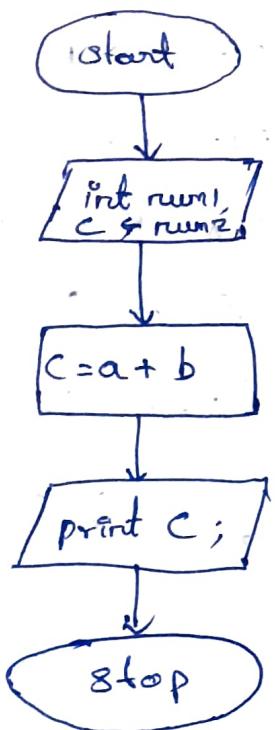


Flow chart Problems

- 1) Draw a flow chart for adding two numbers;



include <stdio.h>

```
int main()
```

```
{
```

```
int num1, num2, temp;
```

```
scanf ("%d %d", &num1, &num2);
```

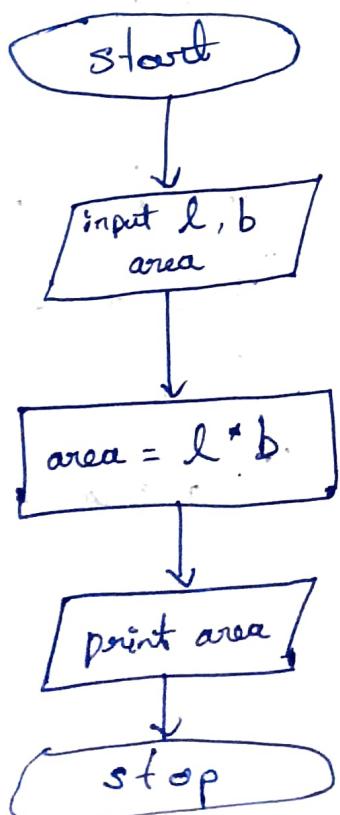
```
temp = num1 + num2;
```

```
printf ("add %d", temp);
```

```
return 0;
```

```
}
```

- 2) draw a flow chart to print area of rectangle;



include <stdio.h>

```
int main()
```

```
{
```

```
int l, b, area;
```

```
scanf ("%d %d", &l, &b);
```

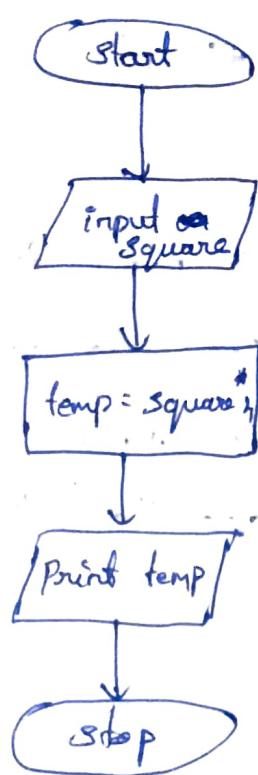
```
area = l * b
```

```
printf ("%d area of rectangle", area);
```

```
return 0;
```

```
}
```

3) draw a flow chart to print perimeter of a Square;



include <stdio.h>

int main()

{

int Square, temp;

printf("Enter the square : ");

scanf("%d", &Square);

temp = Square * 4;

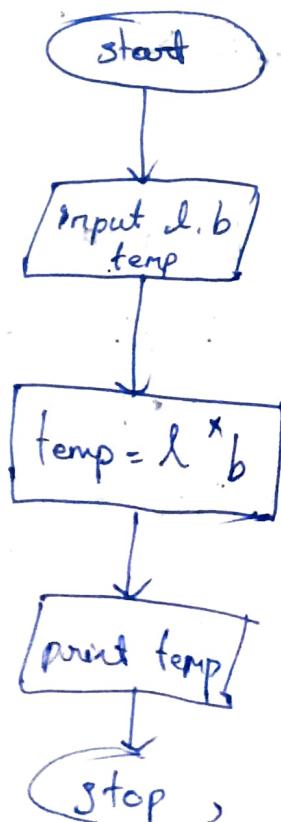
printf("%d, is a perimeter

of a square, temp

return 0;

}

4) draw a flow chart to print area of square



include <stdio.h>

int main()

{

int ; len, br, temp;

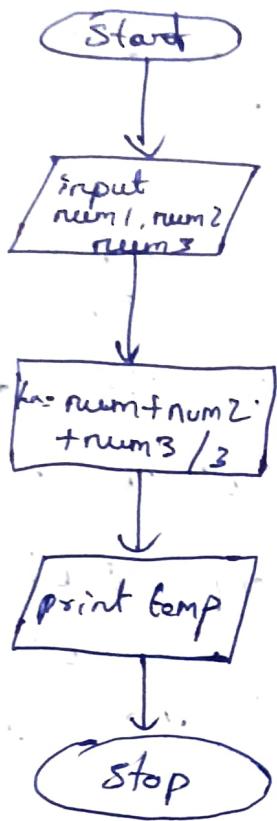
scanf("%d,%d,%d,%d", &len, &br);

temp = l * b;

printf("%d area of square", temp)

}

3) draw a flow chart to print the average of three numbers

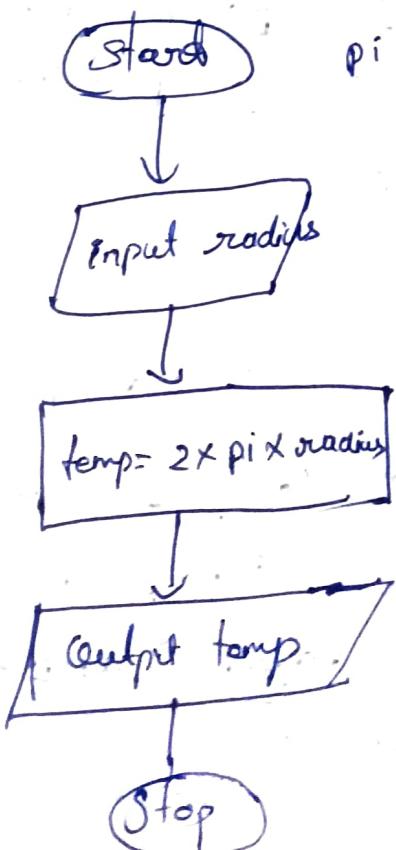


```
#include <stdio.h>
int main()
{
    float num1, num2, num3;
    scanf("%f %f %f", &num1, &num2, &num3);

    float temp = (num1 + num2 + num3) / 3;

    printf("The average is %.2f", temp);
    return 0;
}
```

b) draw a flow chart to print perimeter of a circle

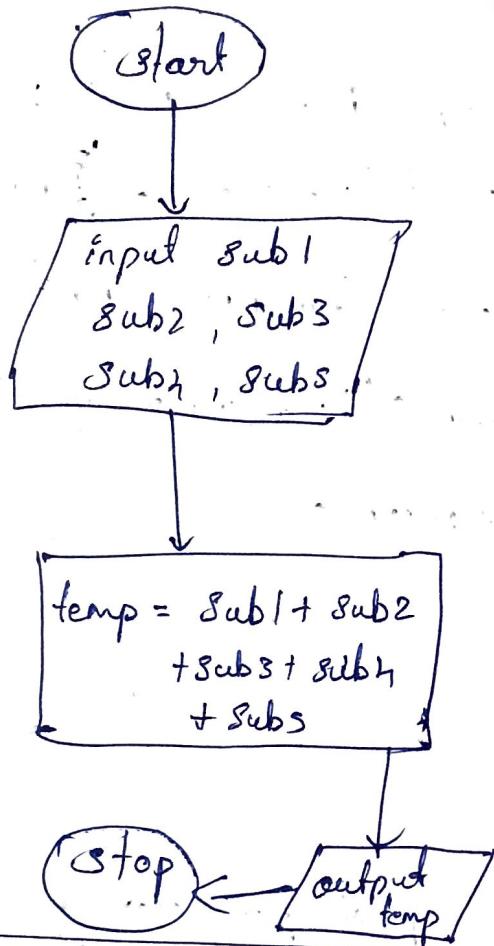


```
#include <stdio.h>
#define pi = 3.1416
int main()
{
    float radius, temp;
    scanf("%f", &radius);

    temp = 2 * pi * radius;

    printf("%.2f", temp);
    return 0;
}
```

7) draw a flow chart to print the sum of 5 subjects of 8 students.



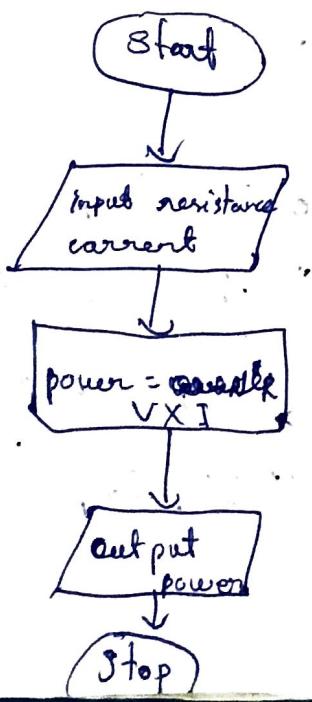
```

#include <stdio.h>
int main()
{
    int sub1, sub2, sub3,
    sub4, sub5;
    scanf("%d,%d,%d,%d,%d",
          &sub1, &sub2, &sub3, &sub4, &sub5);
}
  
```

```

float temp = sub1 + sub2 + sub3
            + sub4 + sub5;
printf("%f", temp);
return 0;
}
  
```

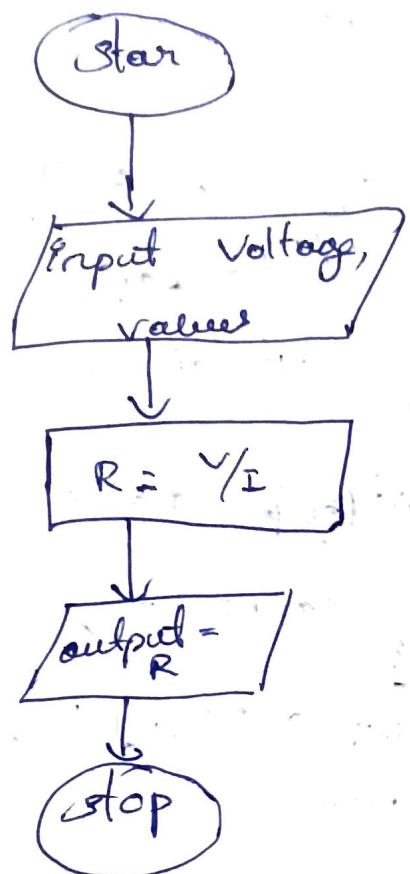
8) draw a flow chart to print the power after The current and resistance values are provided



```

#include <stdio.h>
int main()
{
    int resistance, current, power;
    scanf("%d,%d", &resistance, &current);
    power = current * resistance;
    printf("%d", power);
}
  
```

g) draw a flow chart to print the values of resistance, when voltage and current values are provided.



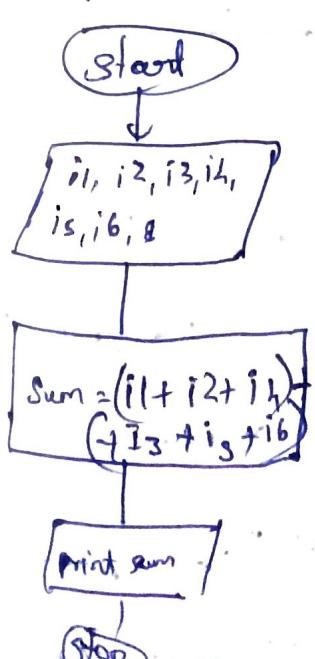
```

#include <stdio.h>
int main()
{
    int voltage, current;
    scanf("%d, %d", &voltage, &current);
    float resistance = voltage / current;
    printf("%f", resistance);
    return 0;
}
  
```

h) draw a flow chart to take the values of

$i_1, i_2, i_3, i_4, i_5, i_6$ current values following

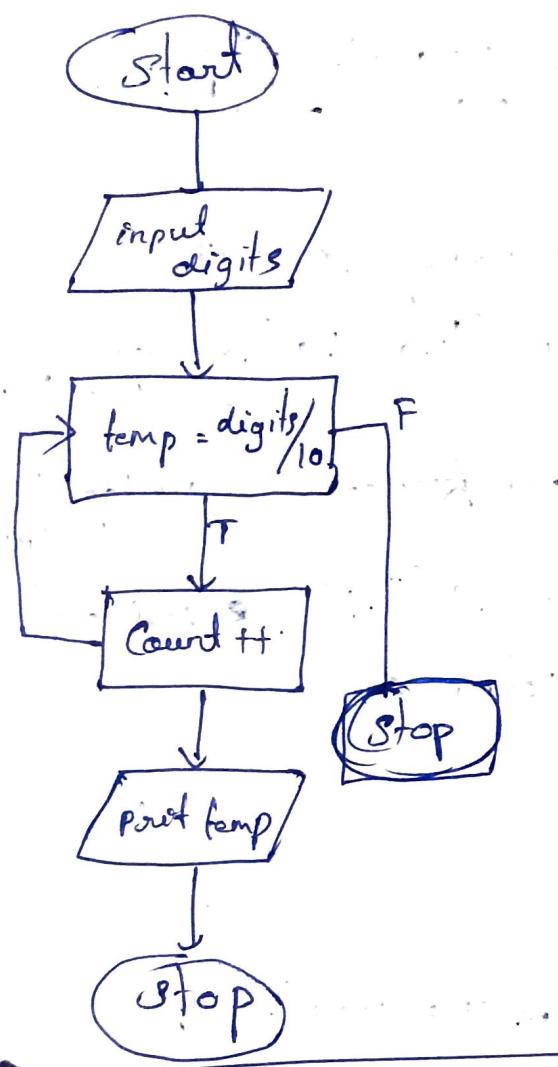
from the point and calculate sum of current.



```

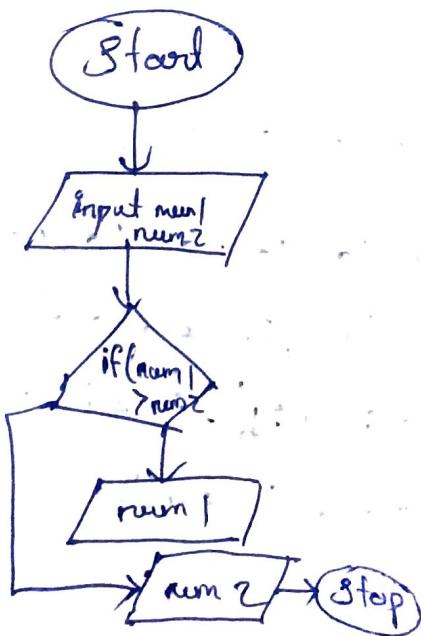
#include <stdio.h>
int main()
{
    int i1, i2, i3, i4, i5, i6, sum;
    scanf("%d, %d, %d, %d, %d, %d", &i1, &i2, &i3, &i4, &i5, &i6);
    sum = (i1 + i2 + i3 + i4 + i5 + i6);
    printf("%d", sum);
    return 0;
}
  
```

1) Draw a flow chart to read the number of digits in given number.



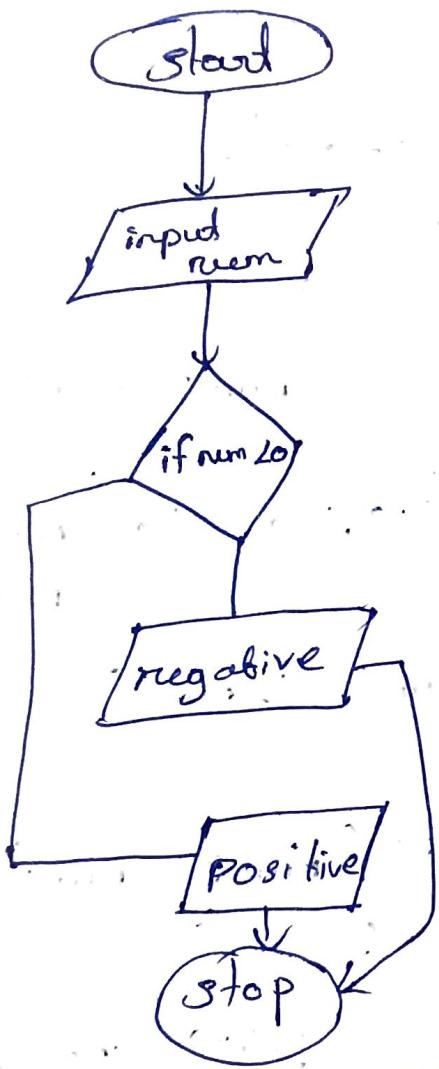
```
#include <stdio.h>
int main()
{
    int digits, count = 0;
    scanf("%d", &digits);
    while (digits != 0)
    {
        int temp = digits % 10;
        count++;
        digits /= 10;
    }
    printf("%d", count);
    return 0;
}
```

2) draw a flow chart to read the two numbers from user and print bigger value



```
#include <stdio.h>
int main()
{
    int num1, num2;
    scanf("%d %d", &num1, &num2);
    if (num1 > num2)
        printf("%d", num1);
    else
        printf("%d", num2);
}
```

