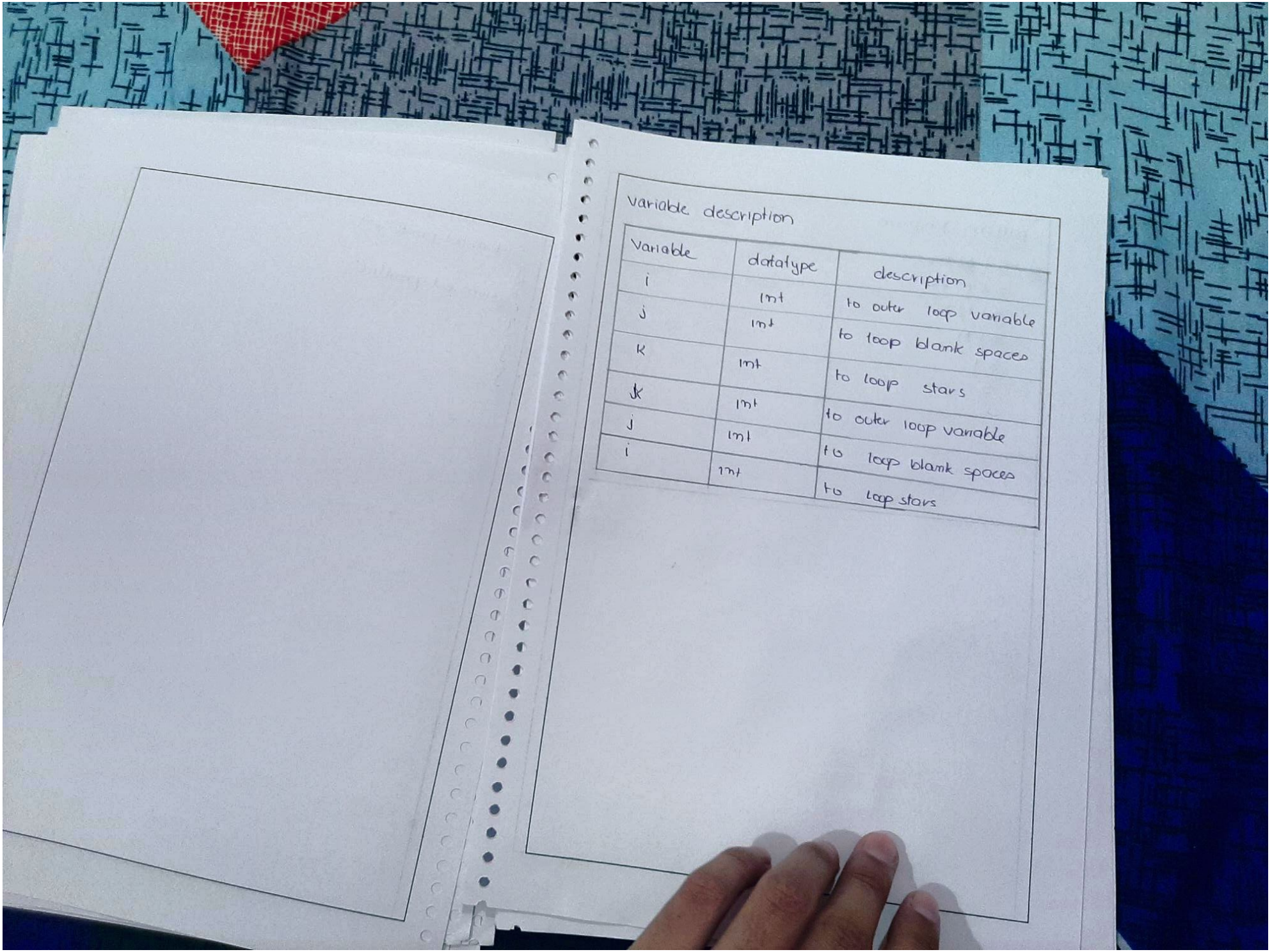
Variable	datatype	description
S	String	to stor user input
leng	int	to calculate length
noncrypt	Strng	to store value
i	int	to loop variable
ch	char	to store character
main	char	to store character
actualmain	char	to store character

Pattern Based Program

1)

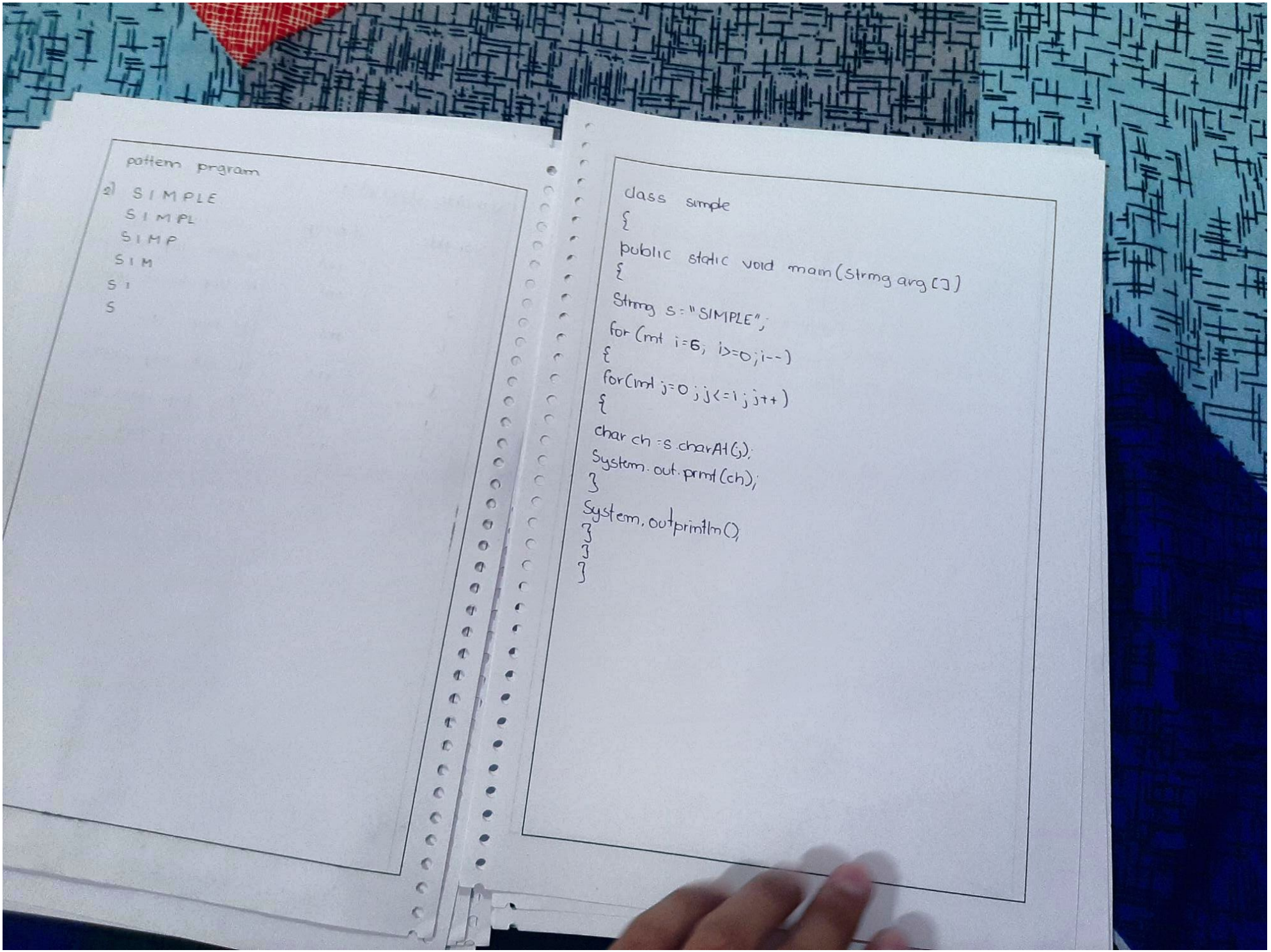
```
class patternstar
{
public static void main(String arg[])
{
for(int i=1; i<=5; i++)
{
for(int j=i; j<=4; j++)
{
System.out.print(" ");
}
for(int k=1; k<=i; k++)
{
System.out.print("*");
}
System.out.println();
}
for(int k=1; k<=4; k++)
{
for(int j=1; j<=k; j++)
{
System.out.print(" ");
}
for(int i=4; i>=k; i++)
{
```

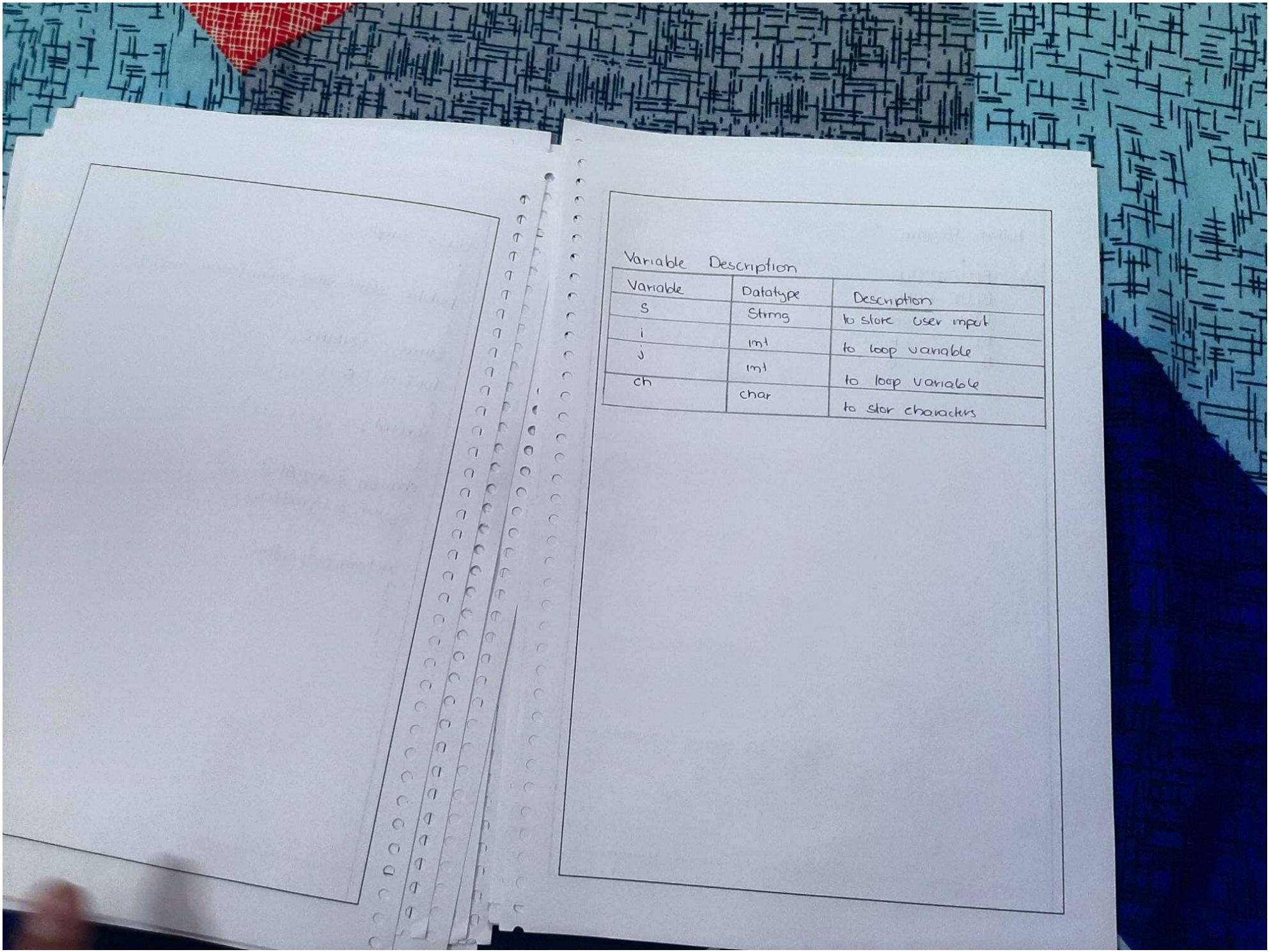
```
        System.out.print("*");
    }
    System.out.println();
}
}
```