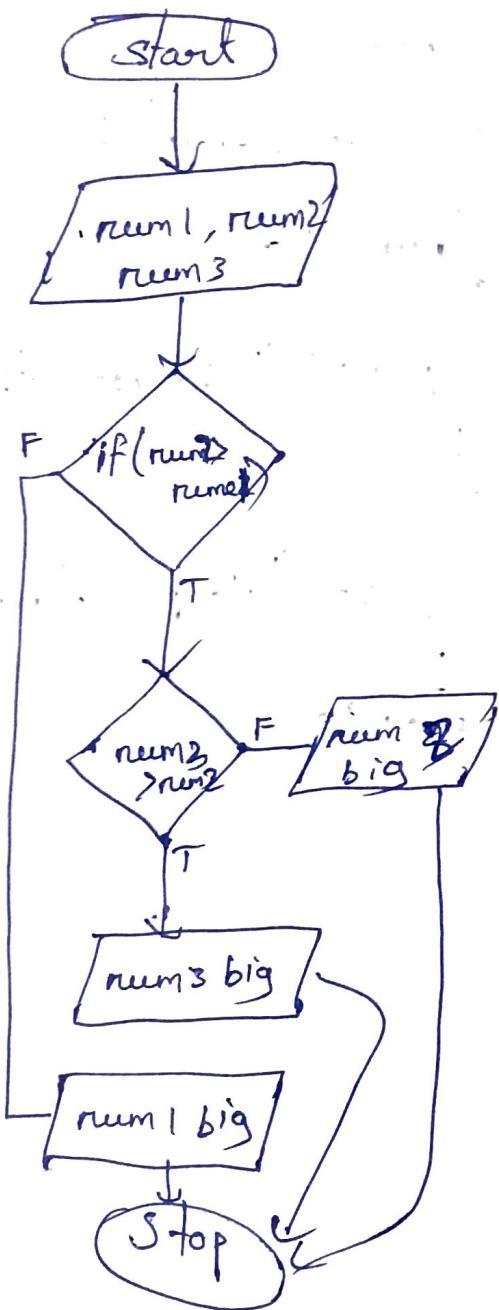
(3) draw a flow chart to point whether the given number is positive or negative:-



```

#include <stdio.h>
int main () {
    int num ;
    scanf ("%d", &num);
    if (num < 0)
        printf ("%d negative");
    else
        printf ("%d positive", num);
    return 0;
}
  
```

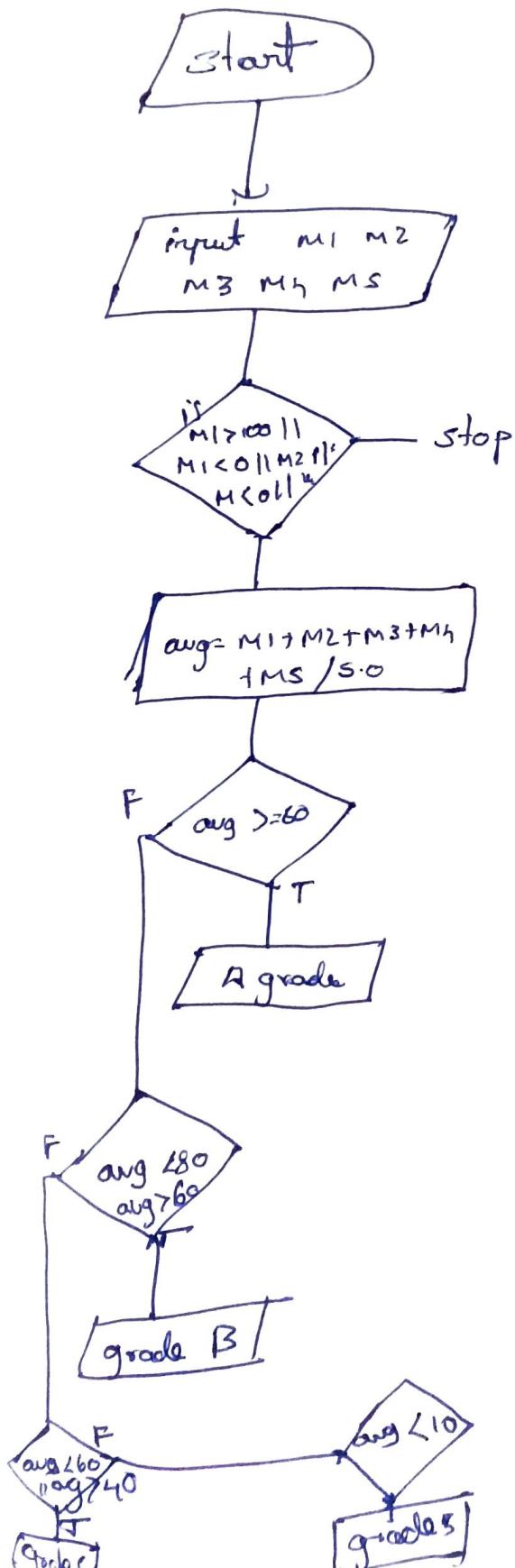
- 13) draw a flow chart to get marks of 5.
- 14) draw a flow chart to read the three numbers from the user and print the biggest value:



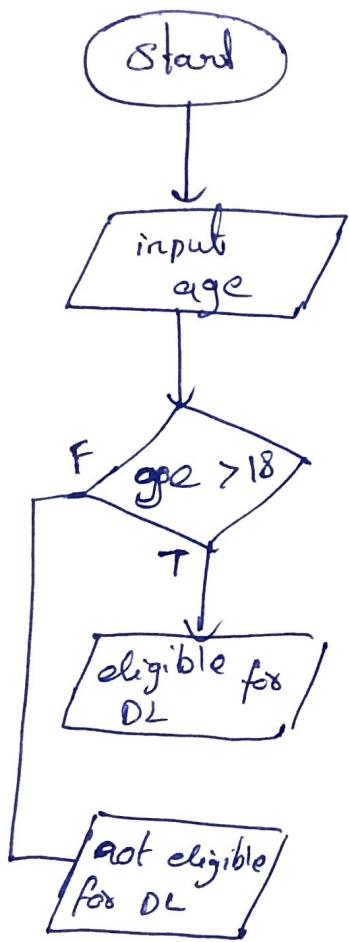
```

#include <stdio.h>
int main ()
{
    int num1, num2, num3;
    Scanf ("%d, %d, %d", num1, num2, num3);
    if (num2 > num1)
    {
        if (num3 > num2)
            printf ("%d", num3);
        else
            printf ("%d", num2);
    }
    else
        printf ("%d", num1);
}
  
```

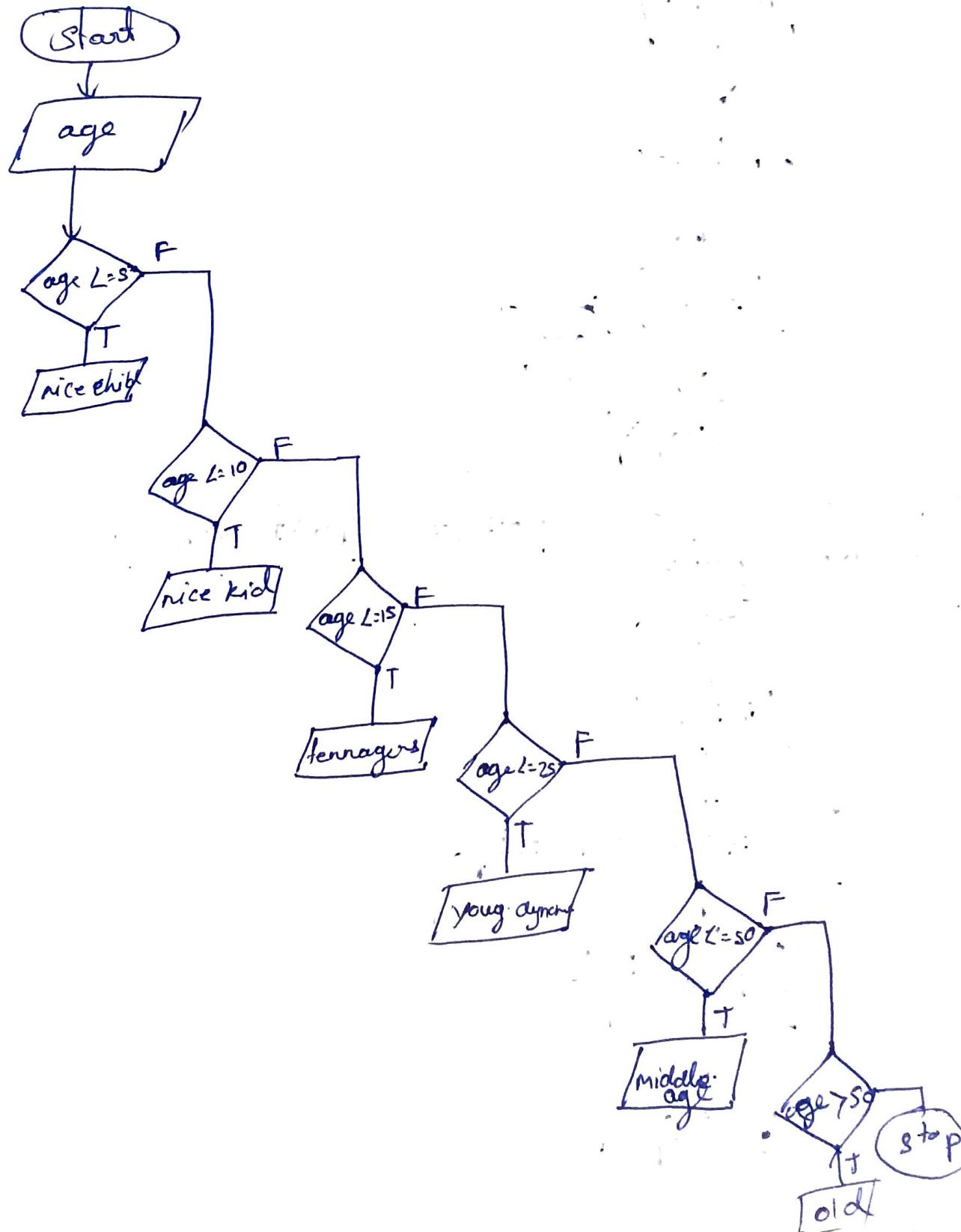
15) draw a flow chart to get marks of 5 subjects of a student, and point grade grade A is above 80%, B is above 60%. C, is above 40%, D is below 40%.



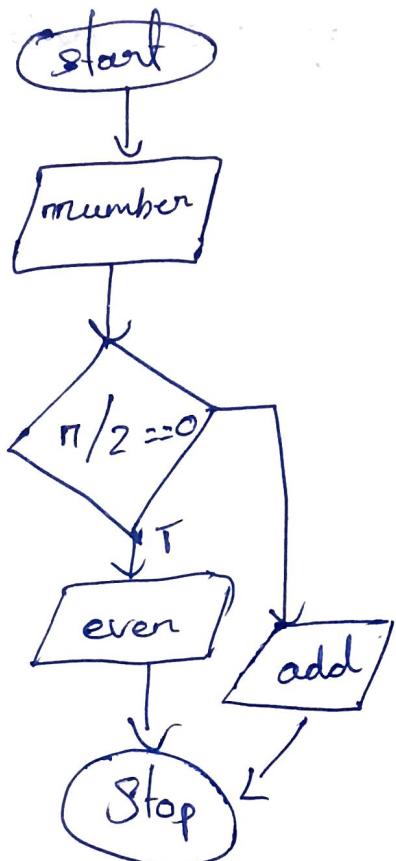
16) draw a flow chart to get age of a person to print whether he is eligible for DL or not



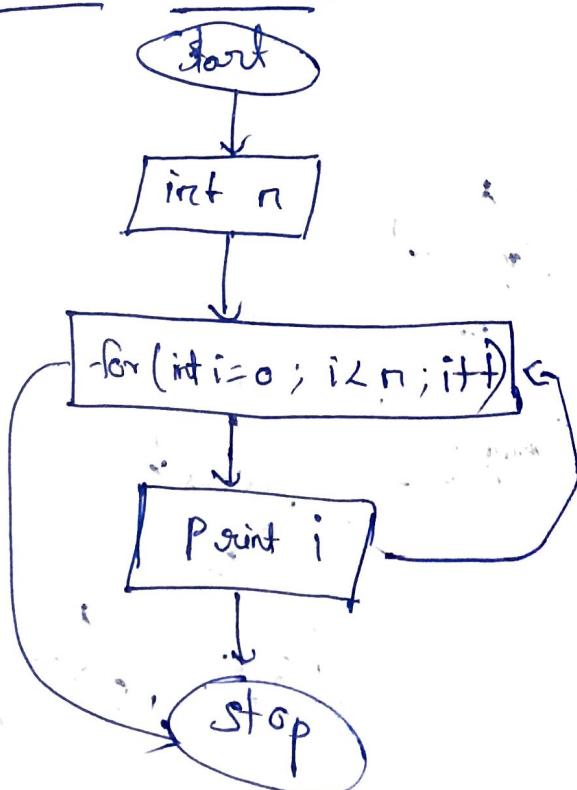
7) draw a flow chart to get age of a person to send weather print greeting message if age ≤ 5 nice child, age ≤ 10 nice kid, age ≤ 15 teenagers, age ≤ 25 young dynamic, age ≤ 50 middle age age ≥ 50 old:



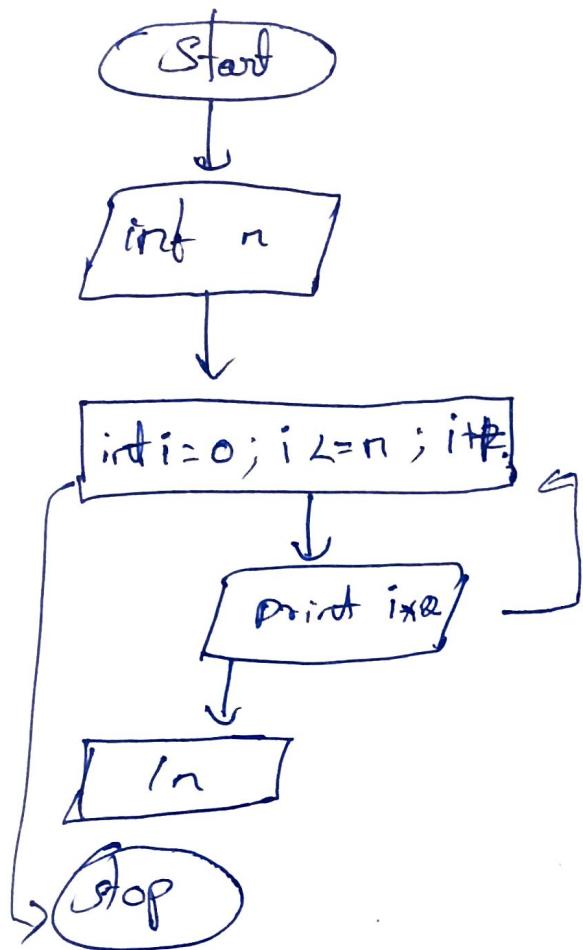
18) draw a flow chart to get whether the given number is even or odd:



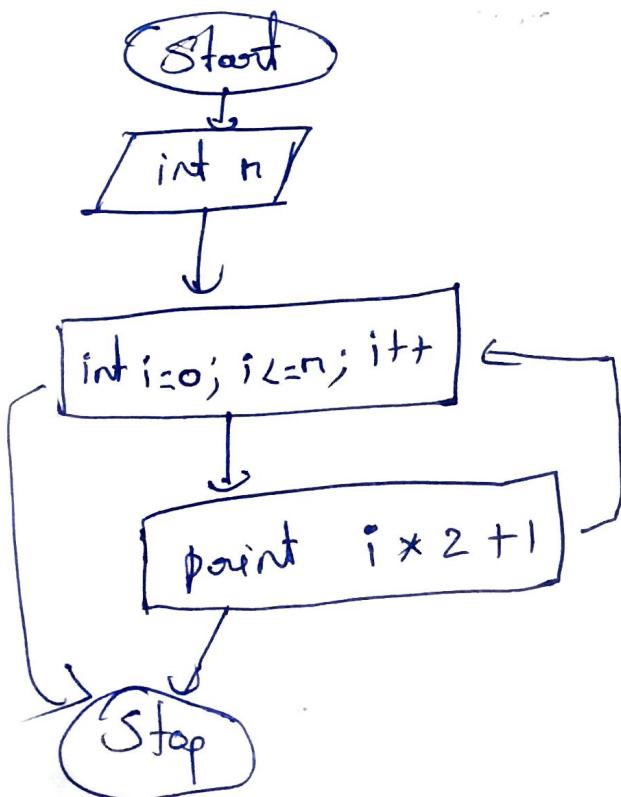
19) draw a flow chart to print first 20 natural numbers



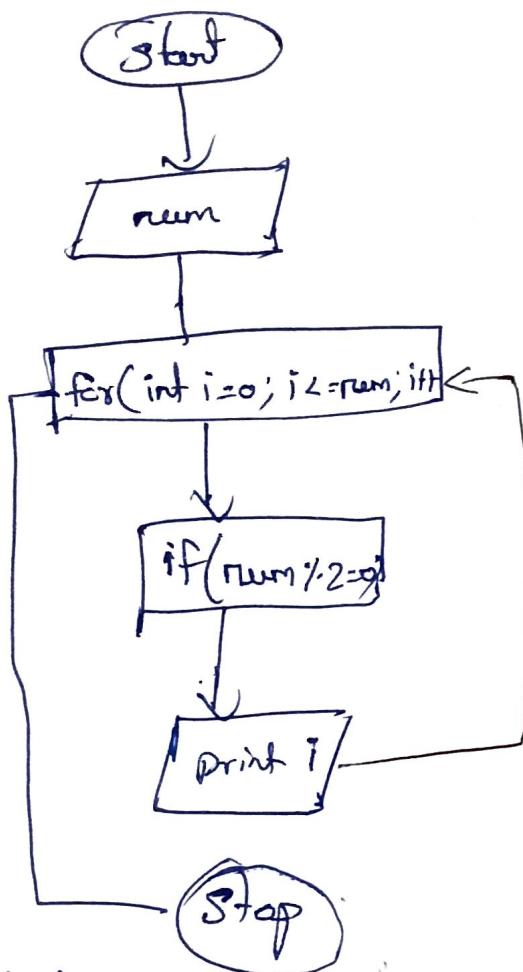
2) draw a flow chart to print first 10 even numbers:



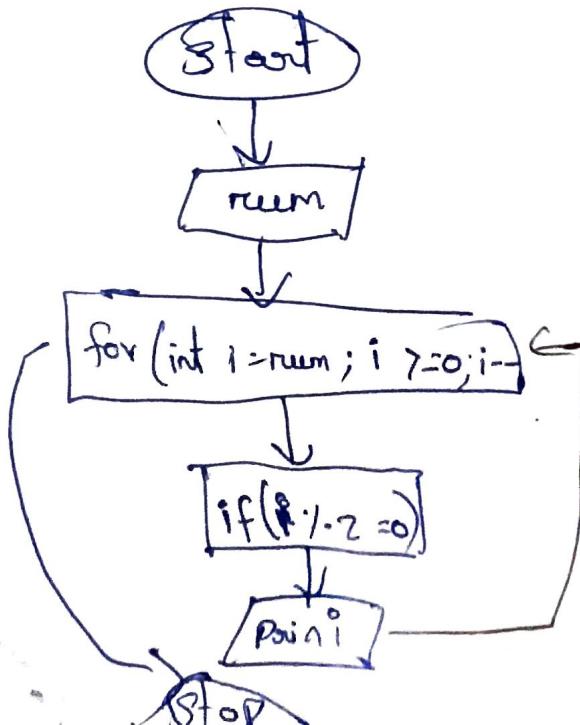
2) draw a flow chart to print first 30 odd numbers



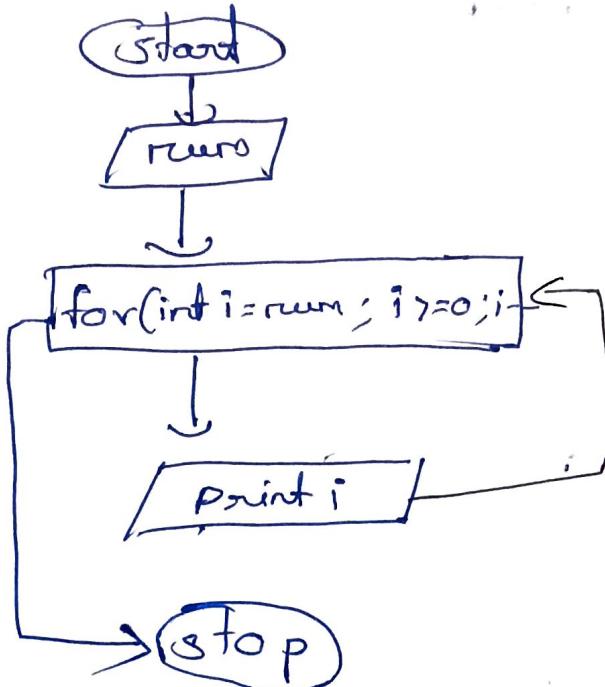
22) draw a flow chart to print all the even numbers in range?



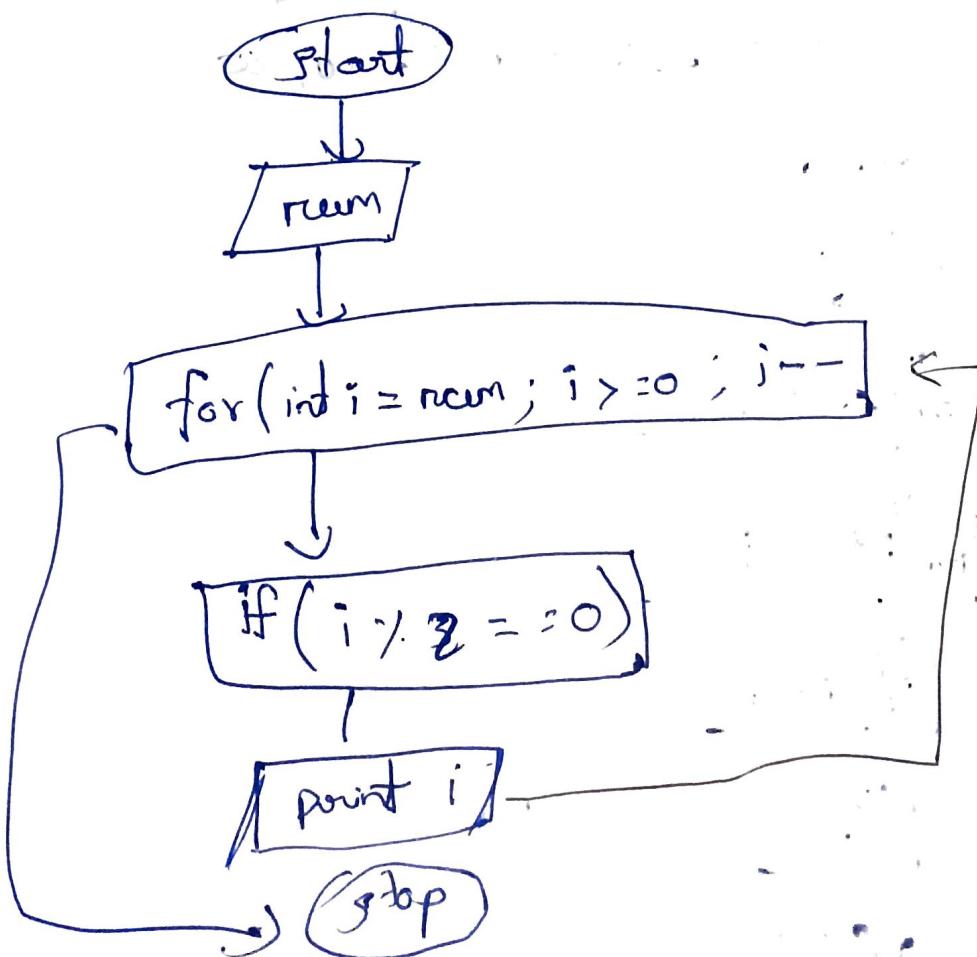
23) draw a flow chart to print first 20 even numbers in reverse order.



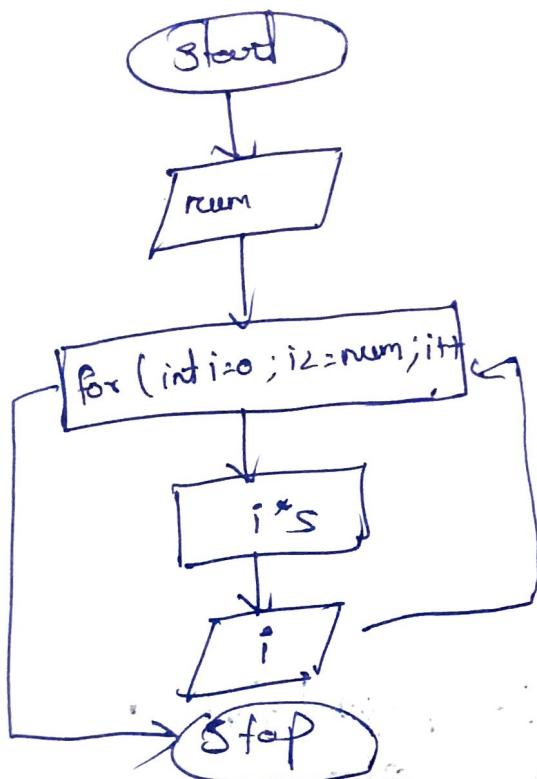
24) draw a flow chart to print first n natural numbers in reverse order



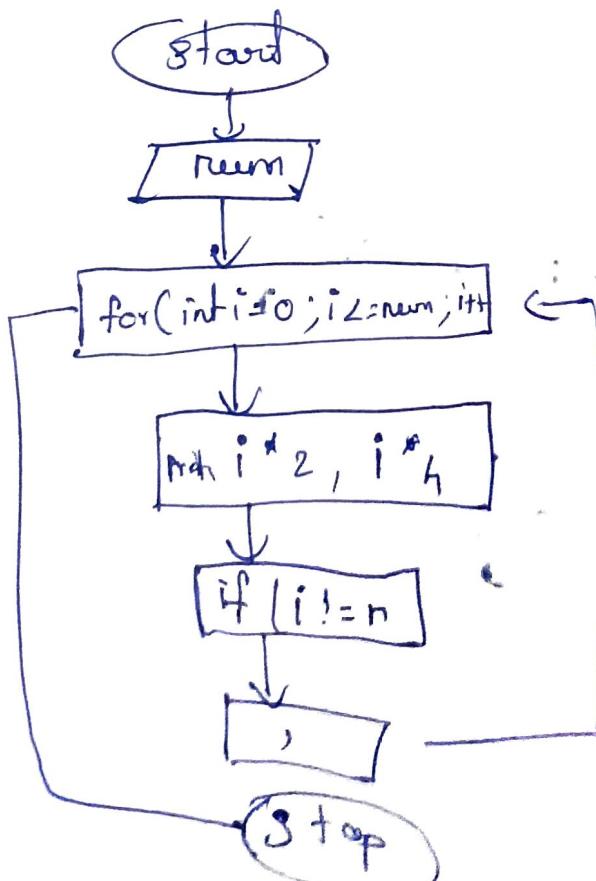
25) draw a flow chart to print first n even numbers in reverse order



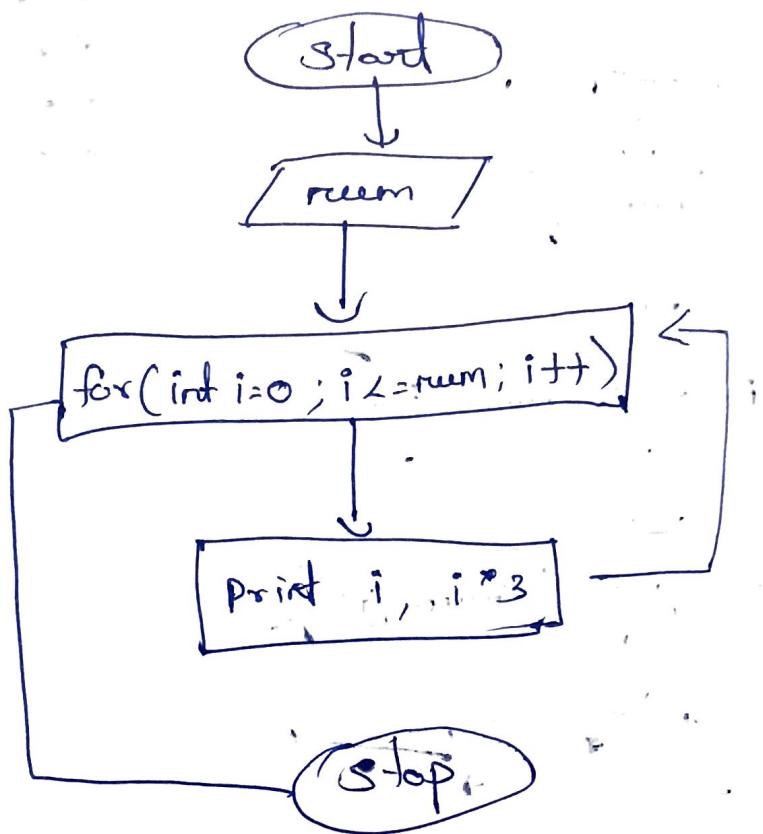
26) draw a flow chart to print the series of $5, 10, 15, 20, \dots, n$ numbers



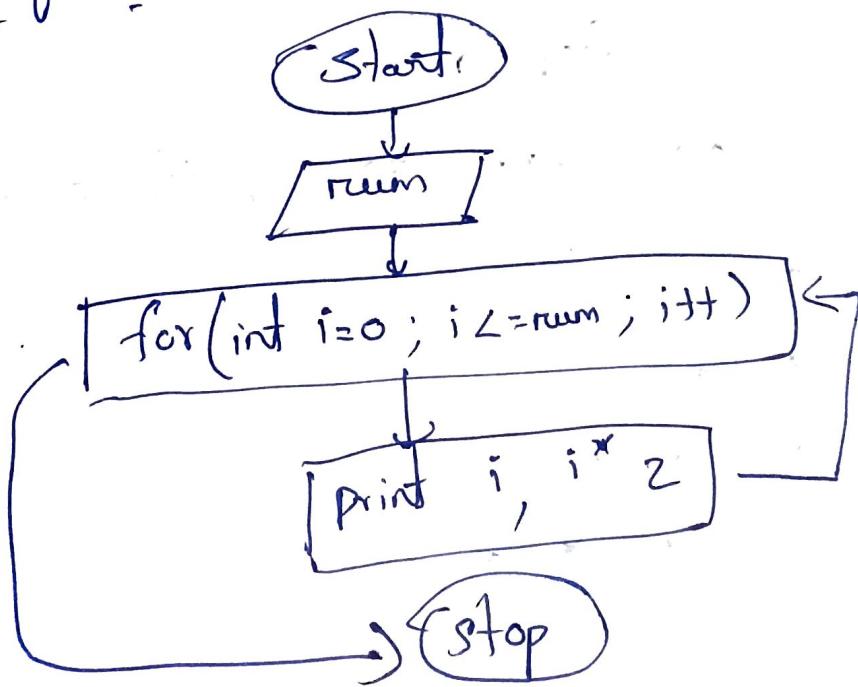
27) draw a flow chart to print the series of numbers $2, 4, 4, 8, 6, 12, 8, 16, 10, 20$



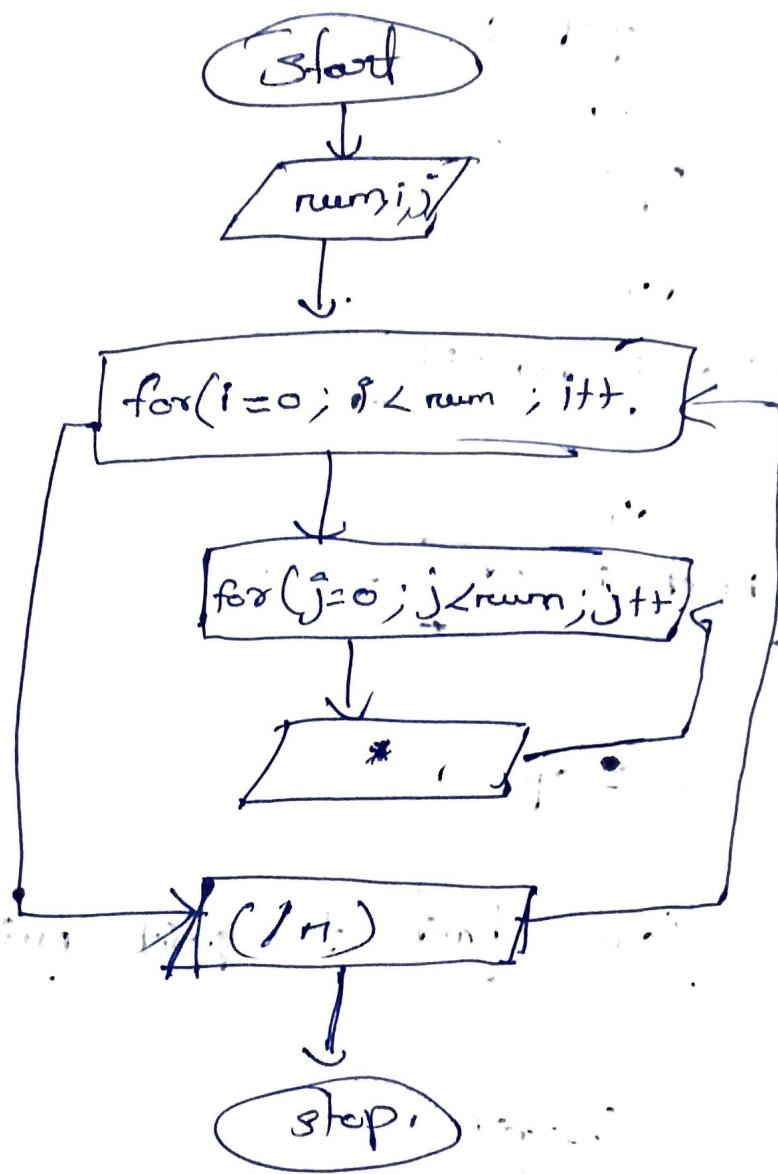
28) draw a flow chart to print multiplication table of 3



29) draw a flow chart to print multiplication table of 2



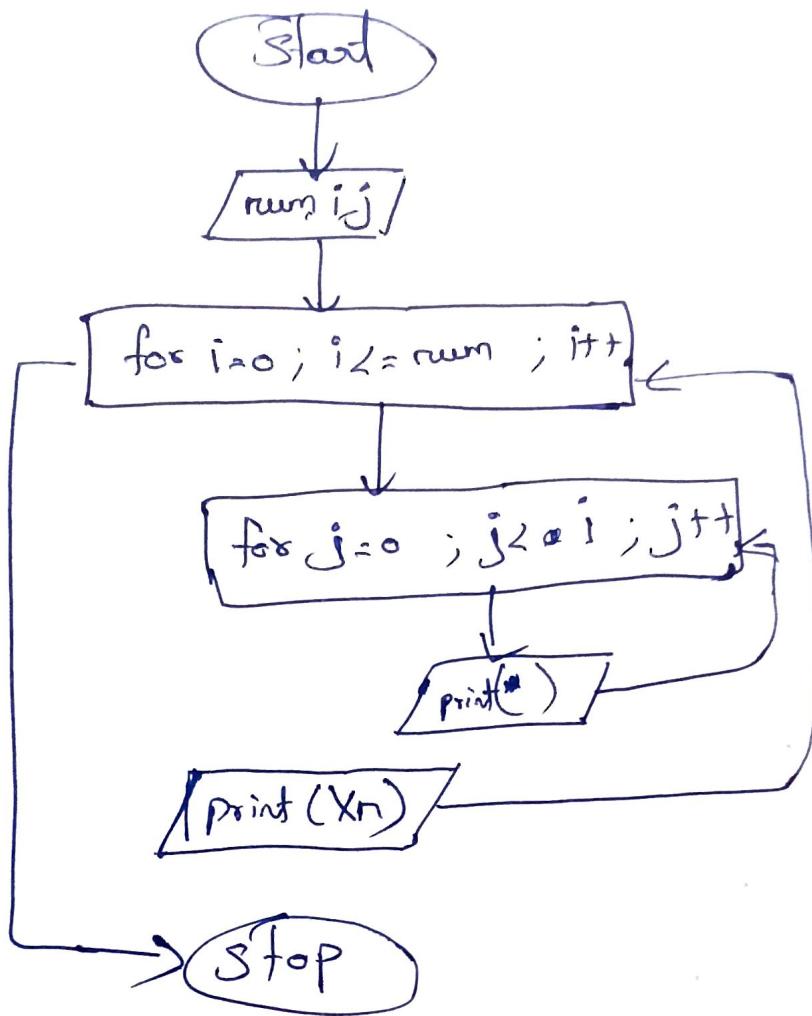
30) draw a flow chart to print given pattern.