Variable description

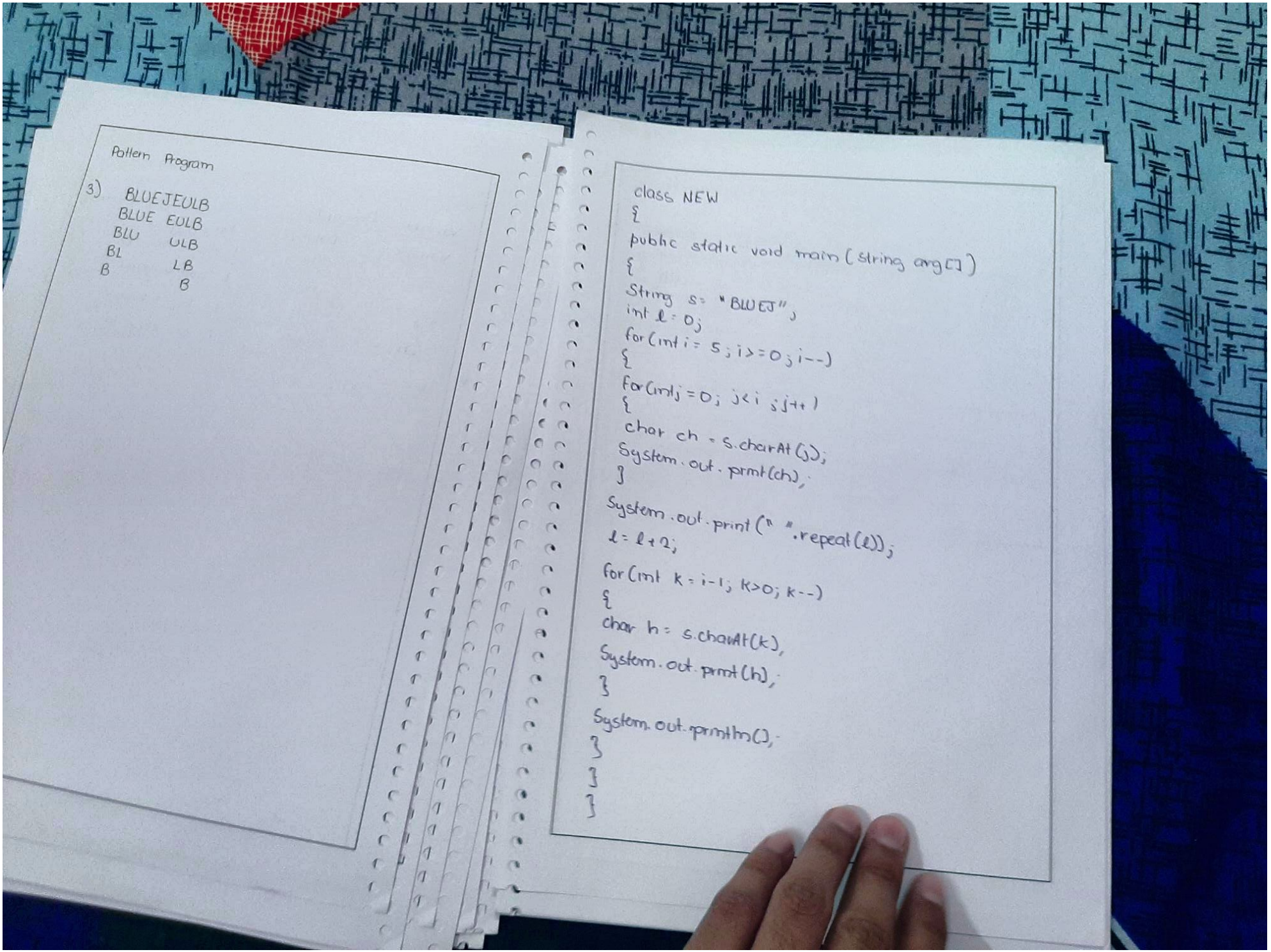
Variable	datatype	description
i	int	to outer loop variable
j	int	to loop blank spaces
k	int	to loop stars
jk	int	to outer loop variable
j	int	to loop blank spaces
i	int	to loop stars

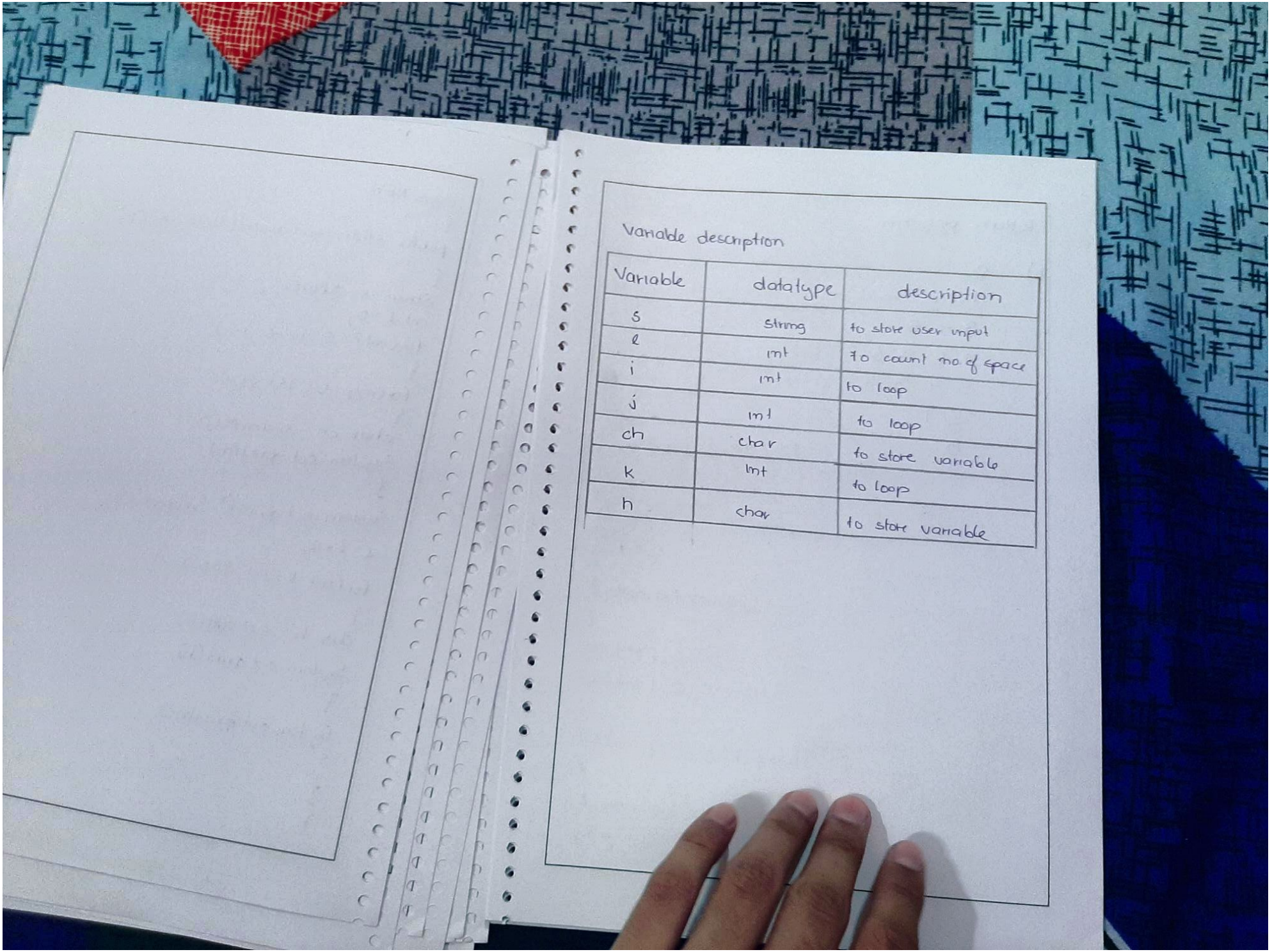

pattern program
2) SIMPLE
SIMPL
SIMP
SIM
SI
S
class simple
{
public static void main(String arg[])
{
String s="SIMPLE";
for(int i=6; i>=0;i--)
{
for(int j=0;j<=i;j++)
{
char ch=s.charAt(j);
System.out.print(ch);
}
System.out.println();
}
}



Variable Description

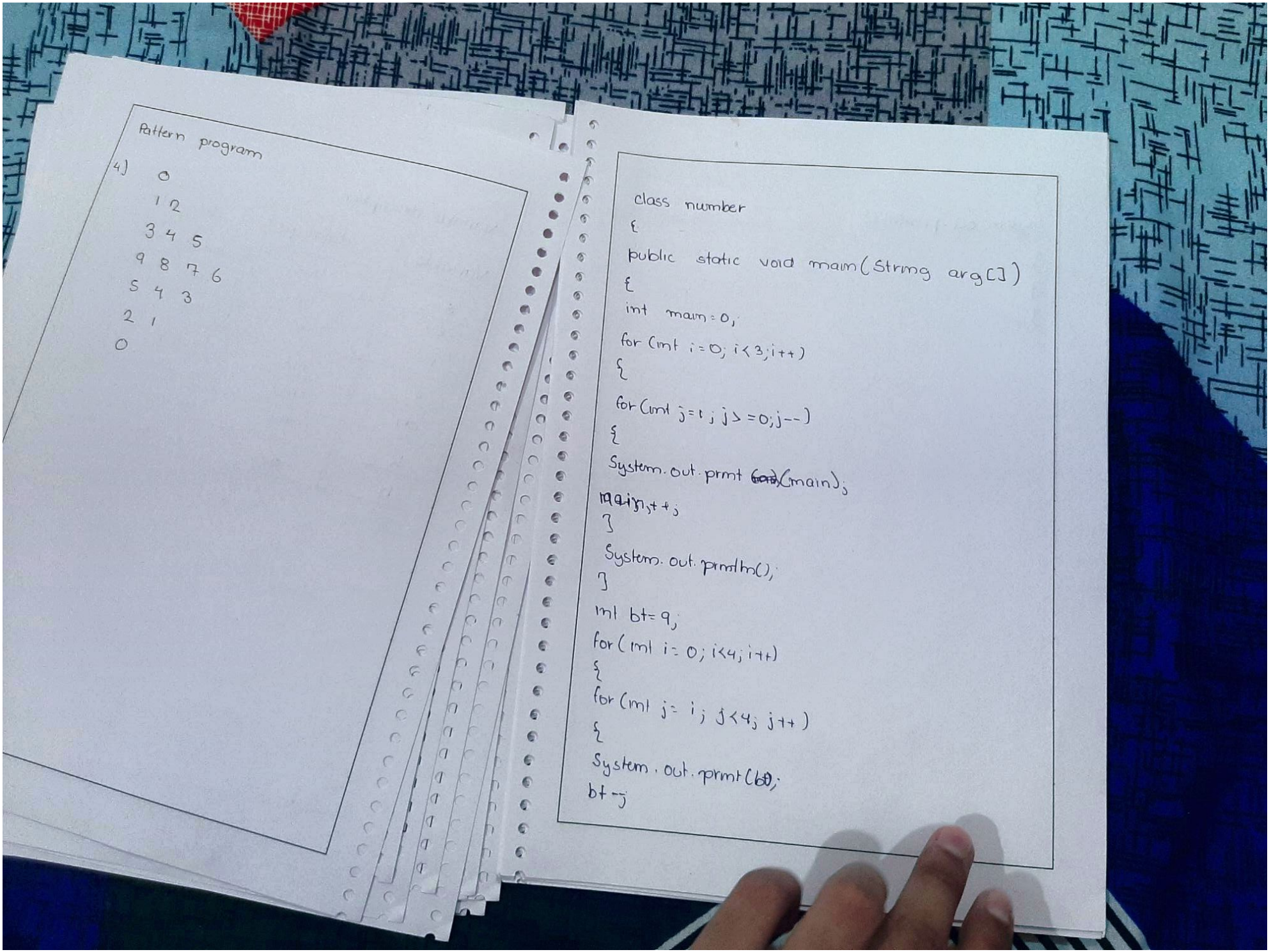
Variable	Datatype	Description
s	Strng	to store user input
i	int	to loop variable
j	int	to loop variable
ch	char	to stor characters