* * * *
* * * *
* * * *
* * * *

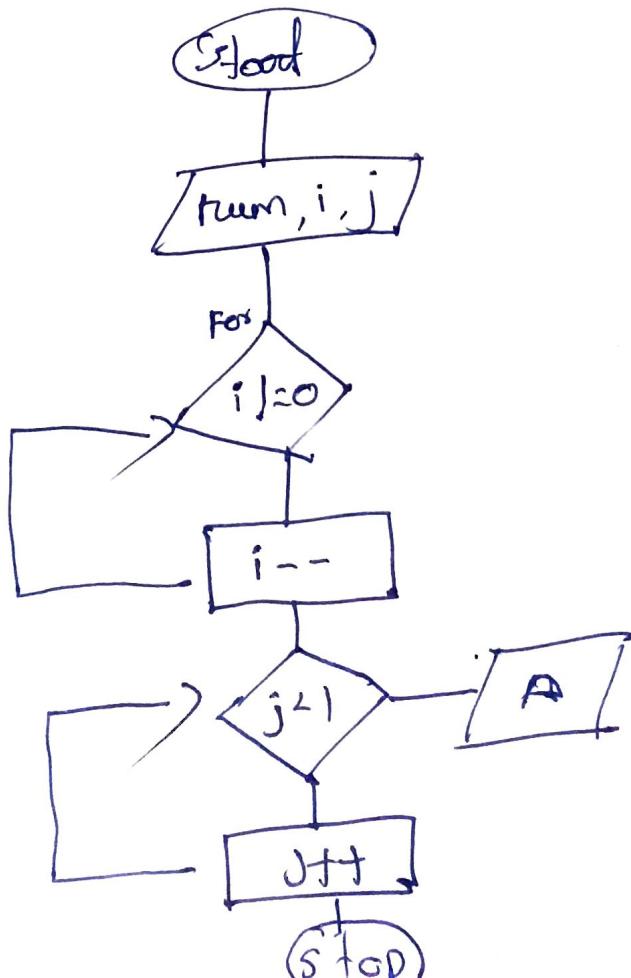
31) draw a flow chart to print given pattern

* . . .
* . .
* . . . :
* - - - *

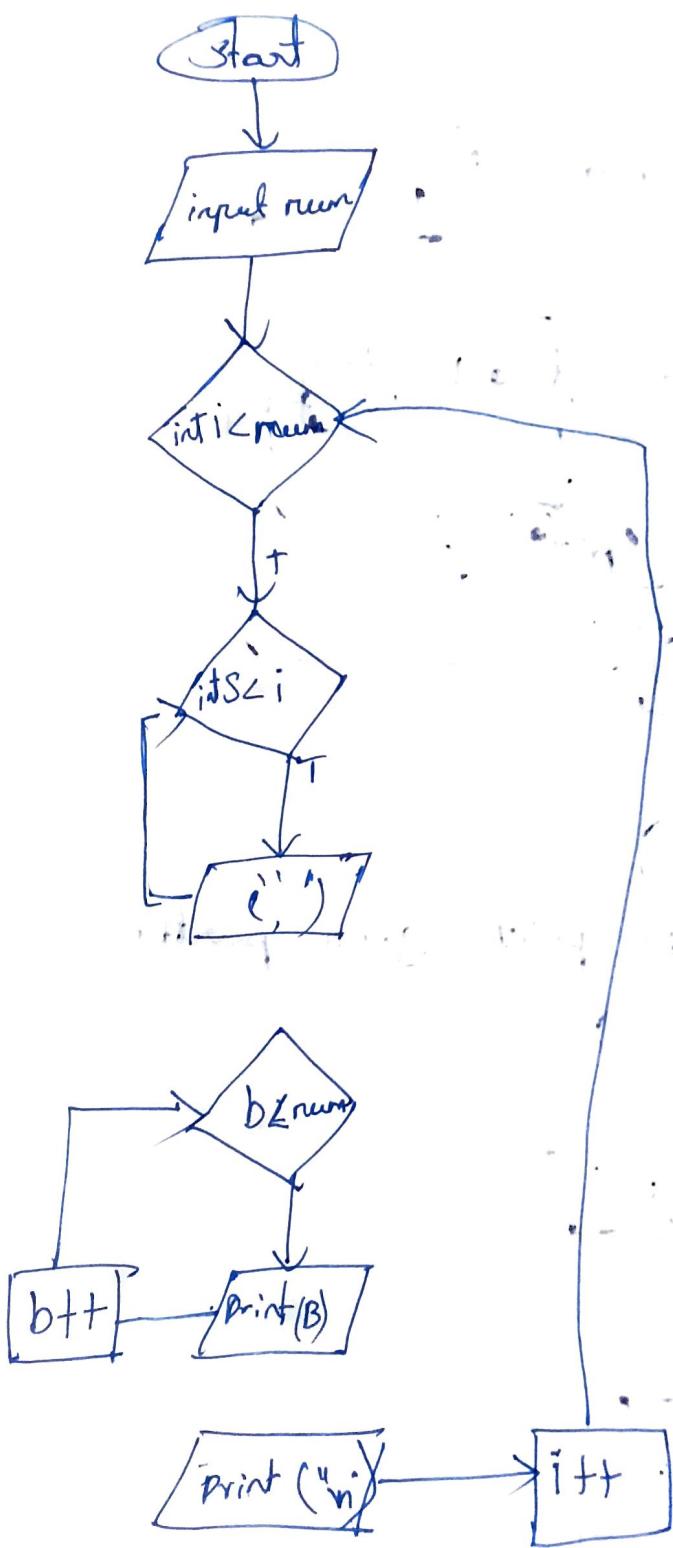


32) draw a flow chart to print given pattern

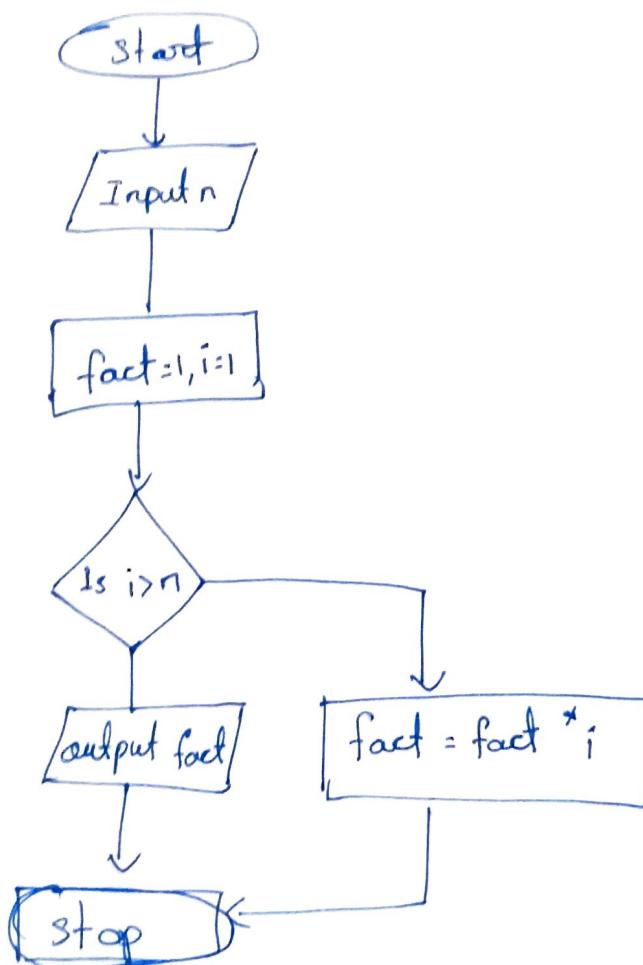
AAAAAA
 A AAAA
 AAAA
 AA
 A



33) draw a flow chart for print given pattern



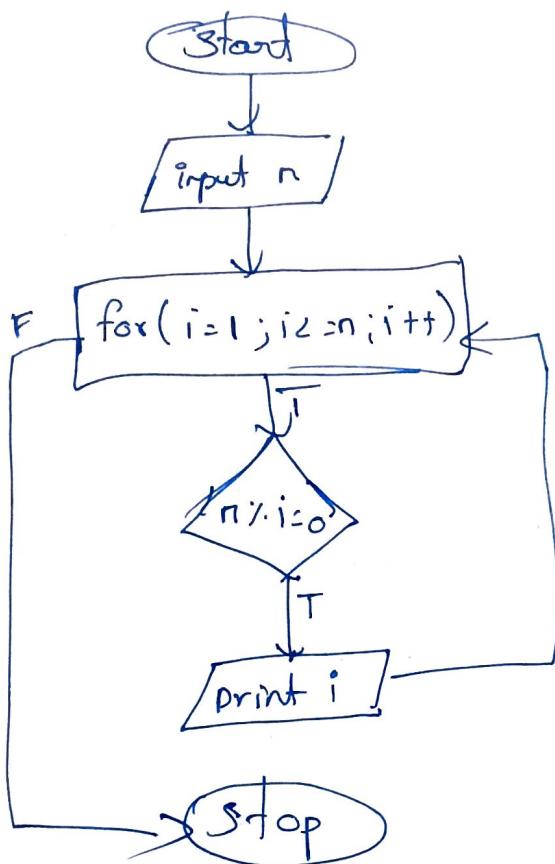
36) draw a flow chart to print factorial of given number :-



Algorithm Steps

- 1) start
- 2) get input n
- 3) set fact as a 1
- 4) set i as a 1
- 5) Repeat until $i > n$
 - a) $fact = fact \times i$
- b) $i = i + 1$
- c) output fact
- d) stop

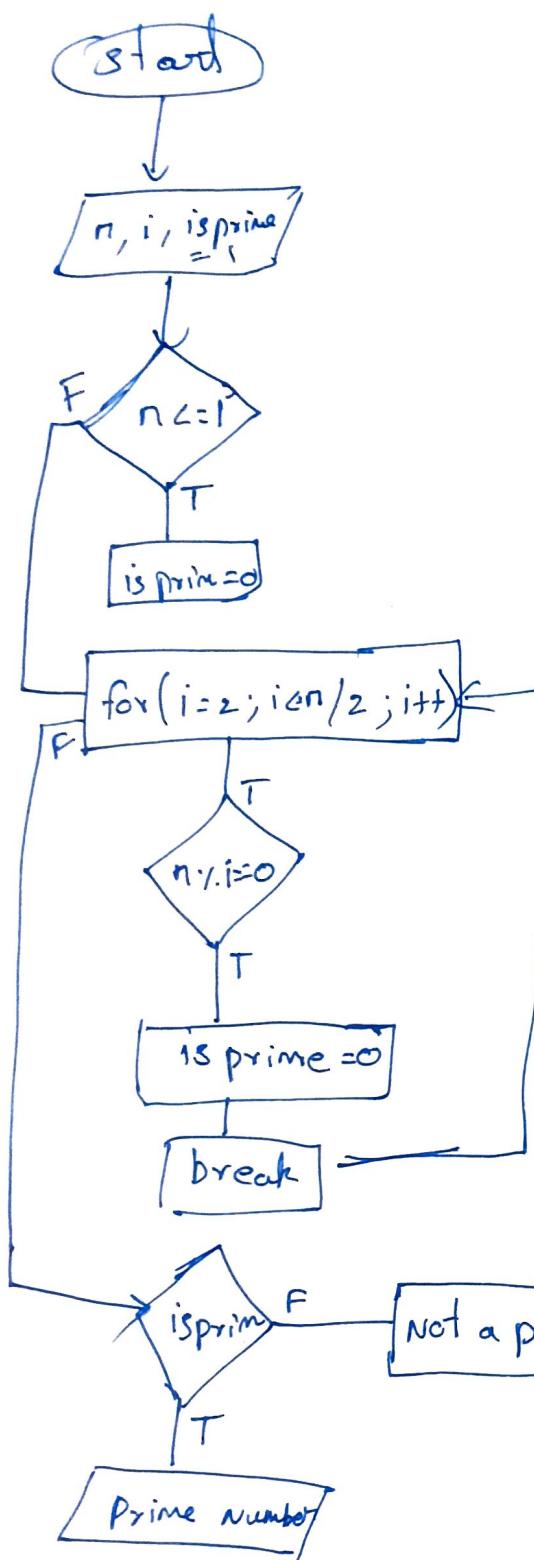
37) draw a flow chart to print factors of given numbers :-



Algorithm

- 1) Start
- 2) get input n
- 3) Set i as a 1
- 4) Repeat while $i \leq n$
 - a. if ($n \% i = 0$)
print i (, it's a factor)
 - b. $i = i + 1$
- 5) End

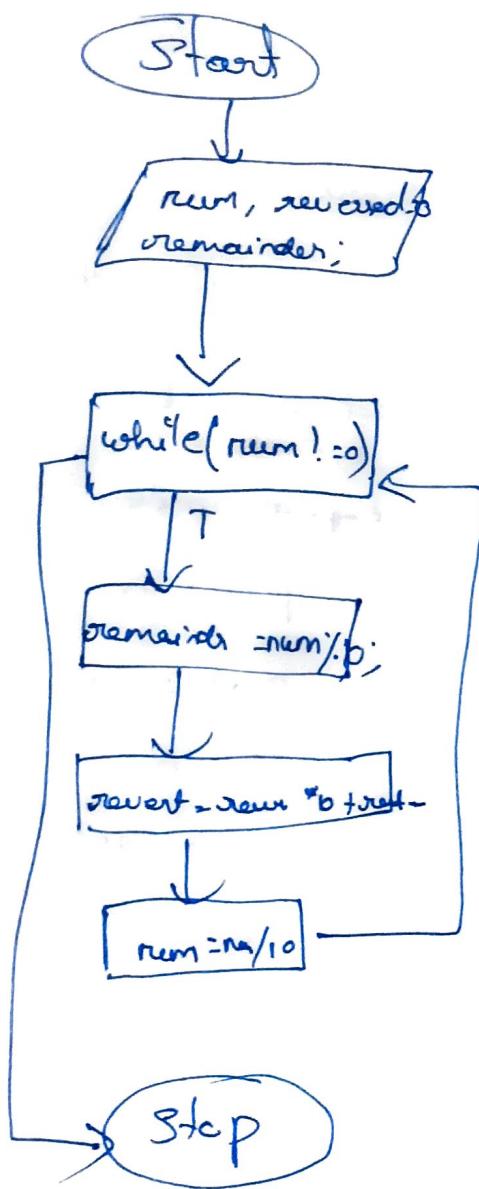
38) draw a flow chart to point whether the given number is prime or not :-



Algorithm :-

- 1) start
- 2) input the number n
- 3) if $n \leq 1$
 n is not a prime
Go to step 7
- 4) initialize $i = 2$
- 5) Repeat while $i \leq n/2$
if $n \% i == 0$
 n is not a prime
Go to step 7
Increment i by 1 and
repeat
- 6) if loop completes without
finding a divisor
 n is a prime number
step 7 print the result
- 8) Stop

39) draw a flow chart to print reverse of a given number:

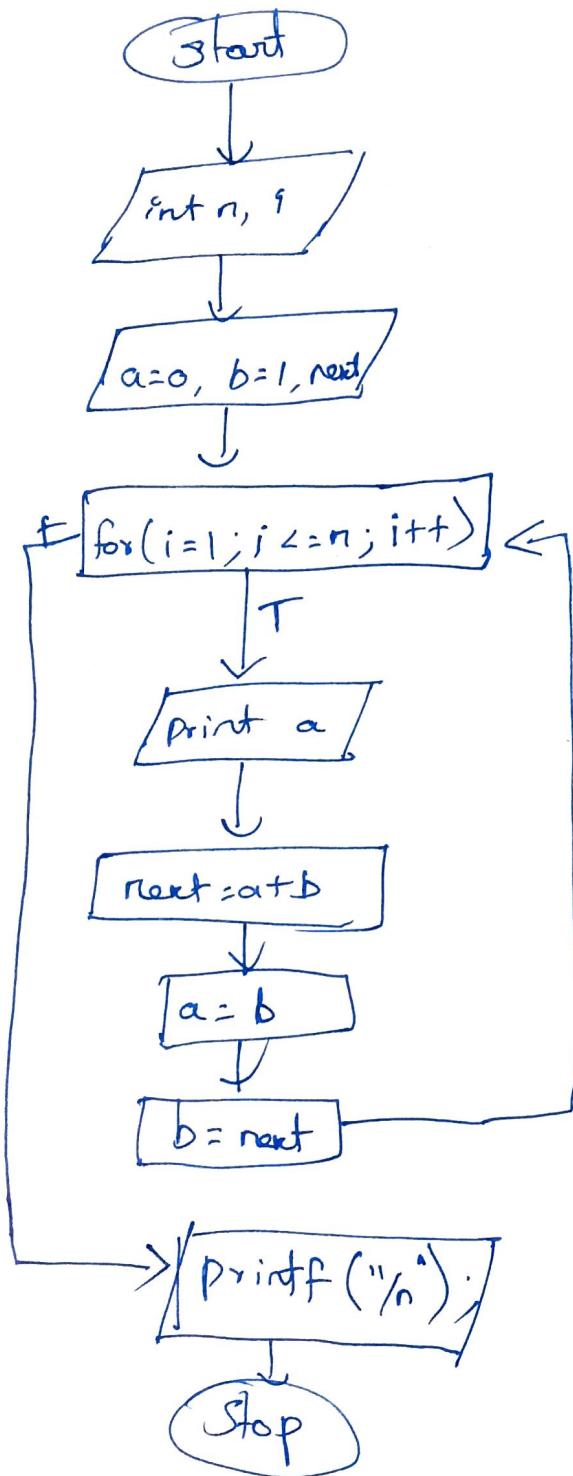


Algorithm:

- 1.) Start
- 2.) Input The number num
- 3.) Initialize reversed = 0
- 4.) Repeat while num not = 0
 remainder = num % 10
 reversed = reversed * 10 + remainder
 num = num / 10
- 5.) reversed output
- 6.) Stop