Pattern Program
3) BLUEJEULB
BLUE EULB
BLU ULB
BL LB
B B
class NEW
{
public static void main (String arg[])
{
String s= "BLUEJ";
int l = 0;
for(int i = 5; i>=0; i--)
{
for(int j = 0; j<i ; j++)
{
char ch = s.charAt(j);
System.out.print(ch);
}
System.out.print(" ".repeat(l));
l = l+2;
for(int k = i-1; k>0; k--)
{
char h = s.charAt(k);
System.out.print(h);
}
System.out.println();
}
}



Variable description

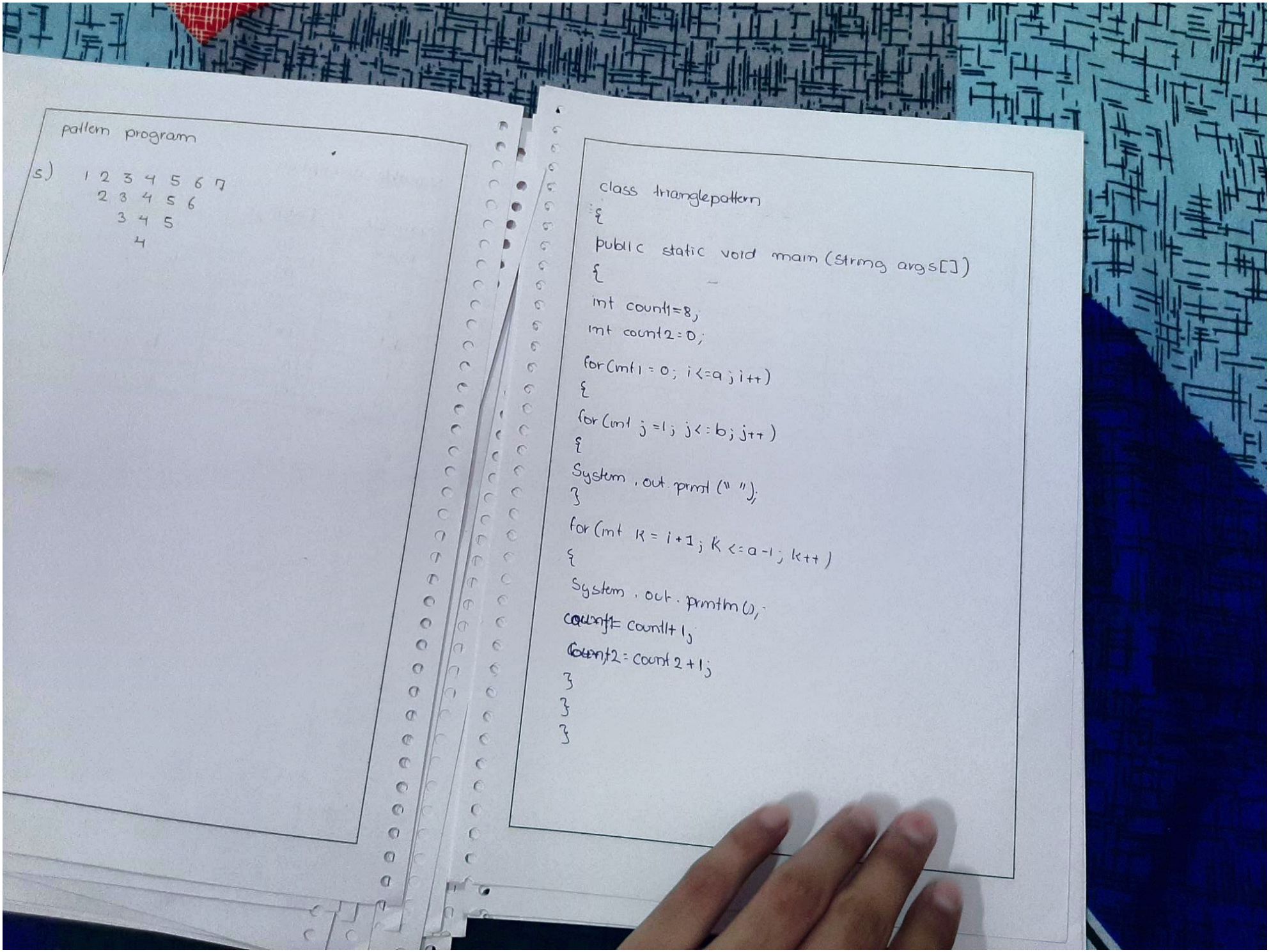
Variable	datatype	description
s	string	to store user input
l	int	to count no of space
i	int	to loop
j	int	to loop
ch	char	to store variable
k	int	to loop
h	char	to store variable


Pattern program
4)
0
1 2
3 4 5
9 8 7 6
5 4 3
2 1
0
class number
{
public static void main(String arg[])
{
int main=0;
for(int i=0; i<3;i++)
{
for(int j=i; j>=0;j--)
{
System.out.print(main);
main,j+j
}
System.out.println();
}
int bt=9;
for(int i=0; i<4;i++)
{
for(int j= i; j<4; j++)
{
System.out.print(bt);
bt-j

mth();
}
}
System.out.println();

Variable description.

Variable	Datatype	Description
main	int	to count
i	int	to loop
j	int	to loop
bt	int	to count
i	int	to count loop
j	int	to loop

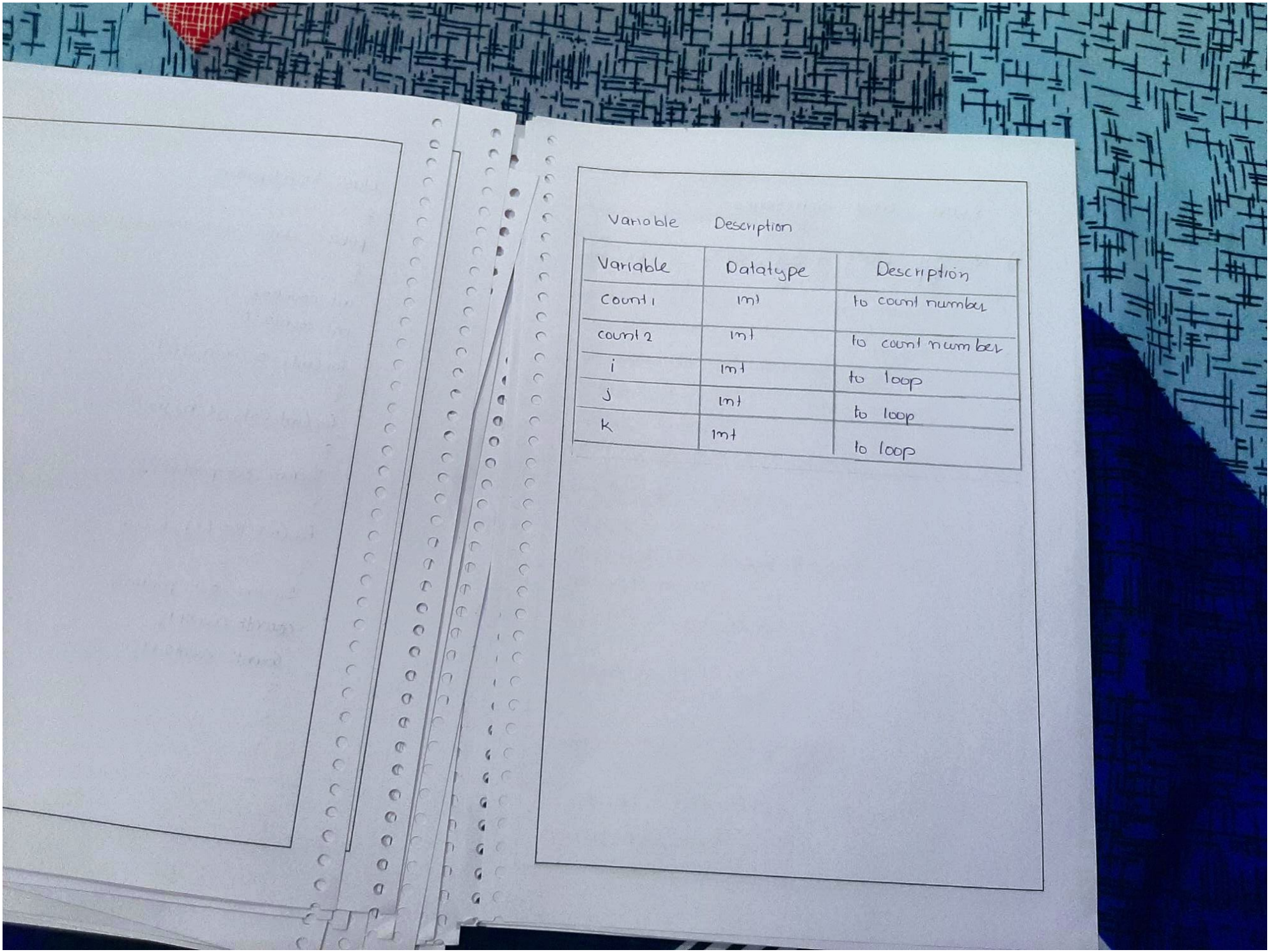


pattern program

s)
```
1 2 3 4 5 6 7
  2 3 4 5 6
    3 4 5
      4
```

```
class trianglepattern
{
public static void main (String args[])
{
int count1=8;
int count2=0;

for(int i = 0; i<=a; i++)
{
for(int j =1; j<=b; j++)
{
System.out.print(" ");
}
for(int k = i+1; k<=a-1; k++)
{
System.out.println();
count1= count1+1;
count2 = count2+1;
}
}
}
```