40) draw a flow chart to print fibonacci series;

Algorithm



1) Start

2) Input n (number of terms)

3) Set a=0, b=1, i=1

4) Repeat while i <= n:

 print a

 next = a+b

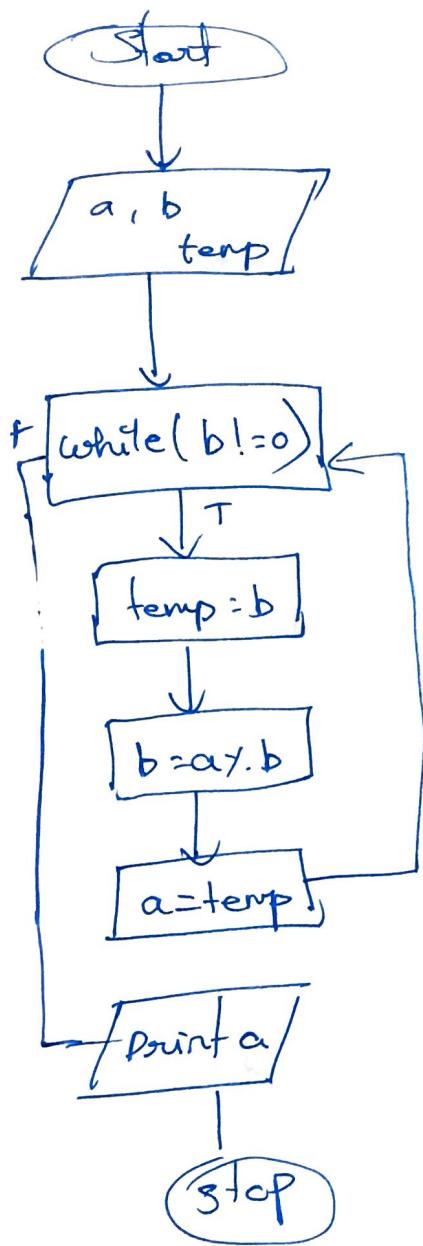
 a = b

 b = next

 i = i + 1

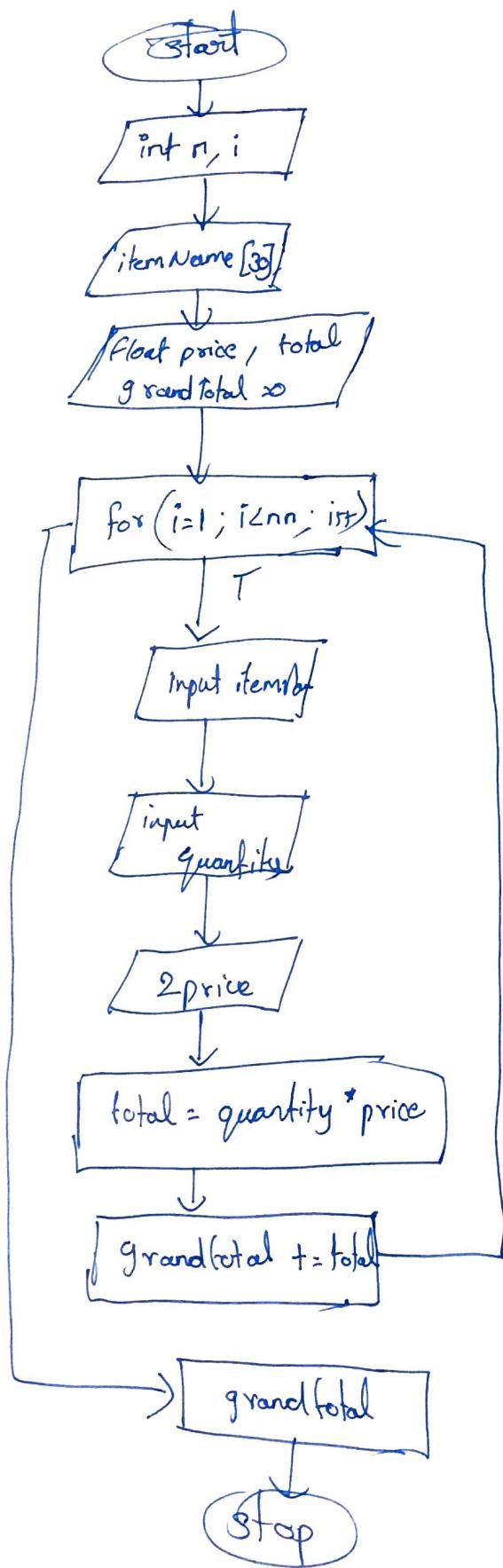
5) Stop

4) draw a flow chart to find GCD of two given numbers



- 1) start
- 2) Input two numbers a and b
- 3) Repeat while $b \neq 0$
 - $\text{temp} = b$
 - $b = a \% b$
 - $a = \text{temp}$
- 4) when loop ends, a holds the GCD
- 5) output a
- 6) stop

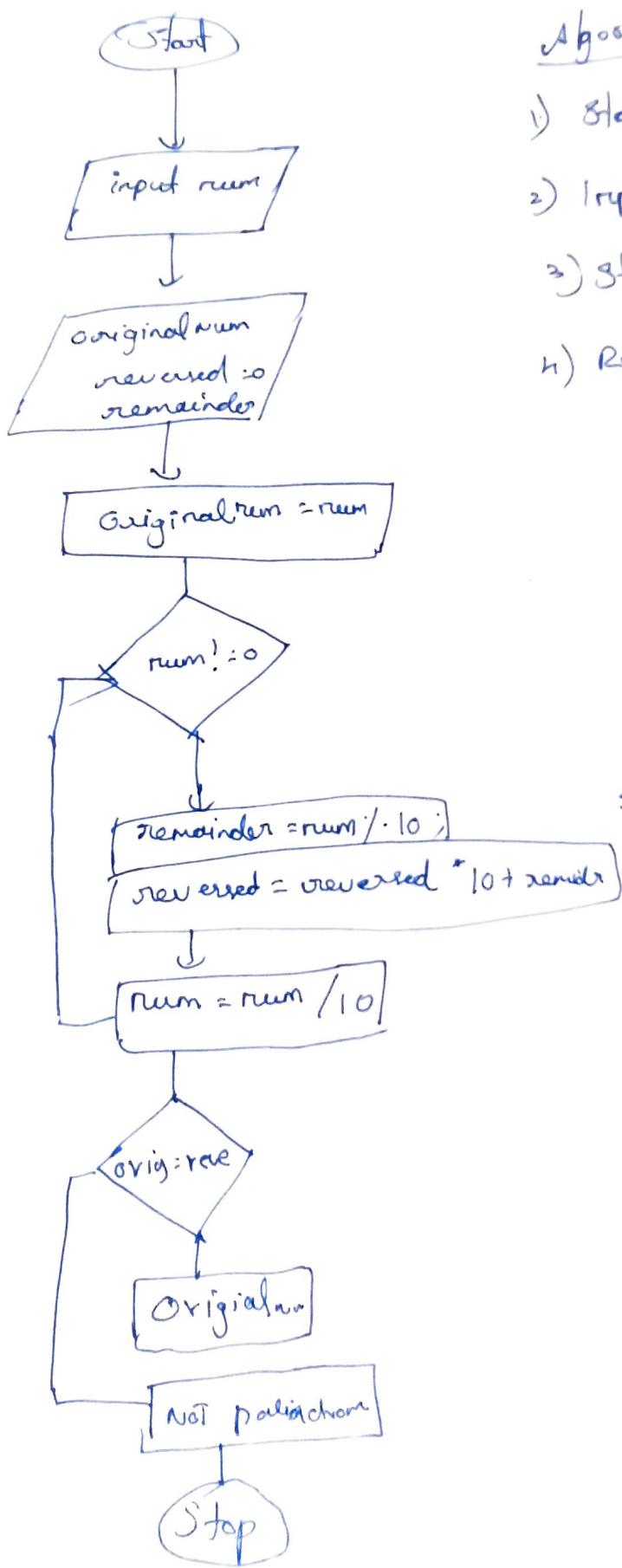
h2) draw a flow chart to print bill of shopping items, quantity. unit price to be entered.



Algorithm :-

- 1) Start
- 2) Input number of items
- 3) Initialize grandTotal
- 4) Loop from 1 to n
 - Input item name
 - Input quantity and unit price
 - calculate total = quantity * price
 - Add to grand total
- 5) Print each item with quantity * price
- 6) Print grand total
- 7) End.

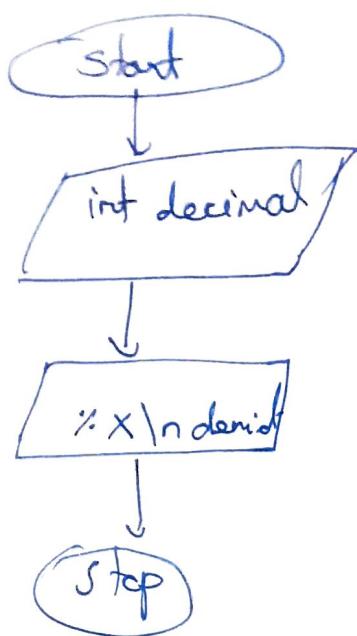
4) draw a flow chart to print check whether the given number is palindrome or not



Algorithm

- 1) Start
- 2) Input the number num
- 3) Store originalnum = num
- 4) Reverse the number
reversed = 0
while num ≠ 0
 remainder = num % 10
 reversed = reversed * 10 + remainder
 num = num / 10
- 5) if originalnum = reversed
 its a palindrome
- 6) else not a palindrome
- 7) stop

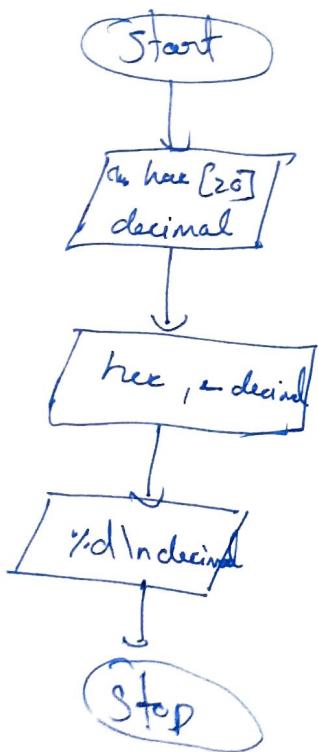
draw a flow chart to print decimal value of given decimal no.



Algorithm:

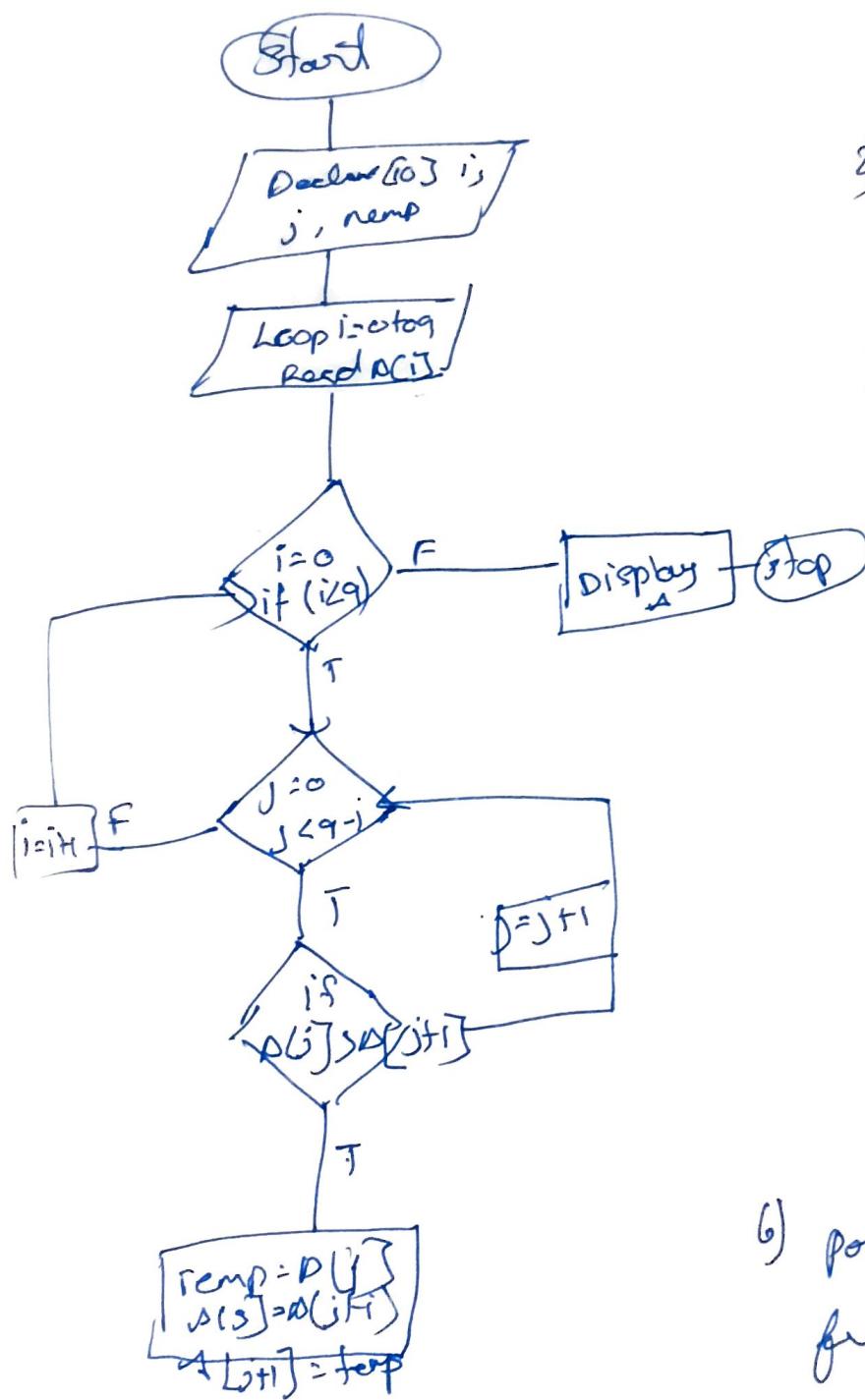
- 1) Start
- 2) Input a decimal number
- 3) use `%x` format specifier
Print the hexadec
Value
- 4) Stop

Q5) draw a flow chart to print decimal value of hex number :-



- 1) Start
- 2) Input hexadecimal number or string (to handle A-F)
- 3) use `scanf(hex, "%x", &decimal)` to convert
- 4) Output decimal value
- 5) Stop

46) draw a flow chart to get 10 numbers from the user and print the ascending order $A[0:9]$.



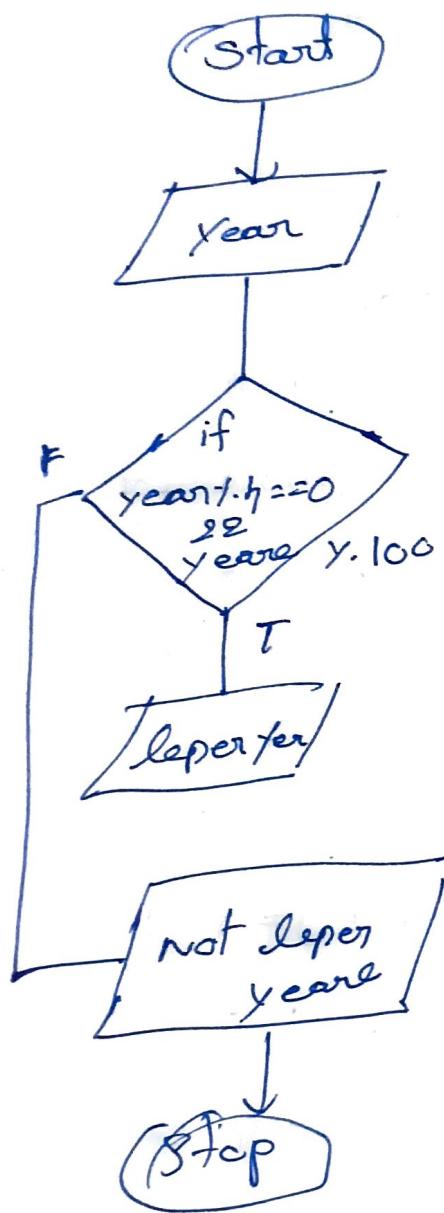
- 1) start
- 2) Declares an array $A[10]$
- 3) set $i=0$
- 4) Repeat while $i \leq 10$
 - Input
 - Increment
- 5) sort the array A in ascending order (using bubble sort)


```

for i=0 to 9
  for j = i+1 to 9
    if A[i] > A[j]
      swap [i] and A[j]
      
```

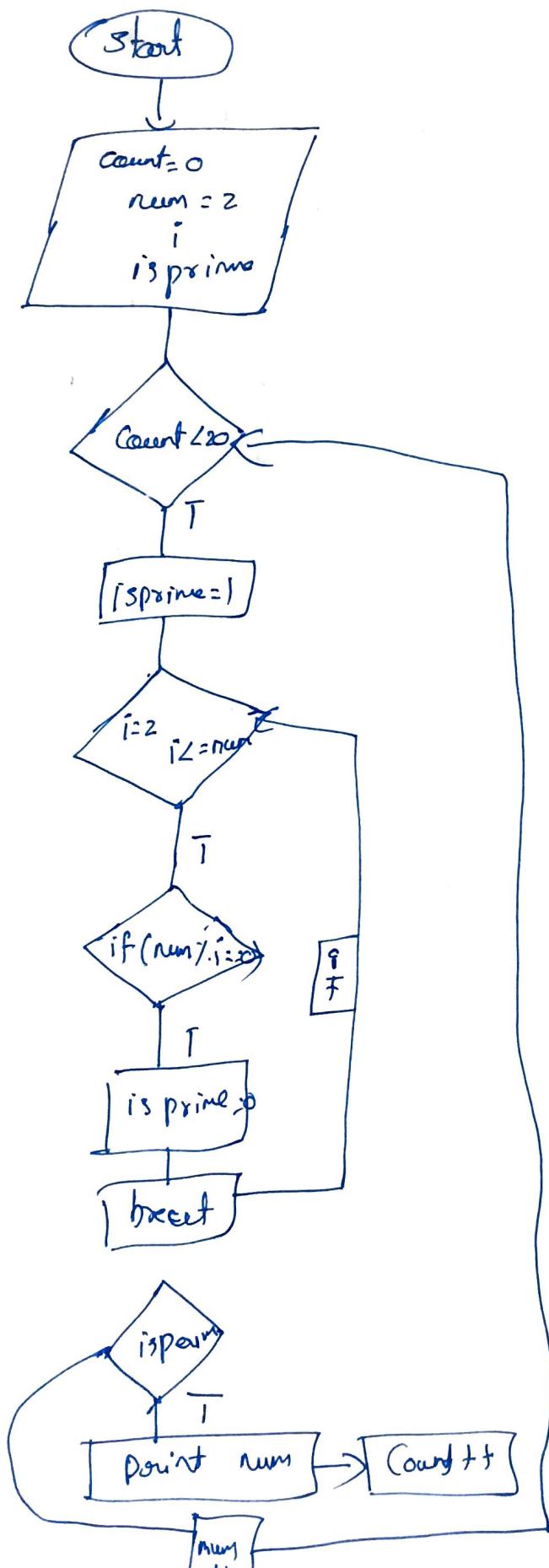
- 6) point all elements of A from index 0 to 9
- 7) print $A[0]$ (smallest number)
- 8) stop.