Variable Description

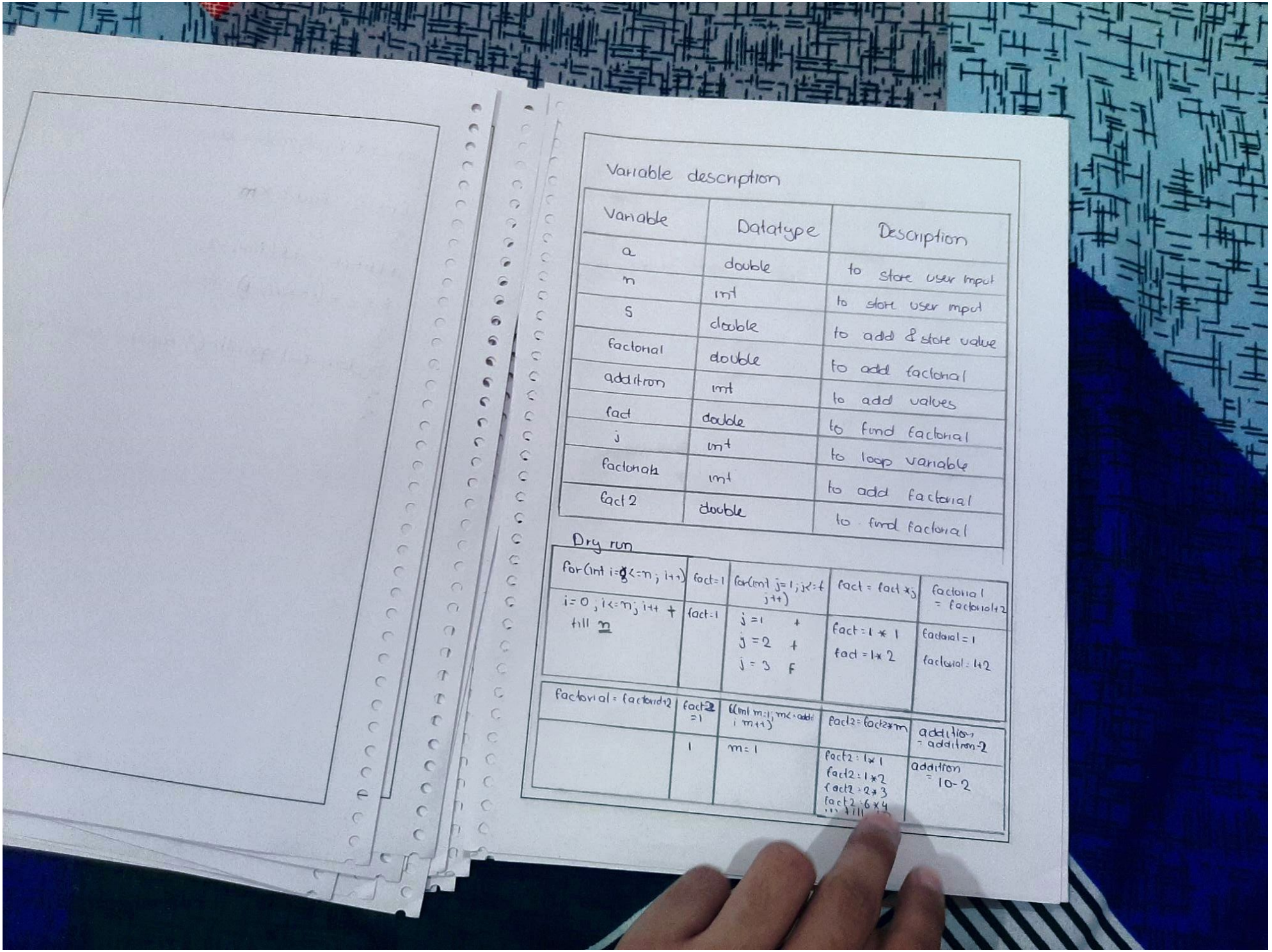
Variable	Datatype	Description
count1	int	to count number
count2	int	to count number
i	int	to loop
j	int	to loop
k	int	to loop

Series and Sum Series

1) $S = \frac{-a}{2!} + 10! + \frac{-a}{4!} + 9! + \frac{-a}{6!} + 8! - \text{nterm}$

```
import java.util.*;
class factorialseries
{
public static void main(String arg[])
{
Scanner sc= new Scanner(System.in);
System.out.println("Enter a value");
double a= sc.nextInt();
System.out.println("Enter n value");
int n= sc.nextInt();
double S = 0.0;
factorial = 0 double factorial = 20;
int addition =10;
for(int i =0 ; i<= n ; i++)
{double fact = 1;
for(int j =0; j< factorial; j++)
{
double fact = fact * j;
}
factorial = factorial+2;
double fact 2=1;
```

```
for (int m = 1; m <= addition; m ++)
{
    fact 2 = fact 2 * m;
}
addition = addition - 2;
s = s + (r-a)/fact2 + fact2;
}
System.out.println("required series :" + s);
}
}
```



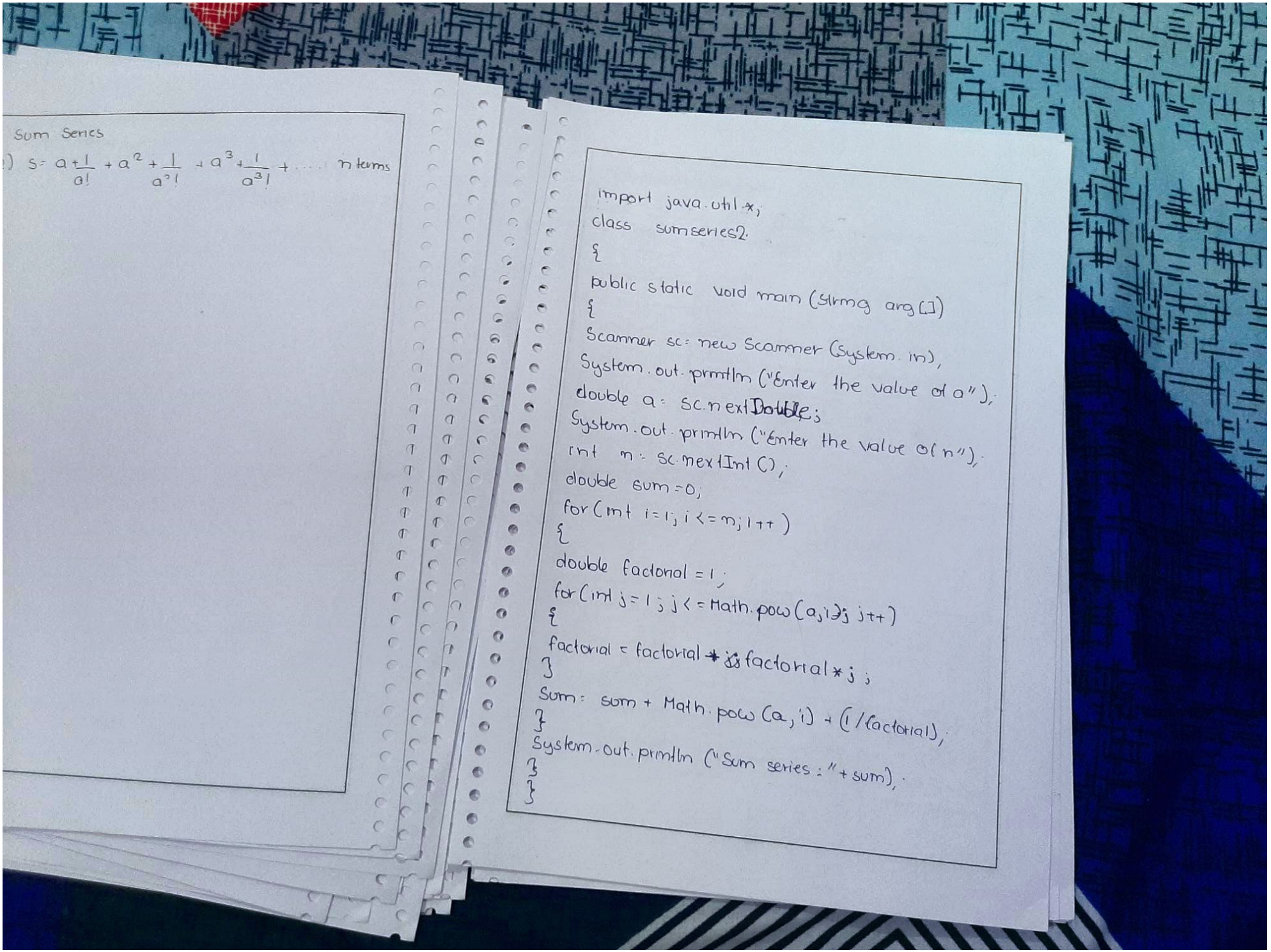
Variable description

Variable	Datatype	Description
a	double	to store user input
n	int	to store user input
s	double	to add & store value
factorial	double	to add factorial
addition	int	to add values
fact	double	to find factorial
j	int	to loop variable
factorial2	int	to add factorial
fact2	double	to find factorial

Dry run

for(int i=0; i<=n; i++)	fact=1	for(int j=1; j<=i; j++)	fact = fact * j;	factorial = factorial+2
i=0, i<=n; i++ till n	fact=1	j=1 t j=2 t j=3 f	fact = 1 * 1 fact = 1 * 2	factorial = 1 factorial = 1+2

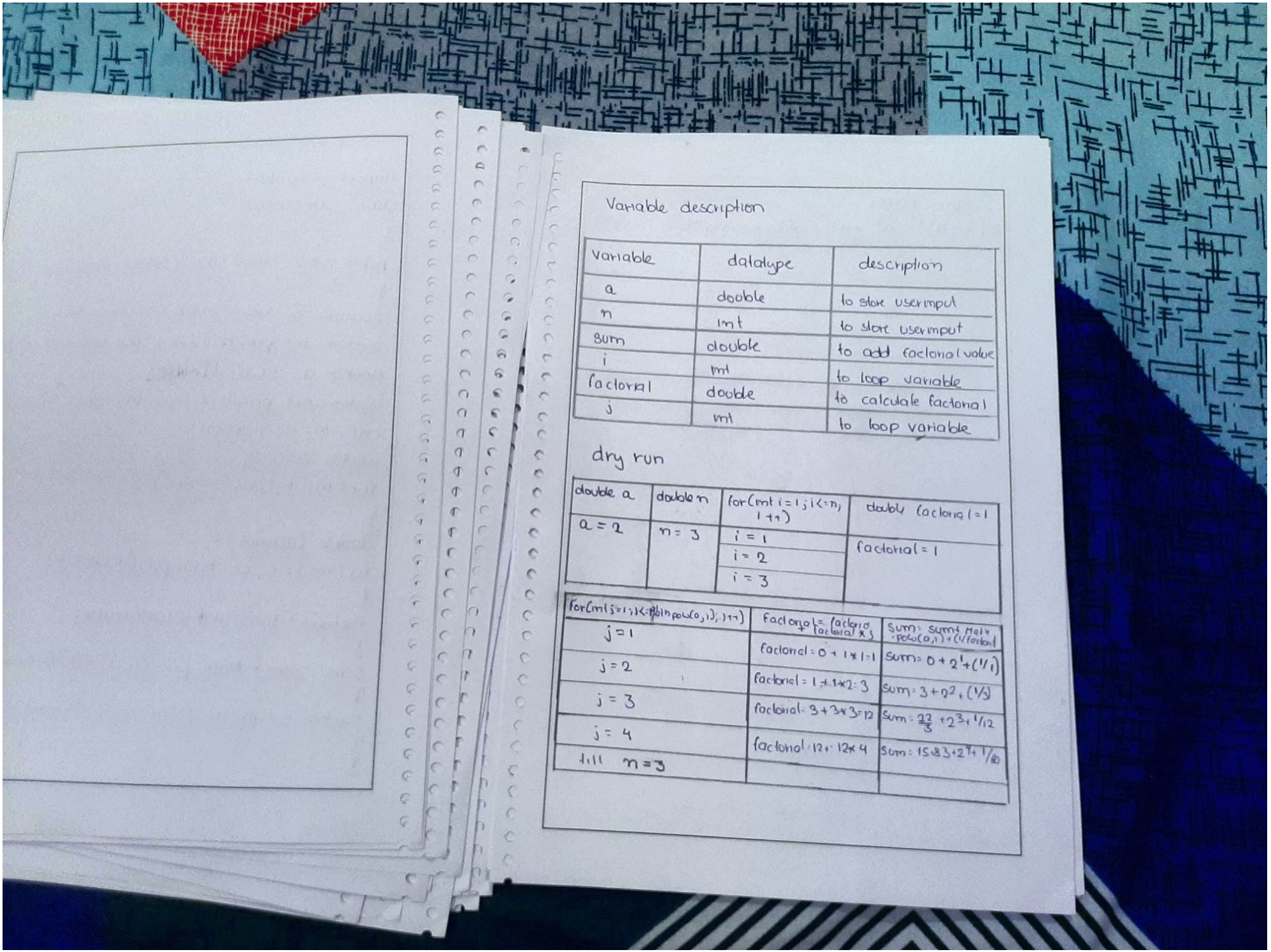
factorial = factorial+2	fact2 = 1	for(int m=1; m<=addition; m++)	fact2 = fact2 * m	addition = addition-2
	1	m=1	fact2 = 1 * 1 fact2 = 1 * 2 fact2 = 2 * 3 fact2 = 6 * 4 ... till ...	addition = 10-2


Sum Series
) S= a+1/a! +a^2+1/a^2! +a^3+1/a^3! + ... n terms
import java.util.*;
class sumseries2.
{
public static void main (String arg[])
{
Scanner sc= new Scanner (System.in);
System.out.println ("Enter the value of a");
double a= sc.nextDouble;
System.out.println ("Enter the value of n");
int n= sc.nextInt();
double sum=0;
for(int i=1; i<=n; i++)
{
double factorial =1;
for(int j=1; j<=Math.pow(a,i); j++)
{
factorial = factorial * j; factorial*j;
}
sum= sum + Math.pow(a,i) + (1/factorial);
}
System.out.println ("Sum series :" + sum);
}
}

Sum Series

2) $S = a + \frac{1}{a!} + a^2 + \frac{1}{a^2!} + a^3 + \frac{1}{a^3!} + \ldots$ n terms

```
import java.util.*;
class sumseries2.
{
public static void main (String arg[])
{
Scanner sc= new Scanner (System.in),
System.out.println ("Enter the value of a"),
double a= sc.nextDouble;
System.out.println ("Enter the value of n"),
int n= sc.nextInt (),
double sum=0;
for(int i=1; i<=n; i++)
{
double factorial =1;
for(int j=1; j<=Math.pow(a,i); j++)
{
factorial = factorial * factorial * j;
}
Sum= sum + Math.pow(a,i) + (1/factorial),
}
System.out.println ("Sum series :" + sum),
}
}
```



Variable description

Variable	datatype	description
a	double	to store user input
n	int	to store user input
sum	double	to add factorial value
i	int	to loop variable
factorial	double	to calculate factorial
j	int	to loop variable

dry run

double a	double n	for(int i=1;i<=n; i++)	double factorial=1
a=2	n=3	i=1	factorial=1
		i=2	
		i=3	

for(int j=1;j<=Math.pow(a,i);j++)	factorial=factorial * factorial*j;	sum=sum+Math.pow(a,i)+(1/factorial)
j=1	factorial=0+1*1=1	sum=0+2¹+(1/1)
j=2	factorial=1+1*2=3	sum=3+2²+(1/3)
j=3	factorial=3+3*3=12	sum=22/3+2³+1/12
j=4	factorial=12+12*4	sum=15.83+2⁴+1/48
i=11 n=3		

Sum Series

3) $1 + a^{2!} - a^{3!} + a^{4!} - a^{5!} \ldots + a^{n!}$

```
import java.util.*;
class  prog3sumseries
{
public static void main(String arg[])
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter a value");
double a=sc.nextDouble();
System.out.println("Enter n value");
int n=sc.nextInt();
double sum=0; fc=2;
for(int i=1;i<=n;i++)
{
double fact=1;
for(j=1;j<=fc;j++)
{
//fact=fact*fact*j;*/ fact=fact*j;
}
fc=fc+1;
if(i%2==0)
sum=sum+Math.pow(a,fact);
else
sum=sum-Math.pow(a,fact);
}
System.out.println("Sum series is:"+sum);
```

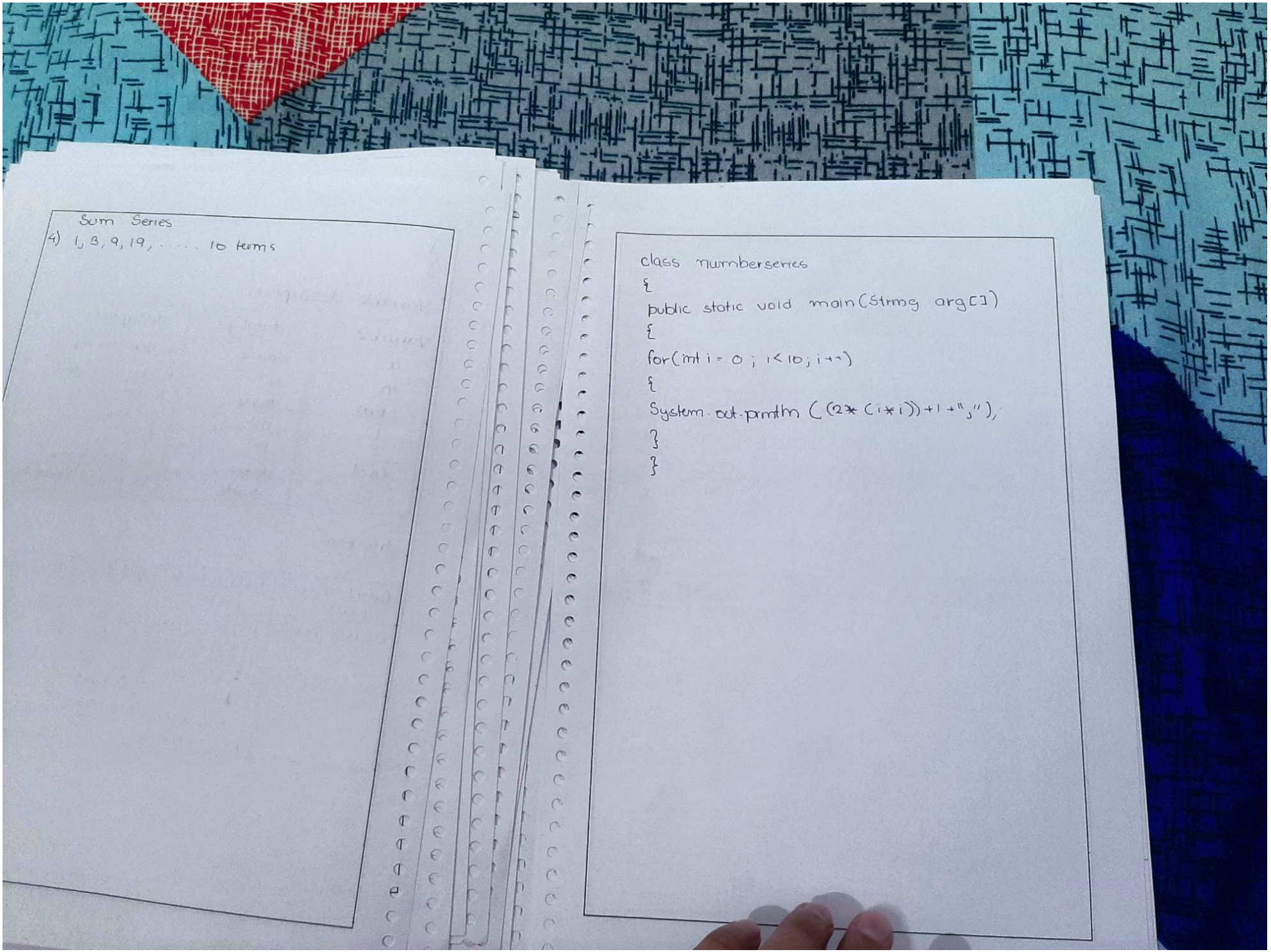
}
}

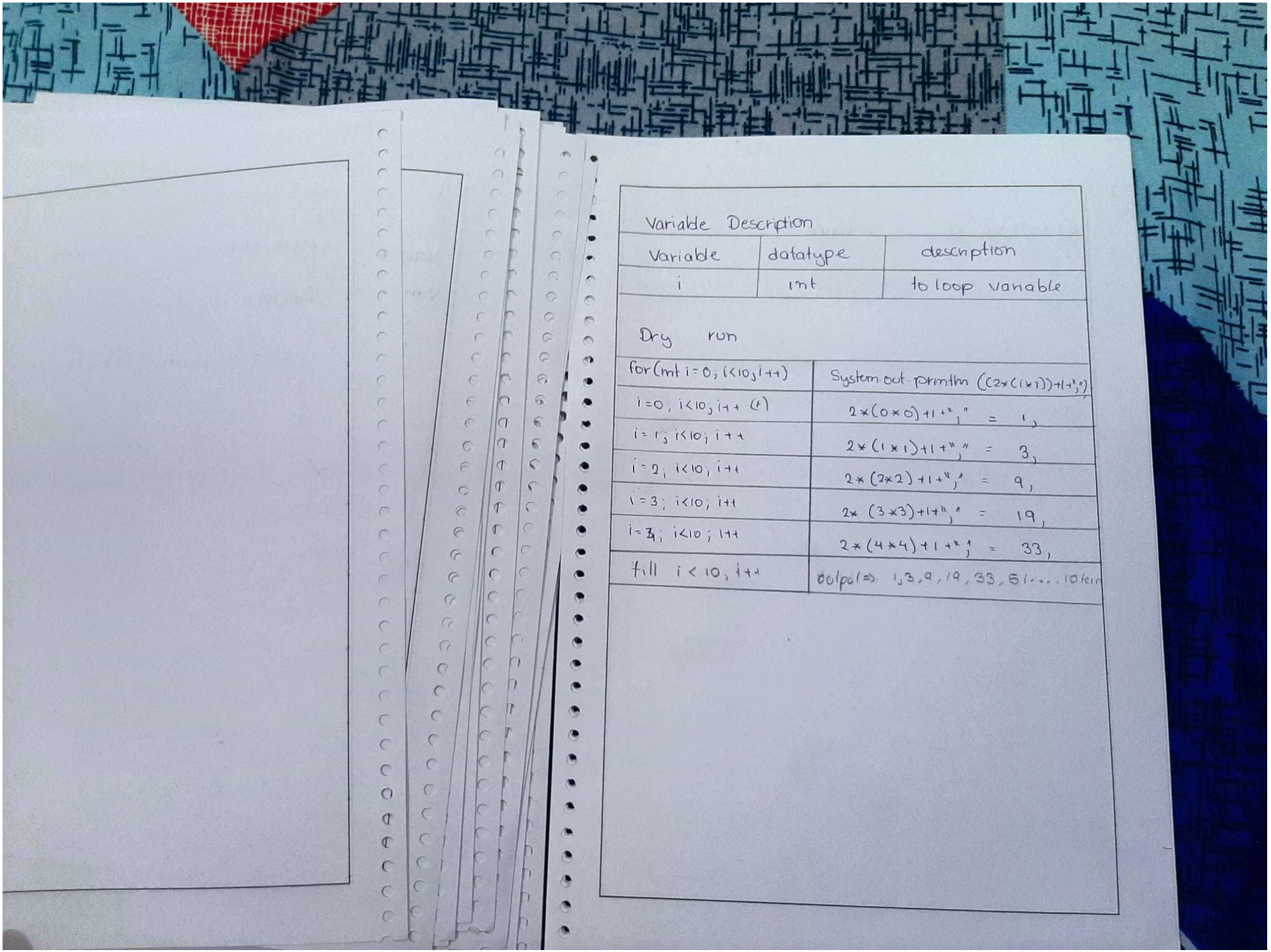
Variable description

Variable	datatype	description
a	double	to store user input
n	int	to store user input
sum	double	to add value
i	int	to loop
j	int	to loop
fact	double	to add factorial
fc	double	to increase value by 1

dry run

for(int i=1;i<=n i++)	for(j=1; j<=fc; j++)	fact=fact*j	if(i%2==0)	else
i=1; i<3; i++ (t)	j=1; j<=2; j++ (t) j=2; j<=2; j++ (t) j=3; j<=2; j++ (f)	fact=1*1 fact=1*2	i%2==0 (f) (1%2==0)	S=0+Math.pow(a,2) fact
i=2; i<3; i++ (t)	j=1; j<=3; j++ (t) j=2; j<=3; j++ (t) j=3; j<=3; j++ (t) j=4; j<=3; j++ (f)	fact=1*1 fact=1*2 fact=2*3	i%2==0 (t) 2%2==0	S=0+Math.pow(a,fact)