47) draw a flow chart if a given year is a leap year or not.



- 1) start
- 2) input year
- 3) if year%4==0 and
year%100!=0 \rightarrow leap year
- 4) Else if year%400==0
- 5) Else \rightarrow Not a Leap year
- 6) output the result
- 7) End.

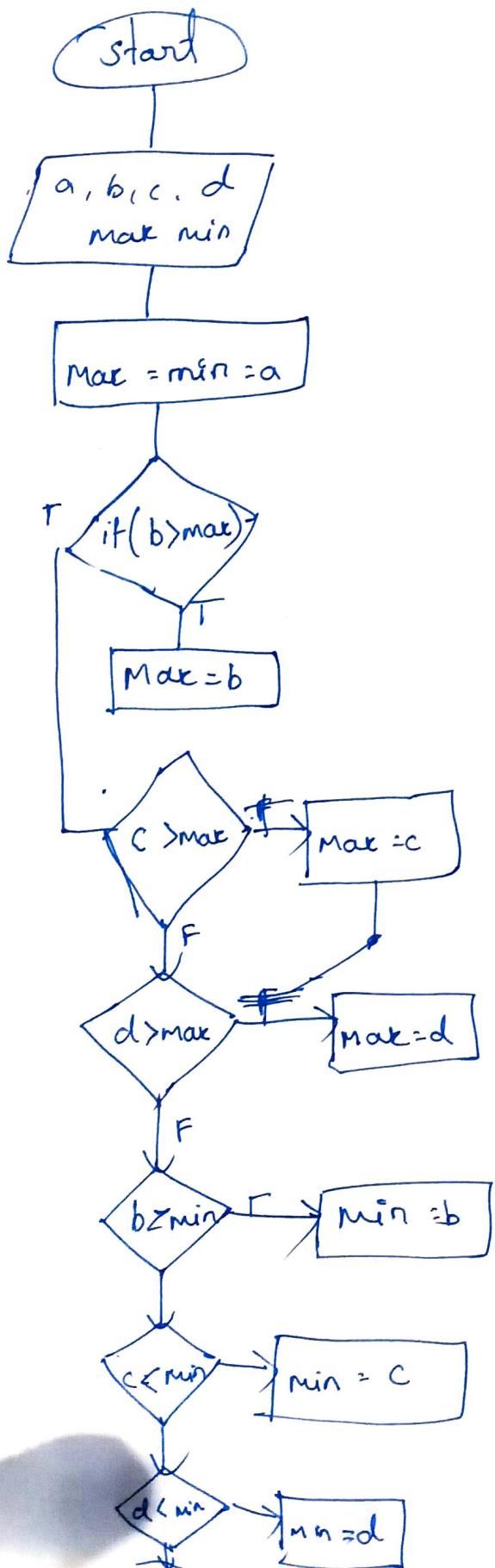
Q8) draw a flow chart to print the first 20 prime numbers :-



Algorithm :-

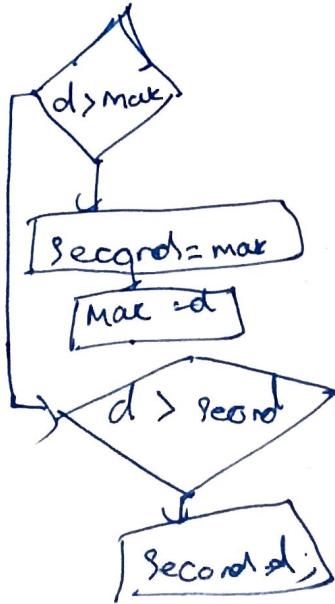
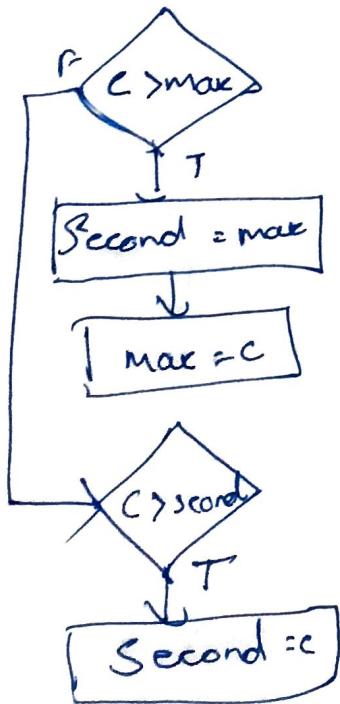
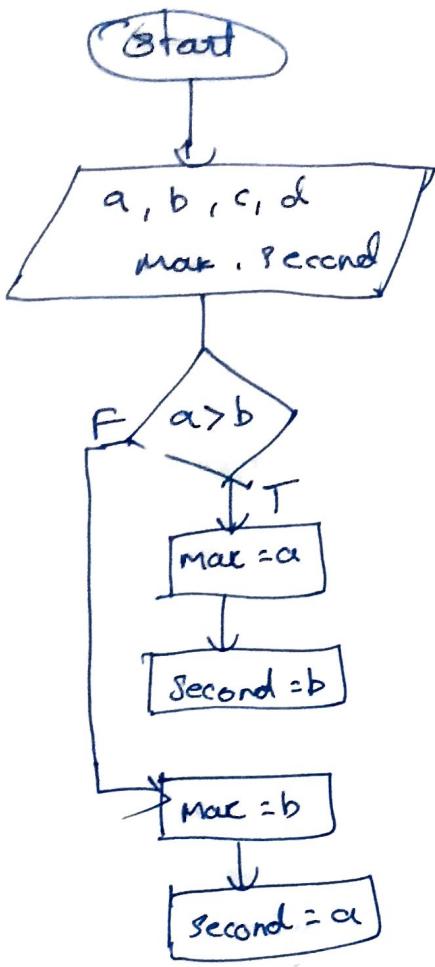
- 1) start
- 2) set count = 0, num = 2
- 3) while count < 20
 - set isprime = 1
 - for i = 2 to num / 2
 - if num % i == 0
 - set isprime = 0
 - break
 - if isprime == 1
 - print num
 - increment count
- 4) increment count num
- 5) stop

h9) Draw a flow chart to find the largest among four given numbers



- 1) Start
- 2) Input 4 numbers a, b, c, d
- 3) Set max = a, min = a
- 4) if b > max \rightarrow max = b
 if c > max \rightarrow max = c
 if d > max \rightarrow max = d
- 5) if b < min \rightarrow min = b
 if c < min \rightarrow min = c
 if d < min \rightarrow min = d
- 6) output max and min
- 7) Stop

50) draw a flow chart to print second largest number among four numbers :-



Algorithm

Initialize two variables

max \leftarrow first number

second \leftarrow second number

compare max and second

if second $>$ max, swap them

check third number (c)

if $c > \text{max}$

second \leftarrow max

max \leftarrow c

else if $c > \text{second}$

\rightarrow second \leftarrow c

check fourth number (d)

if $d > \text{max}$

second \leftarrow max

max \leftarrow d

else if $d > \text{second}$

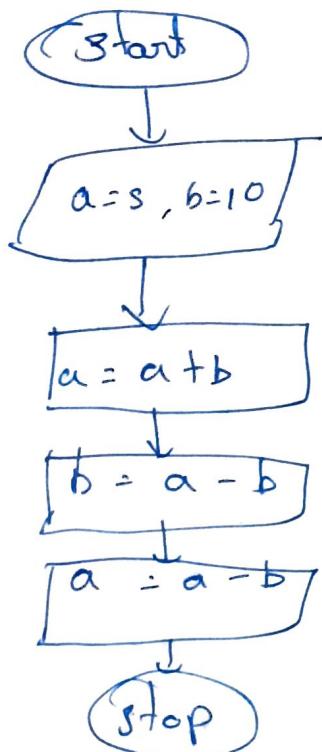
second \leftarrow d

output - second

is the
second largest.

Stop

51) Draw a flow chart to swap two numbers without using temp variable.



Read a and b

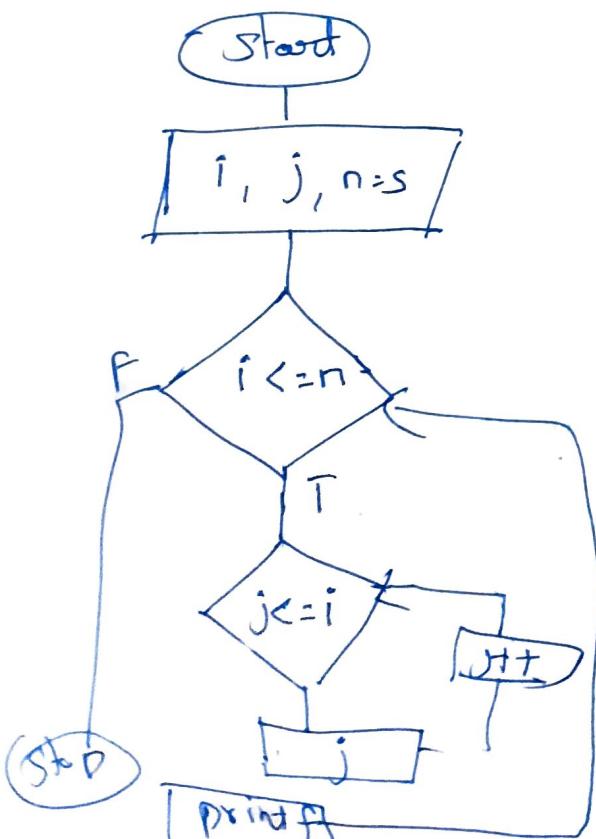
$$a \leftarrow a + b$$

$$b \leftarrow a - b$$

$$a \leftarrow a - b$$

point a and b

52) Draw a flow chart to print given pattern:-



start

Read the value of n

Set outer loop counter $i=1$ for

3.1 For each i, do

Set inner loop counter $j=1$
print value of j

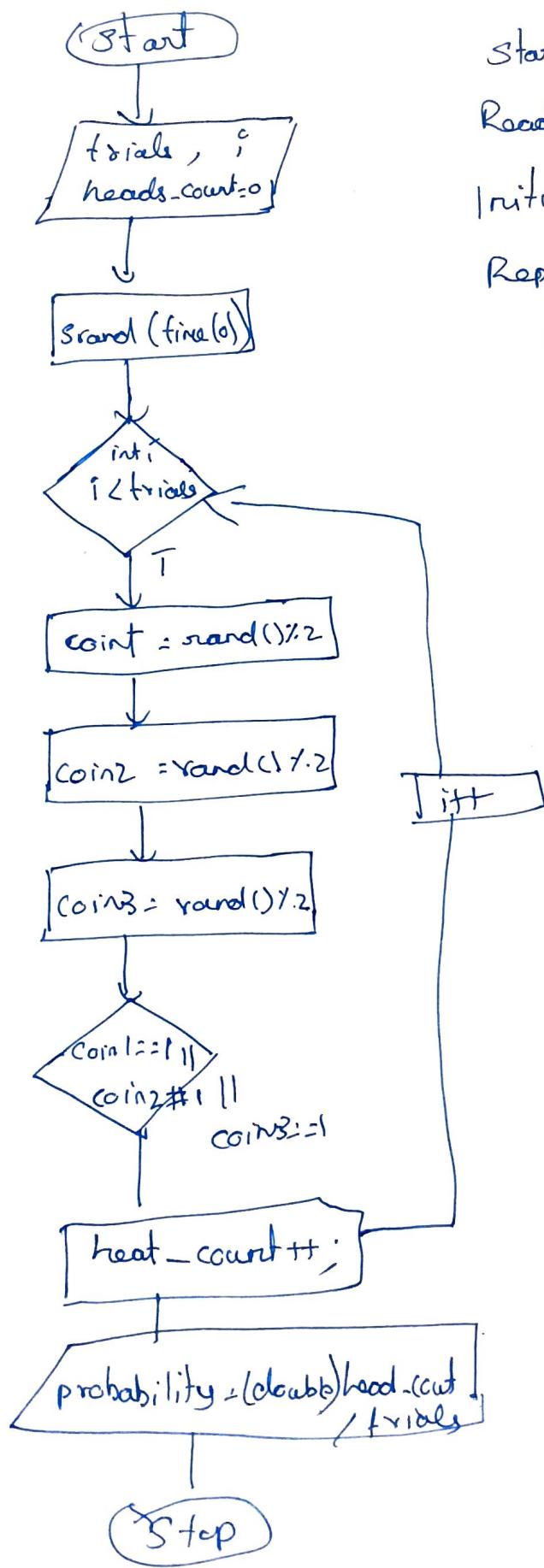
End inner loop

Point a new line

End outer loop

stop

53) Draw a flow chart to find probability of finding head when 3 coins tossed simultaneously.



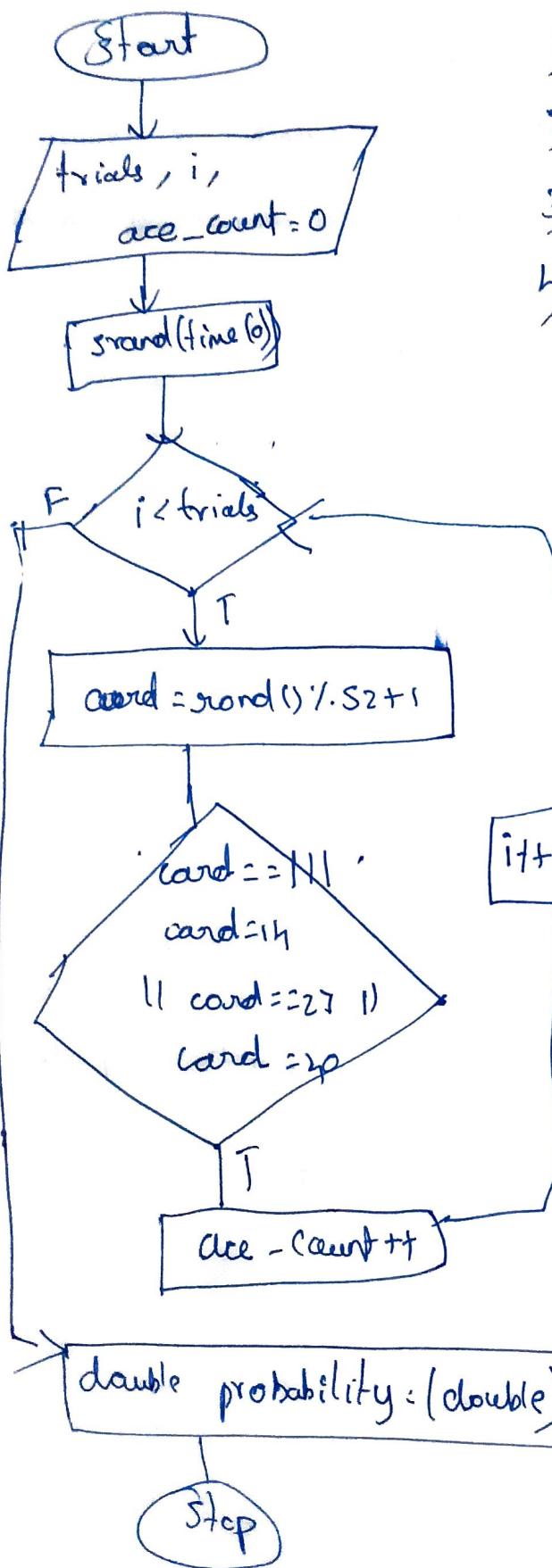
Start
 Read The number of trials N
 Initialize counter head-count=0
 Repeat $i=1$ to N

- toss 3 coins (generate 3 random numbers coin₁, coin₂, coin₃)
- Each coin is either 0 (Tail) or 1 (Head)
- if any one of coin₁, coin₂, or coin₃ is 1 Head

 if any one of
 After loop ends, calculate
 $\text{probability} = \text{head_count} / N$
 print the probability

stop.

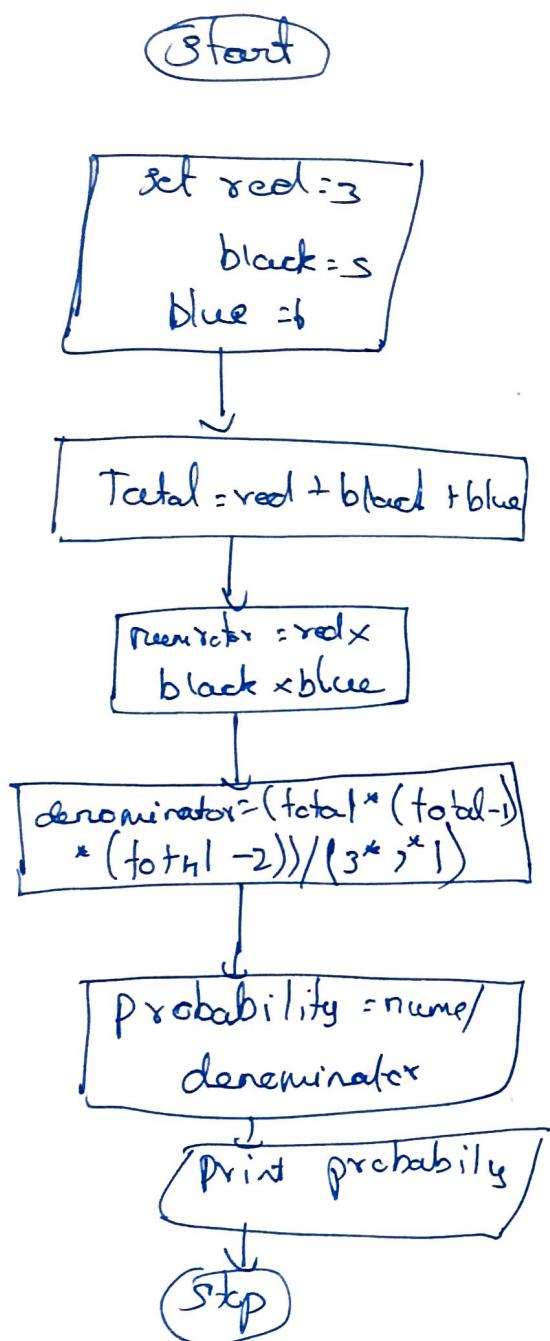
sh) Draw a flow chart to find probability of finding Ace in pack of cards.



- 1) Start
- 2) Read number of trials N
- 3) Set ace - count = 0
- 4) Repeat $i = 1$ to N
Randomly Select a card number from 1 to 52
if the card number is 1, 14, 27, or 20 (ace of Spades, Hearts, Diamonds, Clubs)
Increment ace-count

- 5) After all trials, calculate
- 6) Print the probability
- 7) Stop

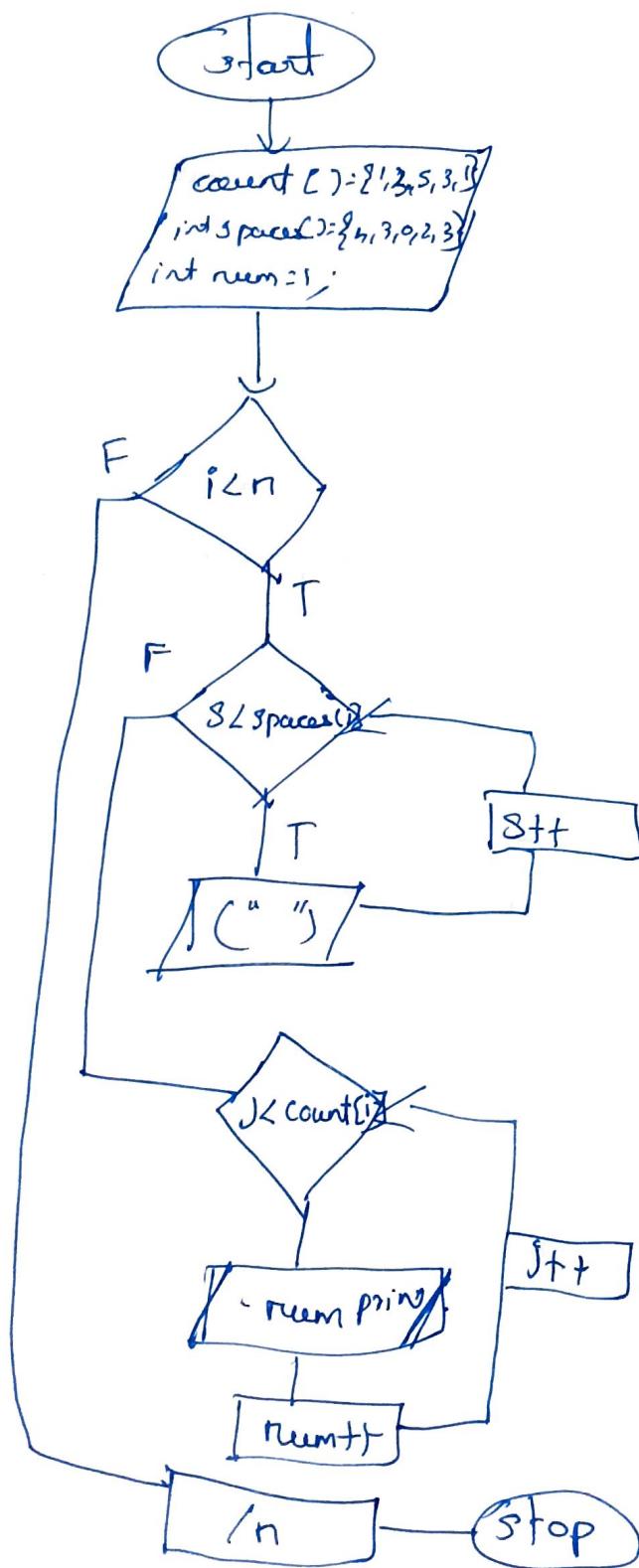
ss) In a box we have 3 red, 5 black, 6 blue balls if we draw 3 balls simultaneously.



- * Start
- * Initialize values
- * calculate numerator
 - * numerator = red * black *
 - blue = 3 * 5 * 6 = 90
- * denominator
 - * denominator = $(14 \times 13 \times 12) / (3 \times 2 \times 1) = 364$
- * probability = numerator / denominator
- * Print
- * Stop

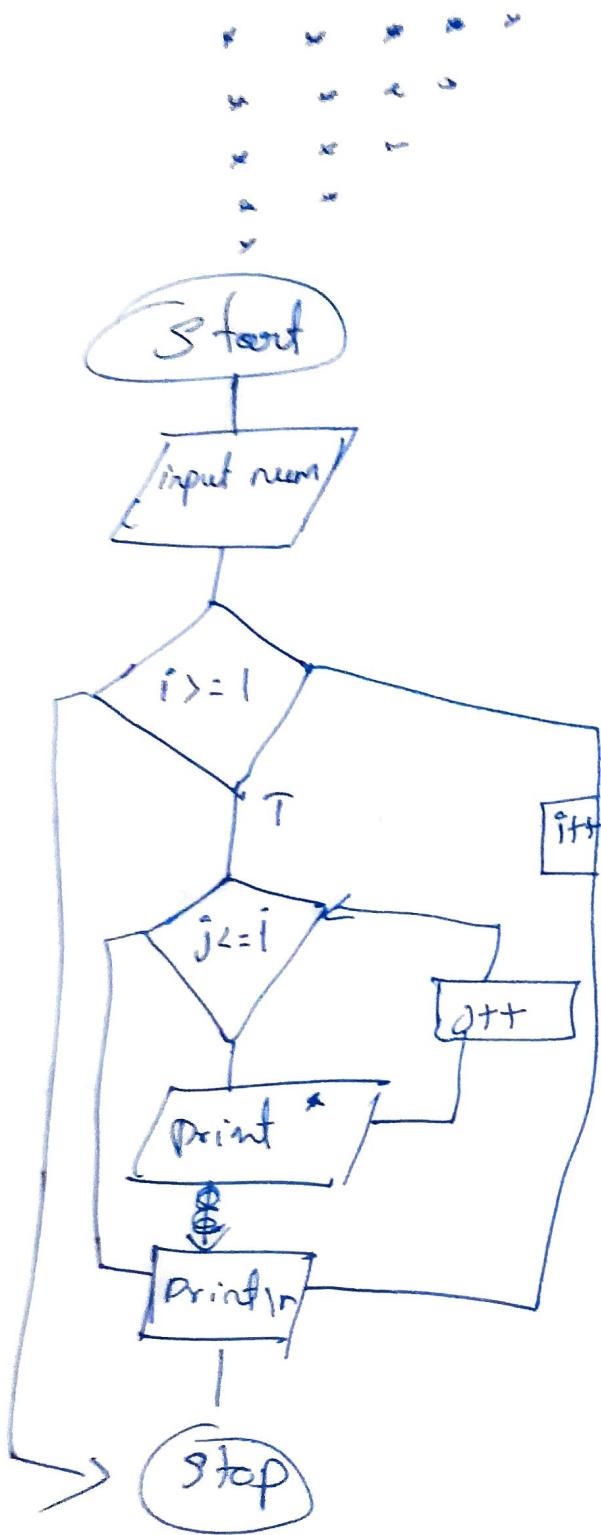
56) Draw a flow chart to print given pattern

1
2 3 4
5 6 7 8 9
10 11 12
13



- Start
- Declare an array count $[] = \{1, 3, 5, 3, 1\} \rightarrow$ number of elements in each row
- Declare an array spaces $[] = \{2, 3, 0, 2, 3\} \rightarrow$ number of leading spaces before printing each row
- Initialize a variable num=1 to start the number sequence
 - a) Loop i from 0 to n (for s rows)
 - Loop s from 0 to spaces[i]-1
 - print two spaces " " to align numbers correctly
 - Loop j from 0 to count[i]-1;
 - print the current number from width 2
 - Increment num by 1
 - print a new line.
 - * Stop

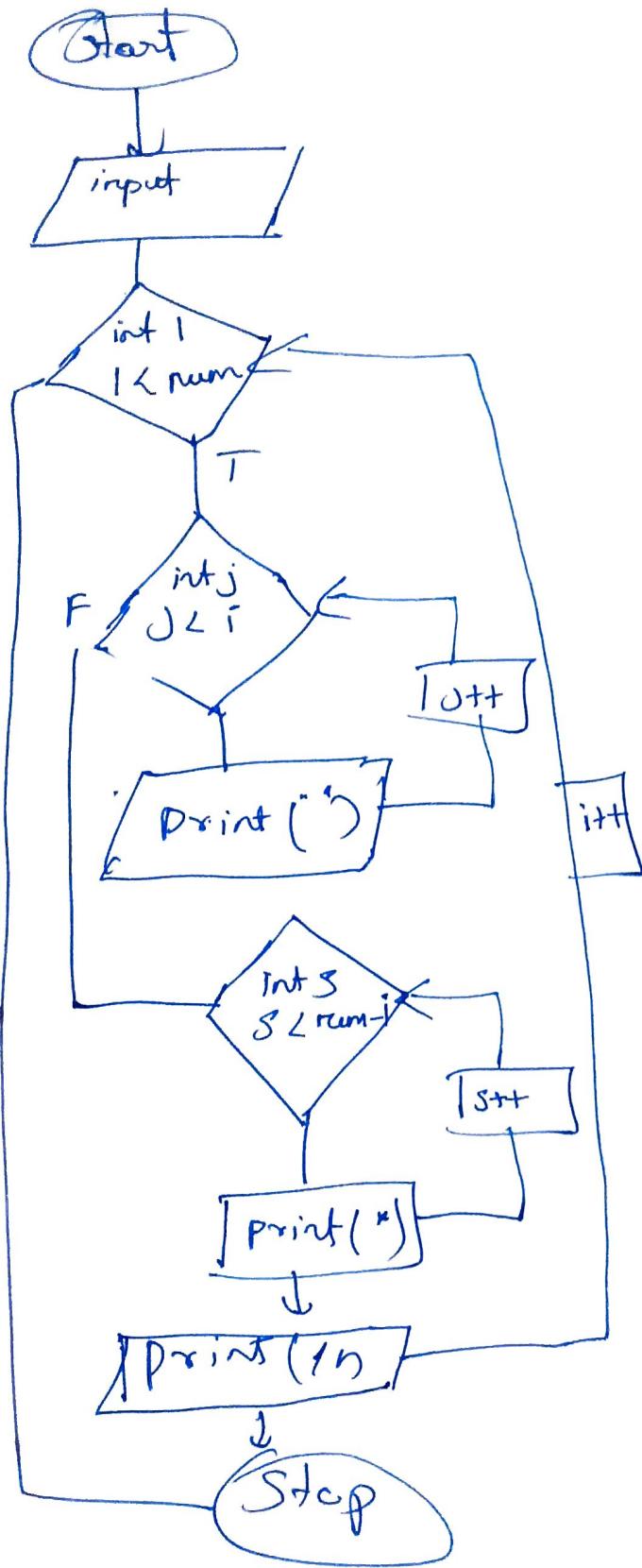
57) Draw a flow chart to print given pattern.



Algorithm

- * Start
- * Read number of rows
- * Loop i from rows down to 1
 - * Loop j from 1 to i
 - * print '*' (standard 8 spaces)
 - * print newline
 - * End

58) Draw a flow chart to given pattern



Algorithm

start

 Read number of rows

 Loop if row >= 0 do

 Loop j from 0 to i

 print (space)

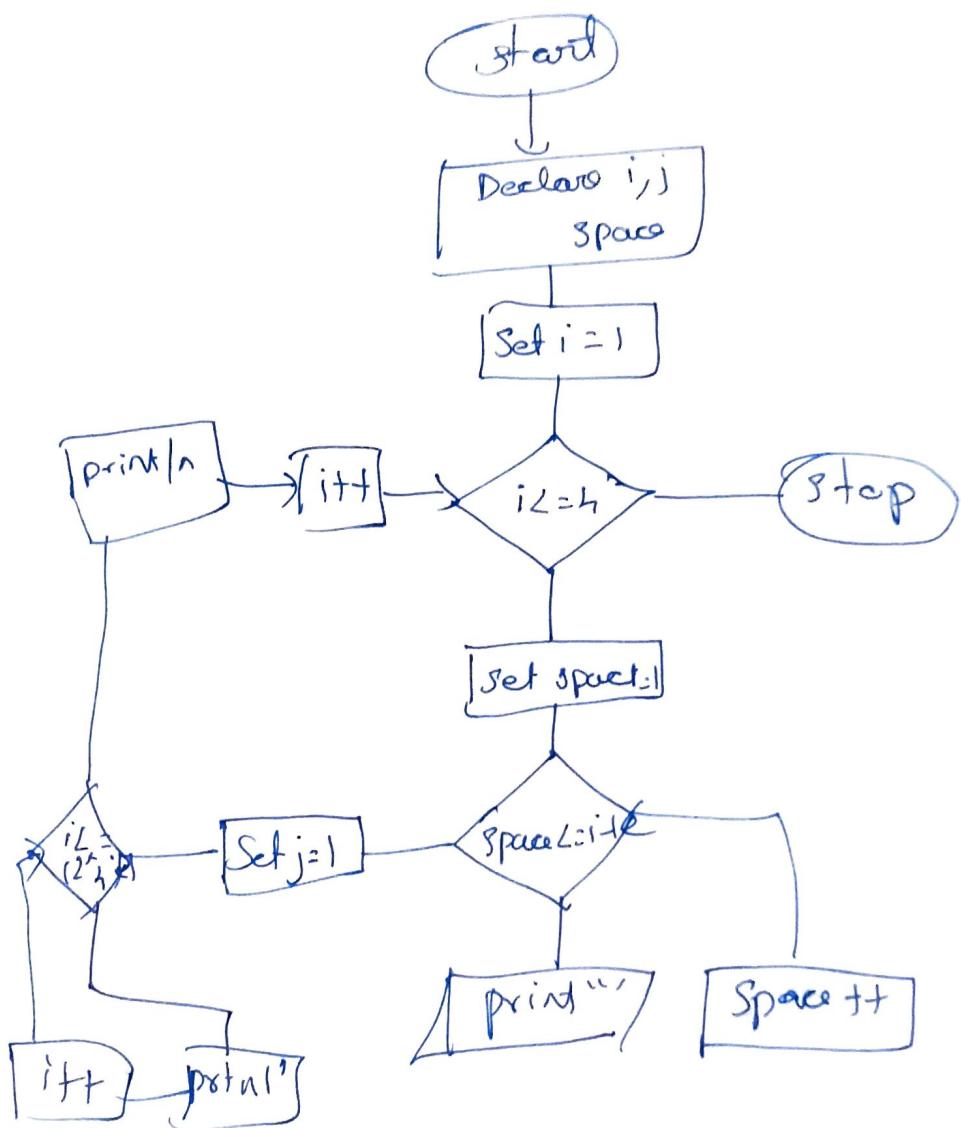
 Loop s from B to n

 print ("")

 new line

 stop

5a) Draw a flow chart to print given pattern



1) start

2) Set rows - 4 (you can change this for a larger pyramid)

3) Loop i from i to rows (each i is one row);