Sum Series
4) 1, 3, 9, 19, ... 10 terms
class numberseries
{
public static void main(String arg[])
{
for(int i = 0 ; i<10; i++)
{
System.out.println ((2*(i*i))+1+",");
}
}



Variable Description

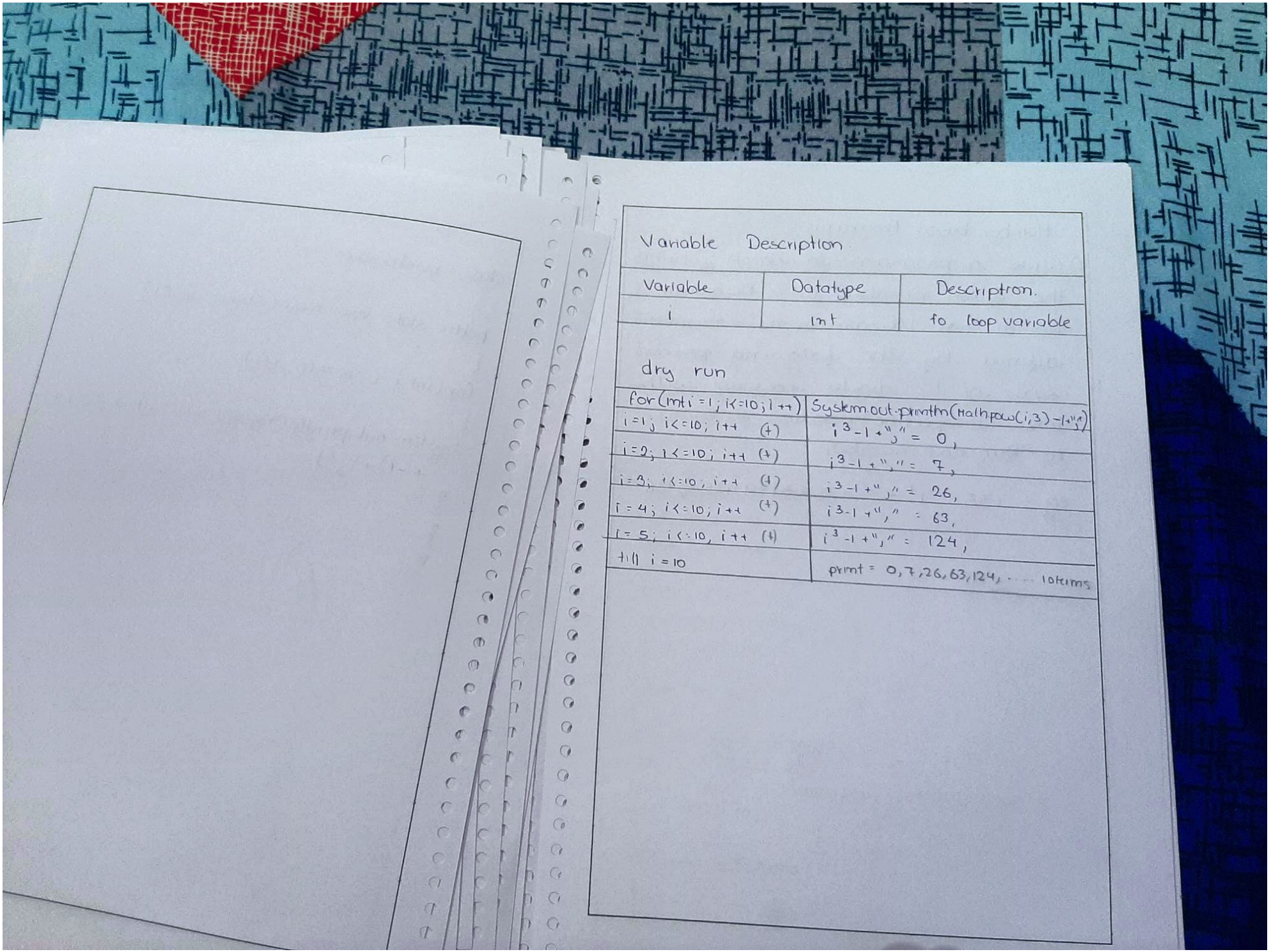
Variable	datatype	description
i	int	to loop variable

Dry run

for(int i=0; i<10; i++)	System.out.println((2*(i*i))+1+",");
i=0; i<10; i++ (1)	2*(0*0)+1+","; = 1,
i=1; i<10; i++	2*(1*1)+1+","; = 3,
i=2; i<10; i++	2*(2*2)+1+","; = 9,
i=3; i<10; i++	2*(3*3)+1+","; = 19,
i=4; i<10; i++	2*(4*4)+1+","; = 33,
till i<10, i++	output => 1,3,9,19,33,51.....10 term

5) 0, 7, 26, 63 .... 10 terms

```
class numberseries
{
public static void main(String arg[])
{
for(int i=1; i<=10; i++)
{
System.out.println("required series" +(Math.pow(i,3)
  * -1) + "j");
}
}
}
```



Variable Description

Variable	Datatype	Descriptron.
i	int	to loop variable

dry run

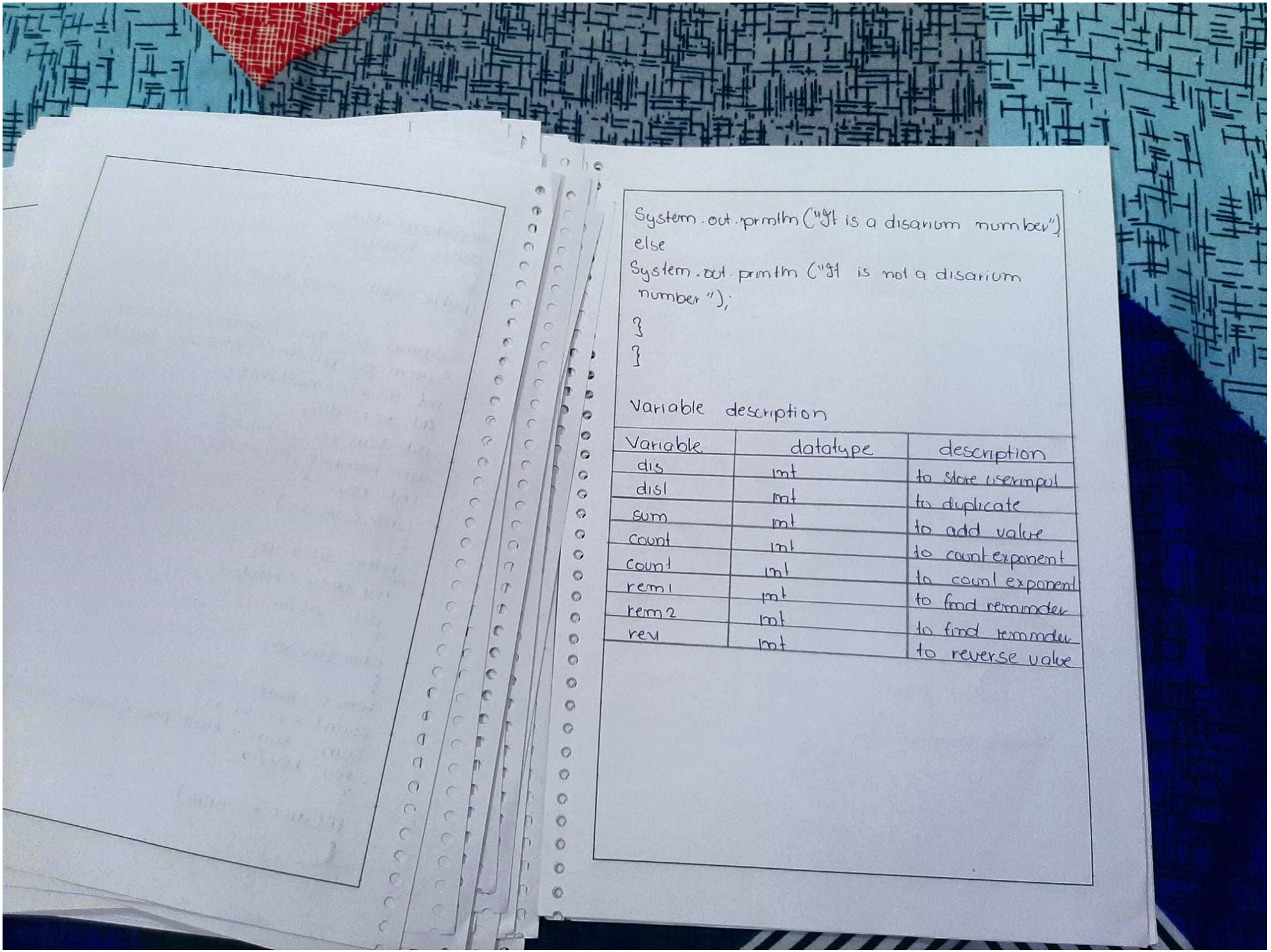
for(int i = 1; i<=10; i++)	System.out.println(Math.pow(i,3)-1+",");
i=1; i<=10; i++ (T)	i^3-1 + "," = 0,
i=2; i<=10; i++ (T)	i^3-1 + "," = 7,
i=3; i<=10; i++ (T)	i^3-1 + "," = 26,
i=4; i<=10; i++ (T)	i^3-1 + "," = 63,
i=5; i<=10, i++ (T)	i^3-1 + "," = 124,
till i = 10	print = 0,7,26,63,124, ... 10terms

Number based Programs

1) Write a program to check whether the given number is a Disarium no. or not. disarium no. = number defined by the following process sum of its digits powered with three respective position is equal to original number

eg - 135 : $1^1 + 3^2 + 5^3 \Rightarrow 1+9+125 \Rightarrow 136$

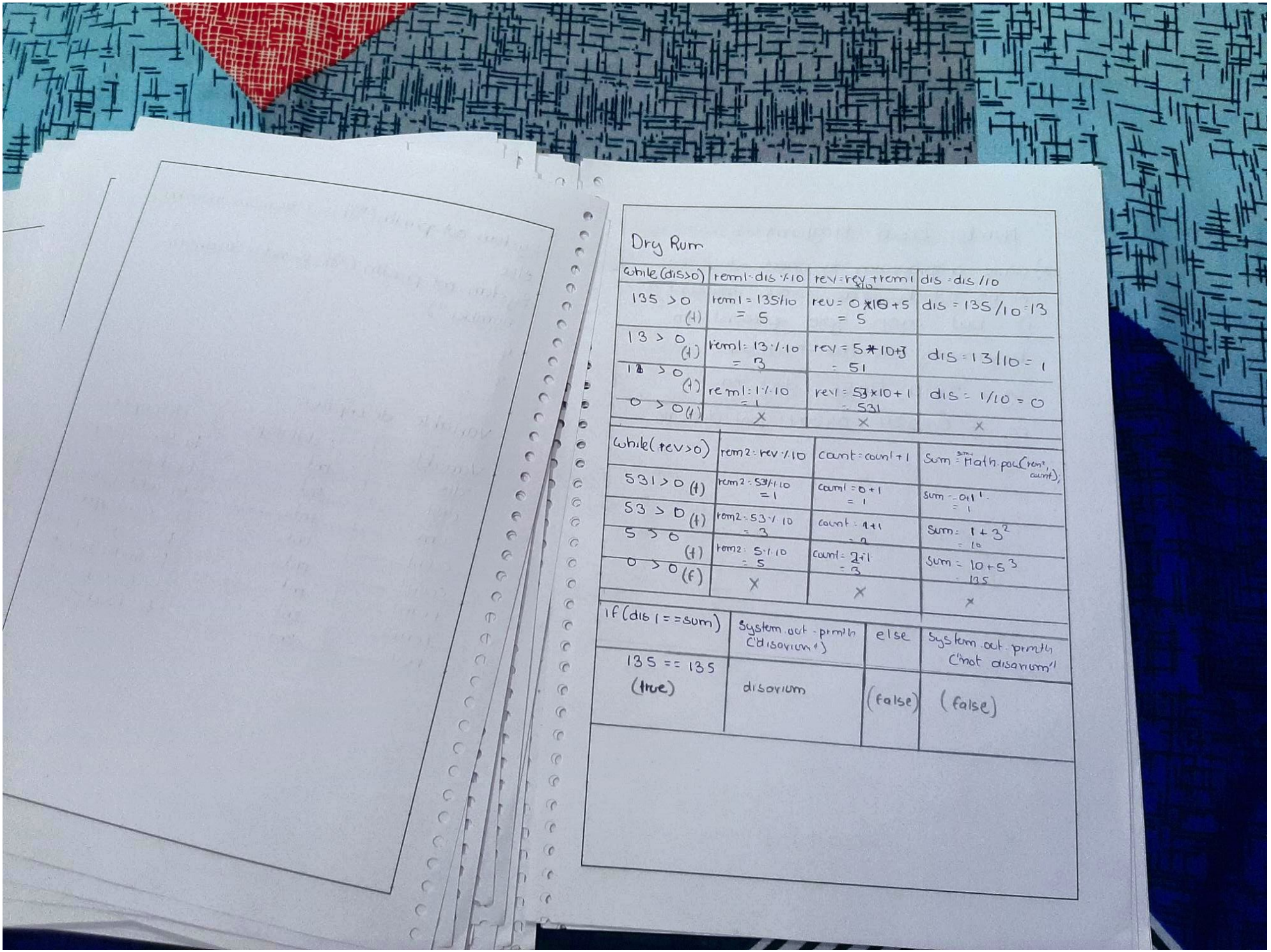
```
import java.util.*;
class disarium
{
public static void main (String arg[])
{
Scanner sc = new Scanner (System.in);
System.out.println("Enter any number");
int dis = sc.nextInt();
int dis1 = dis;
int sum = 0, count = 0;
int rem1 = 0, rem2 = 0;
int rev = 0;
while (dis >0)
{
rem1 = dis % 10;
rev = rev + rem1;
dis = dis /10;
}
while (rev >0)
{
rem2 = rev % 10;
count = count + 1;
sum = sum + Math.pow (rem2, count);
rev = rev /10;
}
if (dis1 == sum)
{
```



```
System.out.println("%d is a disarium number")
else
System.out.println("%d is not a disarium
number");
}
}
```

Variable description

Variable	datatype	description
dis	int	to store userinput
dis1	int	to duplicate
sum	int	to add value
count	int	to count exponent
count	int	to count exponent
rem1	int	to find remainder
rem2	int	to find remainder
rev	int	to reverse value

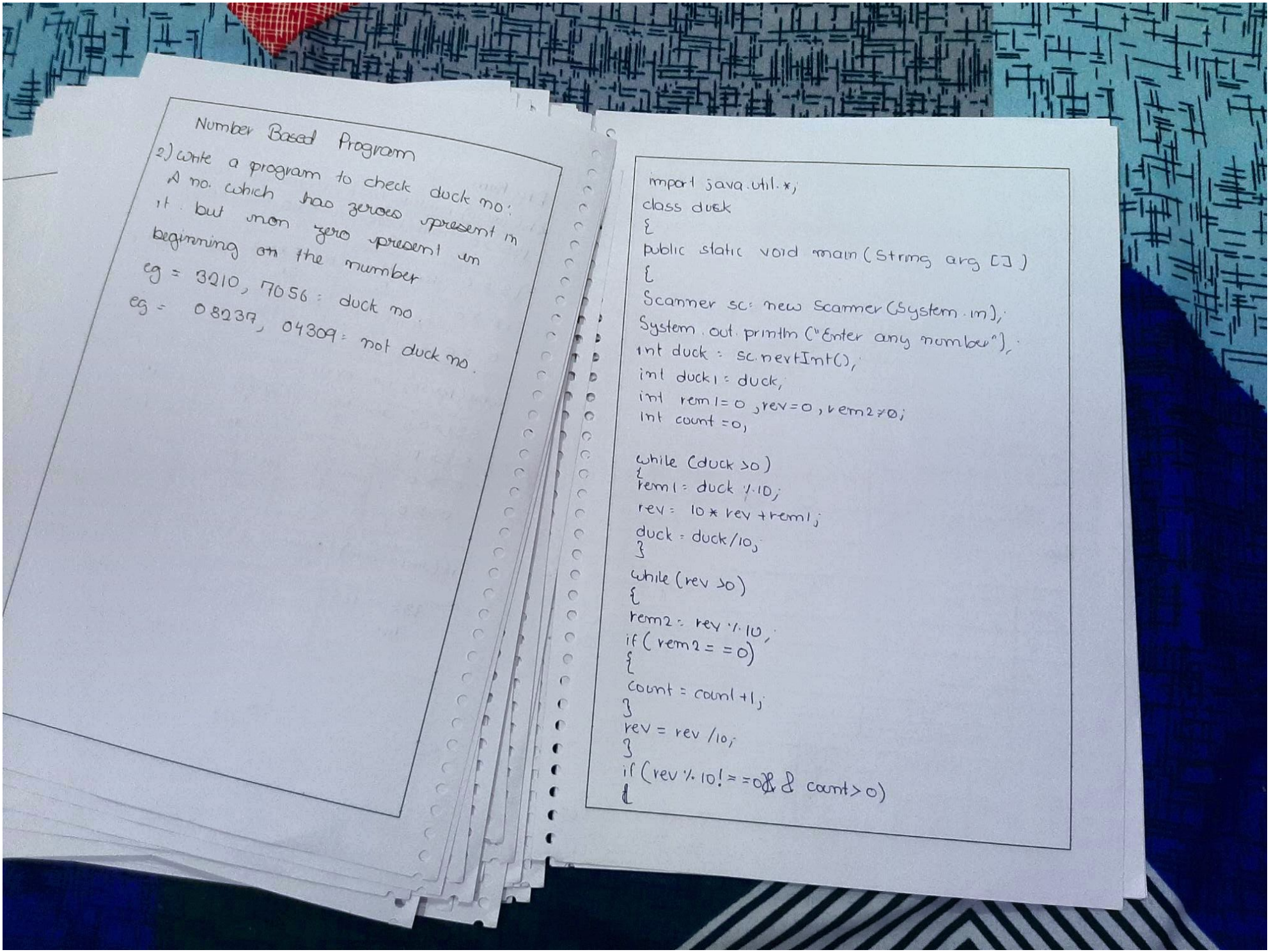


Dry Run

while (dis>0)	rem1 = dis % 10	rev = rev * 10 + rem1	dis = dis / 10
135 > 0 (t)	rem1 = 135%10 = 5	rev = 0 * 10 + 5 = 5	dis = 135 / 10 = 13
13 > 0 (t)	rem1 = 13 % 10 = 3	rev = 5 * 10 + 3 = 53	dis = 13 / 10 = 1
1 > 0 (t)	rem1 = 1 % 10 = 1	rev = 53 * 10 + 1 = 531	dis = 1 / 10 = 0
0 > 0 (f)	X	X	X

while (rev>0)	rem2 = rev % 10	count = count + 1	sum = Math.pow(rem2, count);
531 > 0 (t)	rem2 = 531 % 10 = 1	count = 0 + 1 = 1	sum = 0 + 1^1 = 1
53 > 0 (t)	rem2 = 53 % 10 = 3	count = 1 + 1 = 2	sum = 1 + 3^2 = 10
5 > 0 (t)	rem2 = 5 % 10 = 5	count = 2 + 1 = 3	sum = 10 + 5^3 = 135
0 > 0 (f)	X	X	X

if (dis1 == sum)	System.out.println ("disorium")	else	System.out.println ("not disorium")
135 == 135 (true)	disorium	(false)	(false)



Number Based Program

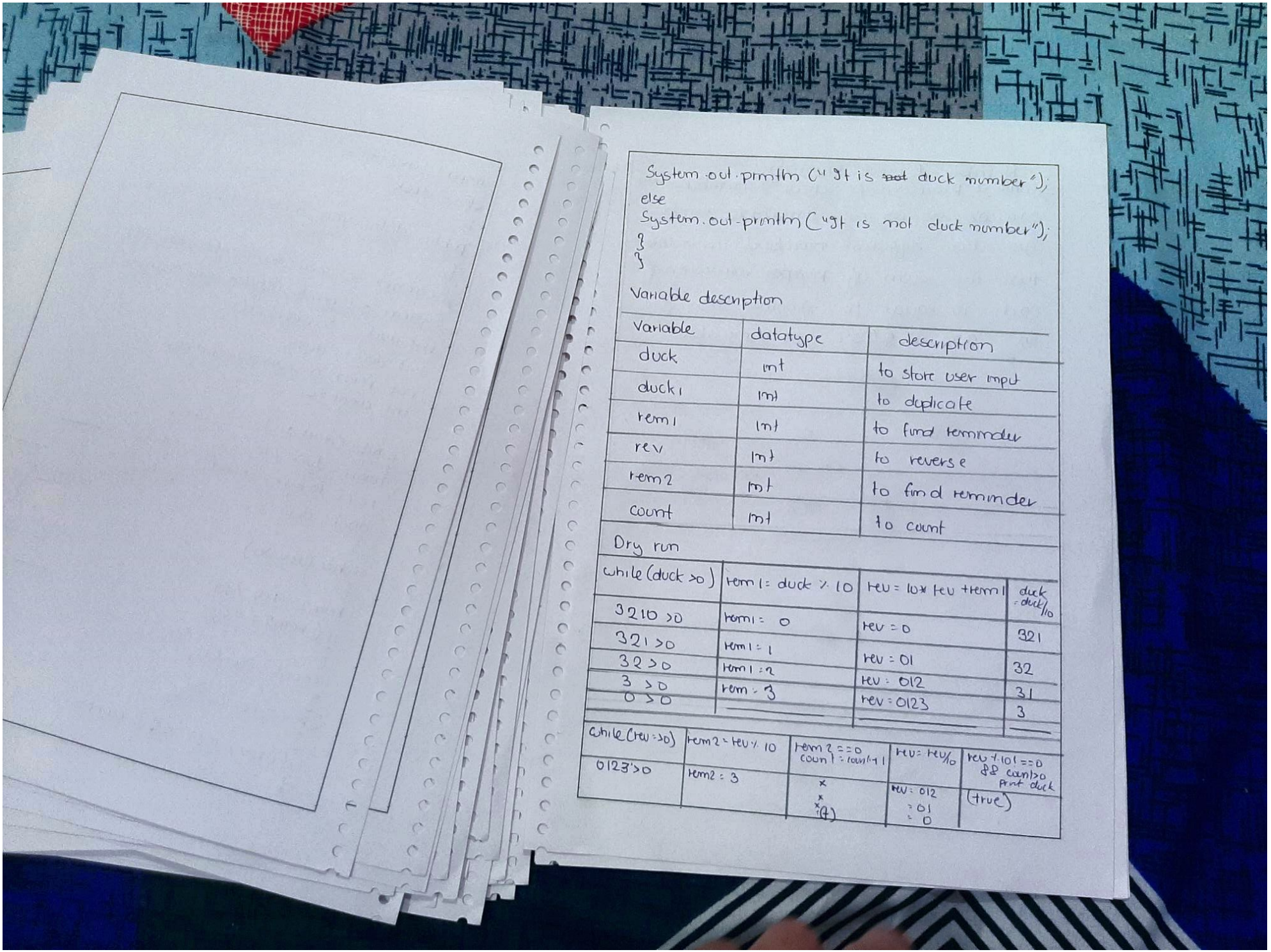
2) Write a program to check duck no:
A no. which has zeroes present in it but non zero present in beginning of the number

eg = 3210, 7056 : duck no.

eg = 08237, 04309 : not duck no.

```
import java.util.*;
class duck
{
public static void main(String arg[])
{
Scanner sc= new Scanner(System.in);
System.out.println("Enter any number");
int duck = sc.nextInt();
int duck1 = duck;
int rem1=0 ,rev=0 ,rem2=0;
int count=0;

while (duck>0)
{
rem1 = duck %10;
rev = 10 * rev +rem1;
duck = duck/10;
}
while (rev>0)
{
rem2 = rev %10;
if (rem2==0)
{
count = count+1;
}
rev = rev /10;
}
if (rev %10!==0 && count>0)
{
```



```
System.out.println("It is ~~not~~ duck number");
else
System.out.println("It is not duck number");
}
}
```

Variable description

Variable	datatype	description
duck	int	to store user input
duck1	int	to duplicate
rem1	int	to find remainder
rev	int	to reverse
rem2	int	to find remainder
count	int	to count

Dry run

while (duck > 0)	rem1 = duck % 10	rev = 10 * rev + rem1	duck = duck/10
3210 > 0	rem1 = 0	rev = 0	321
321 > 0	rem1 = 1	rev = 01	32
32 > 0	rem1 = 2	rev = 012	31
3 > 0	rem = 3	rev = 0123	3
0 > 0			

while (rev > 0)	rem2 = rev % 10	rem2 == 0 count1 = count+1	rev = rev/10	rev % 10 == 0 && count > 0 print duck
0123 > 0	rem2 = 3	x x x (f)	rev = 012 = 01 = 0	(true)

Number Based Program

3) Write a program to check 2 numbers are amicable no or not. amicable numbers are two different numbers so related that the sum of proper division of each is equal to another no.

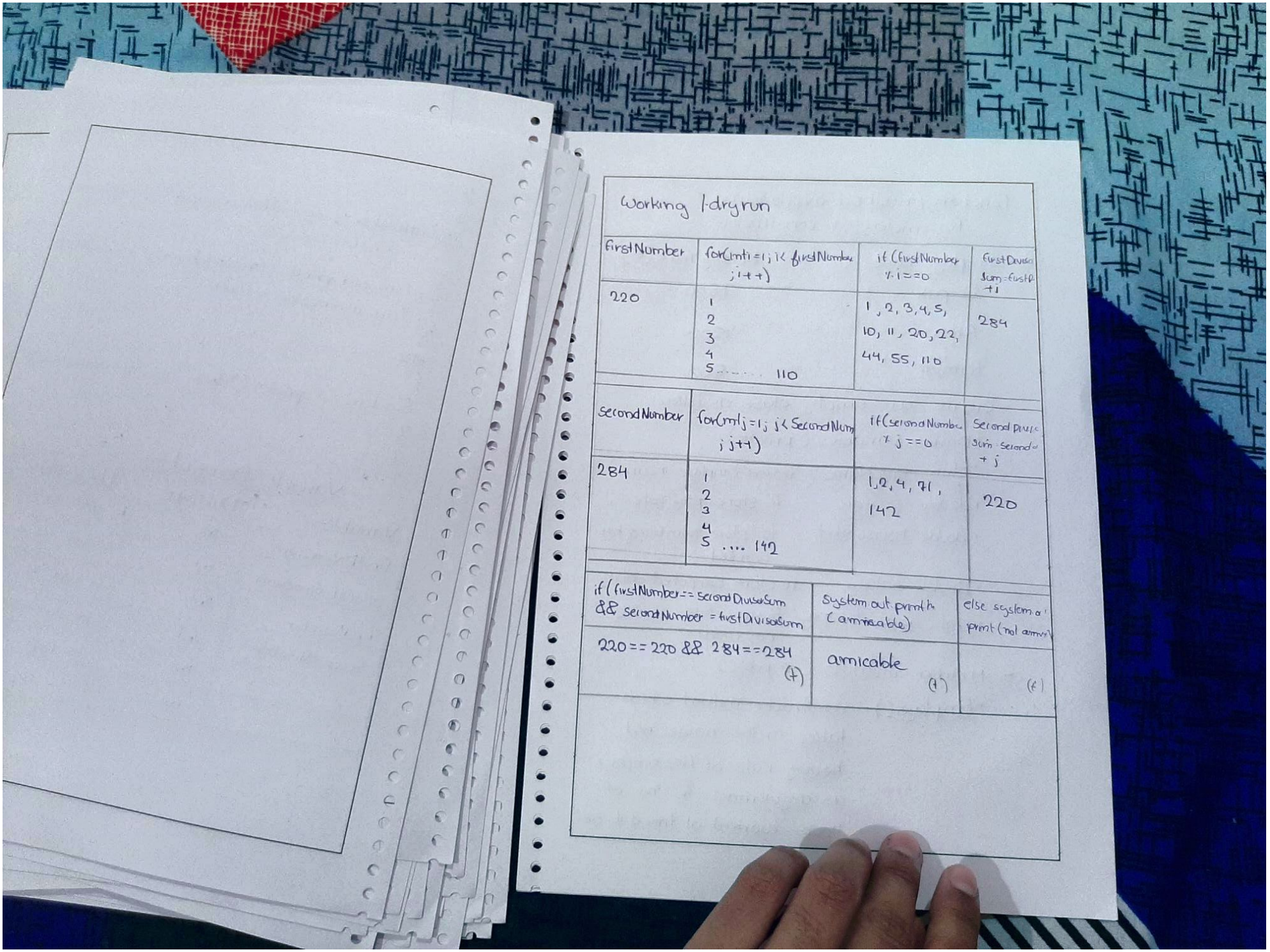
eg. 220 & 284 are amicable

```
import java.util.*;
class Amicablenumber
{
public static void main (String arg[])
{
int firstDivisorSum=0, secondDivisorSum=0;
int firstNumber, secondNumber;
Scanner sc= new Scanner(System.in);
System.out.prit1("Enter first number");
firstNumber = sc.nextInt();
System.out.print("Enter the second number");
secondNumber = sc.nextInt();
for(int i=1;i<firstNumber;i++)
{
if(firstNumber %i==0)
{
FirstDivisorSum = firstDivisorSum +i;
}
}
for(int j=1;j<secondNumber;i++)
{
if(secondNumber %j==0)
{
secondDivisorSum = secondDivisorSum +i;
```

```
}
}
if((firstNumber == secondDivisorSum) && (secondNumber
    == firstDivisorSum))
{System.out.println(firstNumber + " " + secondNumber
    + "are ammicable numbers.");
}
else
{
System.out.println("They are not ammicable no.");
}
}
}
```

Variable Description

Variable	Datatype	Description
firstDivisorSum	int	to add divisor
secondDivisorSum	int	to add divisor
firstNumber	int	to store user input
secondNumber	int	to store user input
i	int	to loop
j	int	to loop

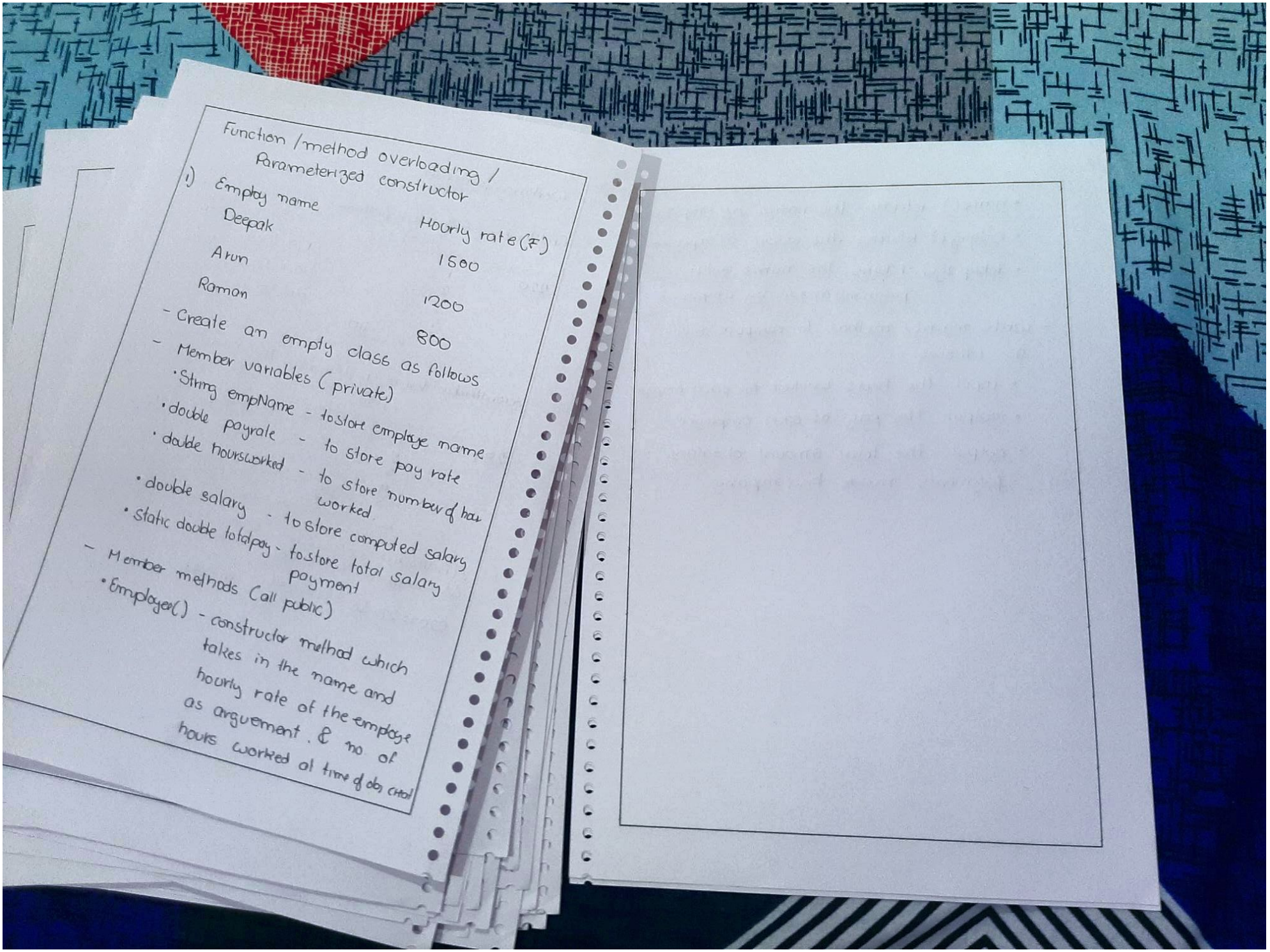


Working / dry run

firstNumber	for(int i=1; i< firstNumber; i++)	if (firstNumber % i==0	firstDivisorSum=firstD... +i
220	1 2 3 4 5 110	1, 2, 3, 4, 5, 10, 11, 20, 22, 44, 55, 110	284

secondNumber	for(int j=1; j< secondNum... ; j++)	if(secondNumber... % j==0	secondDivis... sum=second... + j
284	1 2 3 4 5 142	1, 2, 4, 71, 142	220

if (firstNumber == secondDivisorSum && secondNumber = firstDivisorSum	System.out.println (amicable)	else system.o... print (not amic...
220 == 220 && 284 == 284 (T)	amicable (T)	(F)


Function / method overloading /
Parameterized constructor
1) Employ name
Hourly rate (₹)
Deepak
1500
Arun
1200
Raman
800
- Create an empty class as follows
- Member variables (private)
• String empName - to store employe name
• double payrate - to store pay rate
• double hoursworked - to store number of hou
worked
• double salary - to store computed salary
• static double totalpay - to store total salary
payment
- Member methods (all public)
• Employee() - constructor method which
takes in the name and
hourly rate of the employe
as arguement. & no of
hours worked at time of obj creat

• name() - returns the name of employee
• Salary() returns the salary of employee
• totalpay() - returns the name Salary payment made by company
- write main() method to compute salary as follows
• input the hours worked for each employee
• output the pay of each employee
output the total amount of salary payments made by company.