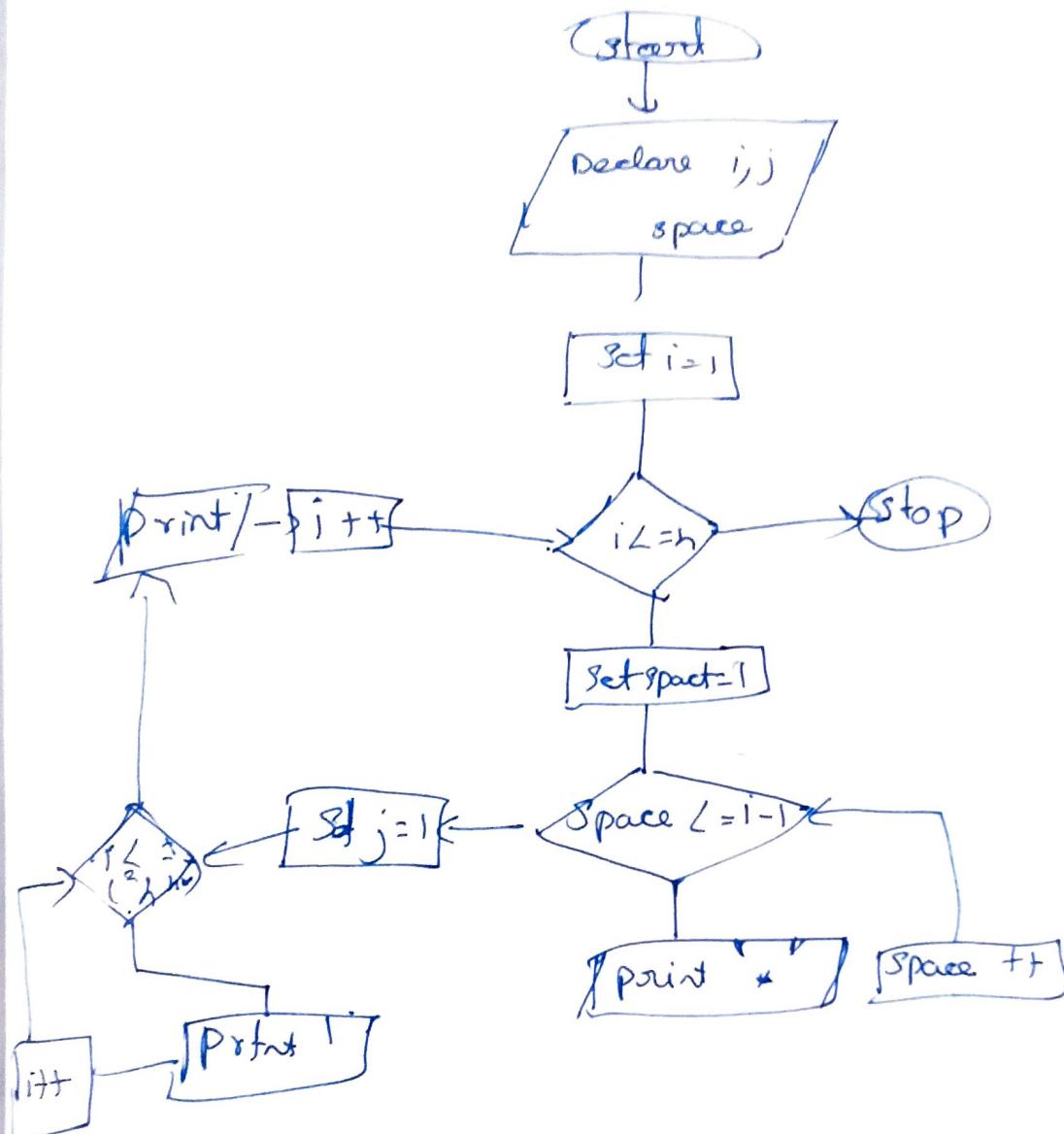
a) Loop Pspace from 1 to rows - i

print 3 Spaces " " to align start

b) Loop star from 1 to 2*i - 1

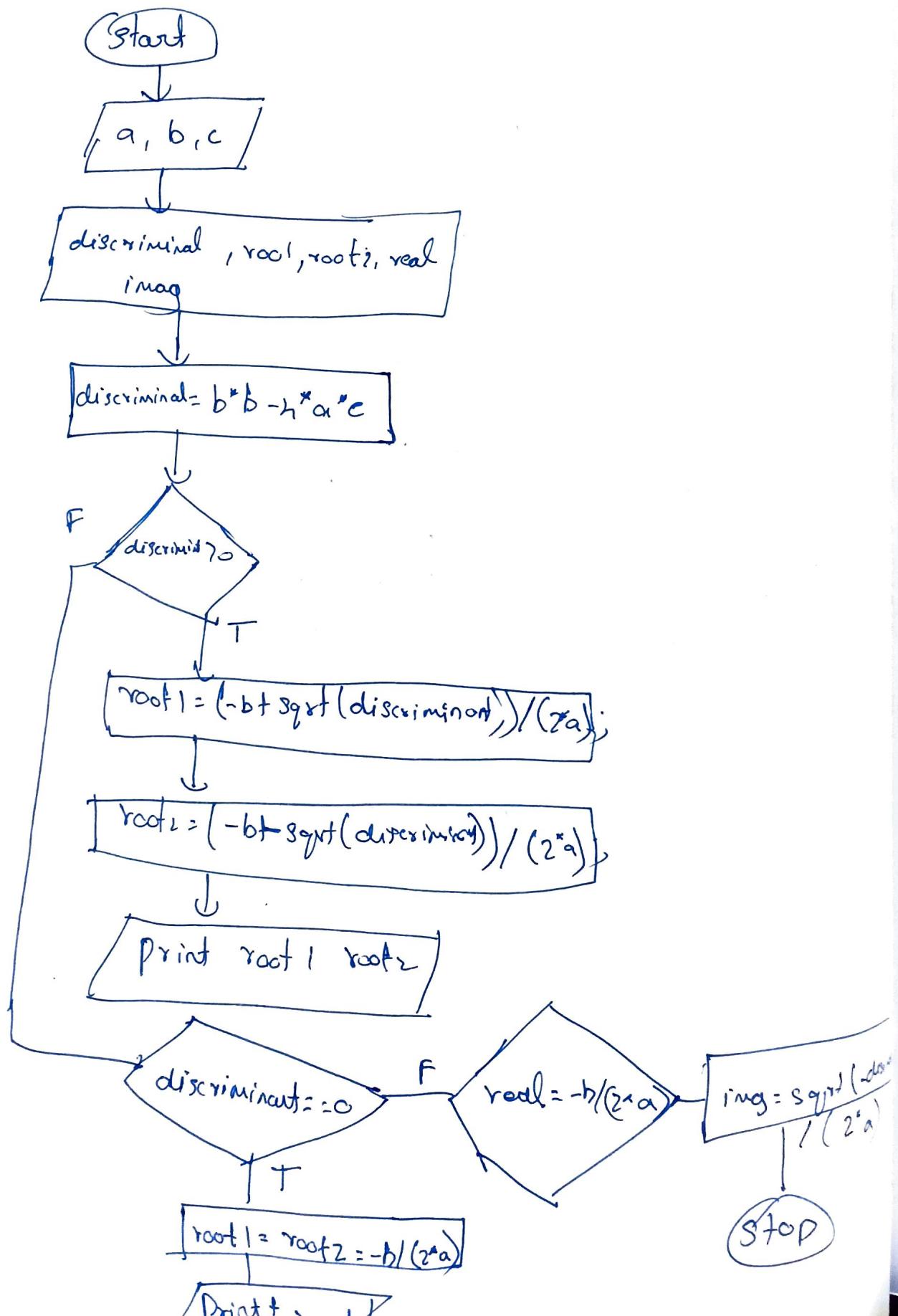
4) Stop

b) Draw a flow chart to print given pattern

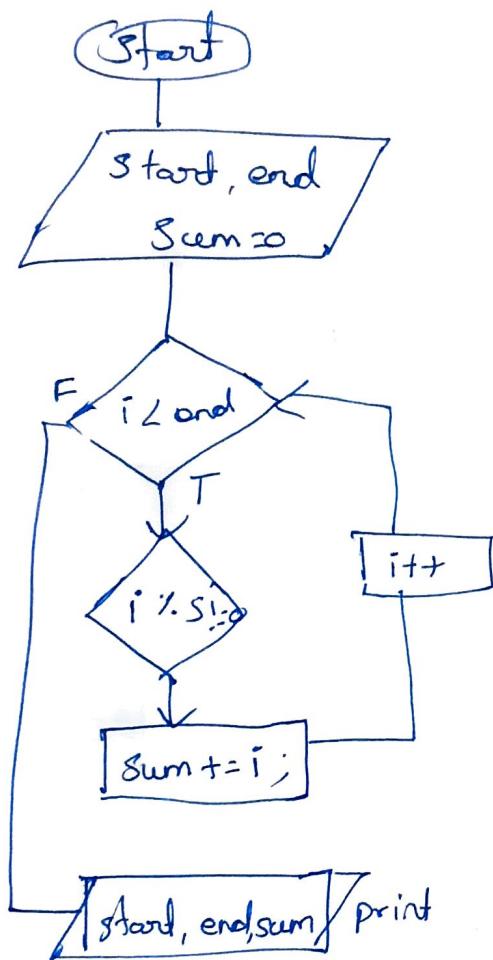


- 1) Start
- 2) set rows - 3 | you can change this for a larger program
- 3) Loop i from 0 to rows (each i is one row)
 - a) Loop f Space from 1 to rows-i
print 3 spaces "x" to align star
 - b) Loop star from 1 to $2^*i - 1$
- 4) Stop

61) Draw a flow chart to print given up roots of a quadratic equation!



62) take two numbers from the user and flow chart to find the sum of numbers which are not divisible by 5 between the range

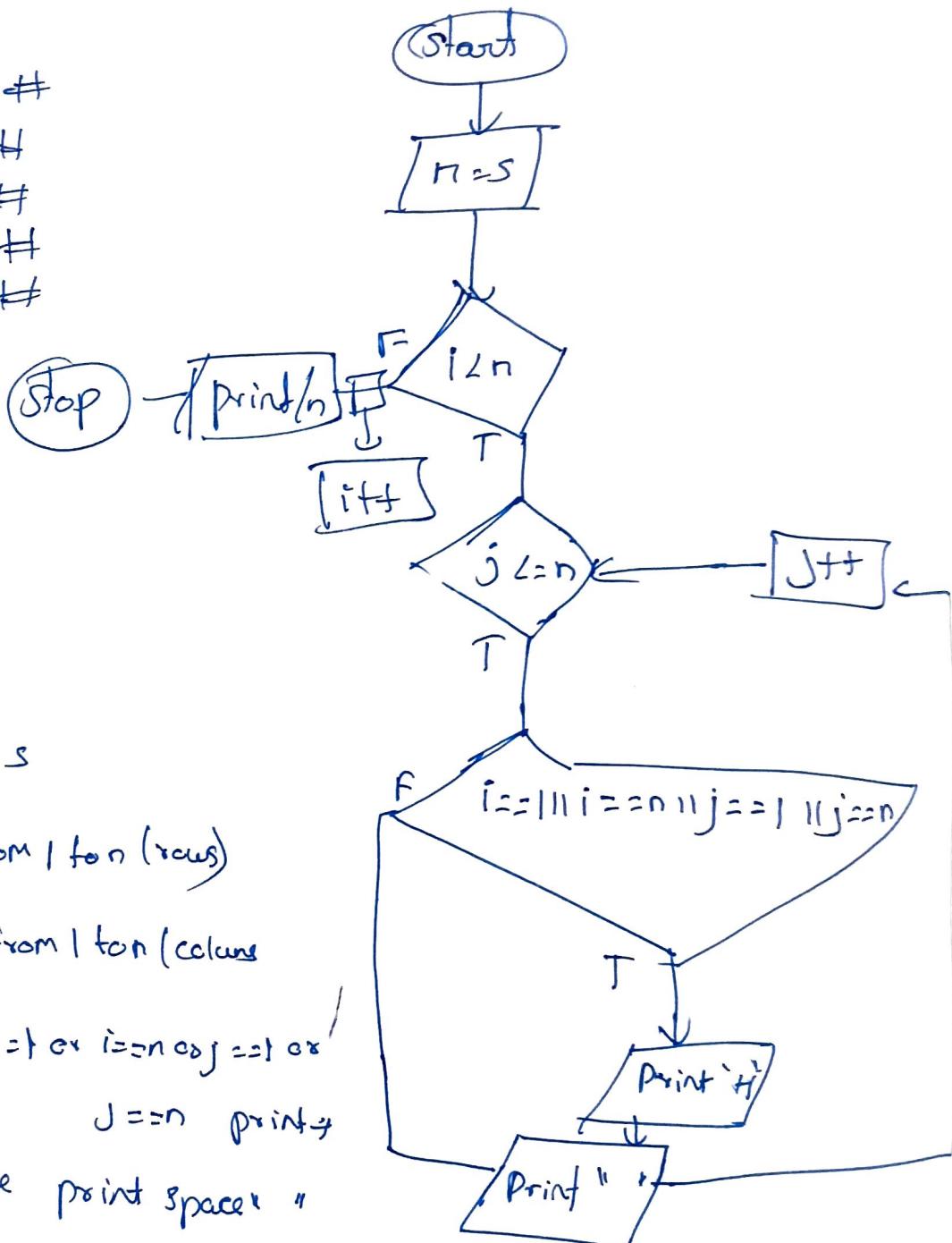


Algorithm:-

- Start
- Input two numbers start, end
- Initialize sum $\leftarrow 0$
- Loop from $i = start + 1$ to $end - 1$
 - if $i \% 5 \neq 0$
 - Add i to sum
- Print sum

63) Draw a flow chart to print given pattern
Hollow Square:

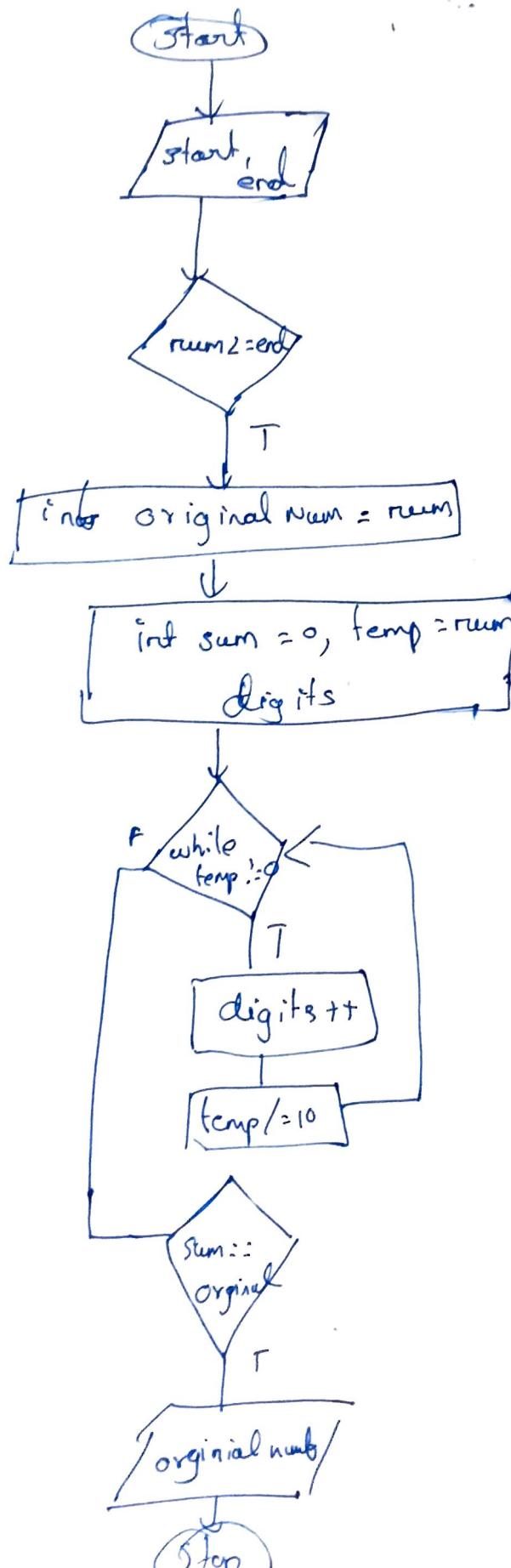
```
# # # #
#       #
#       #
#       #
# # # #
```



Algorithm :

- 1) Start
- 2) Set $n \leftarrow s$
- 3) Loop i from 1 to n (rows)
 - 4) Loop j from 1 to n (columns)
 - If $i == 1$ or $i == n$ or $j == 1$ or $j == n$ print '*'
else print space "
 - print newline
 - 5) Stop

6n) Draw a flow chart to find Armstrong numbers between two intervals.



start the process

Read a, b

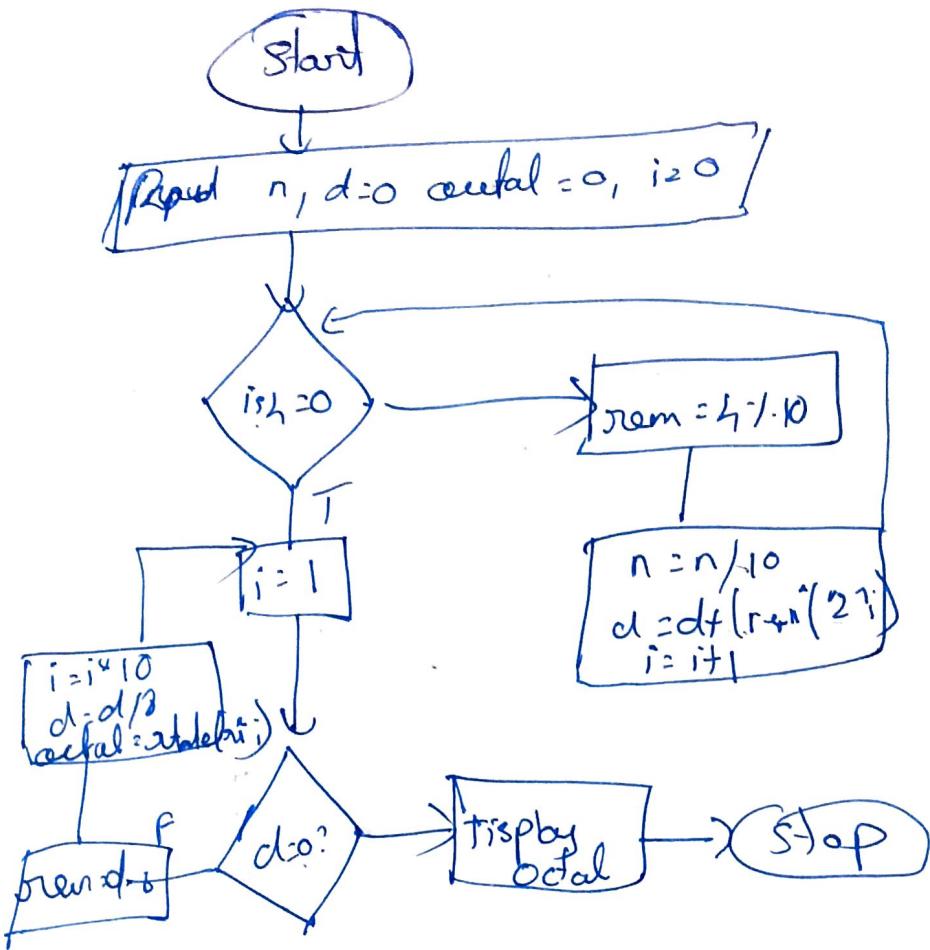
pointer for loop with condition $a \leq b$ the loop has one inner while loop

The when loop control $a \neq 0$
here comp

$$r = a \% 10$$
$$sum = sum * 10 + r$$
$$a = a / 10$$

stop

6s) Draw a flow chart to convert binary number to octal

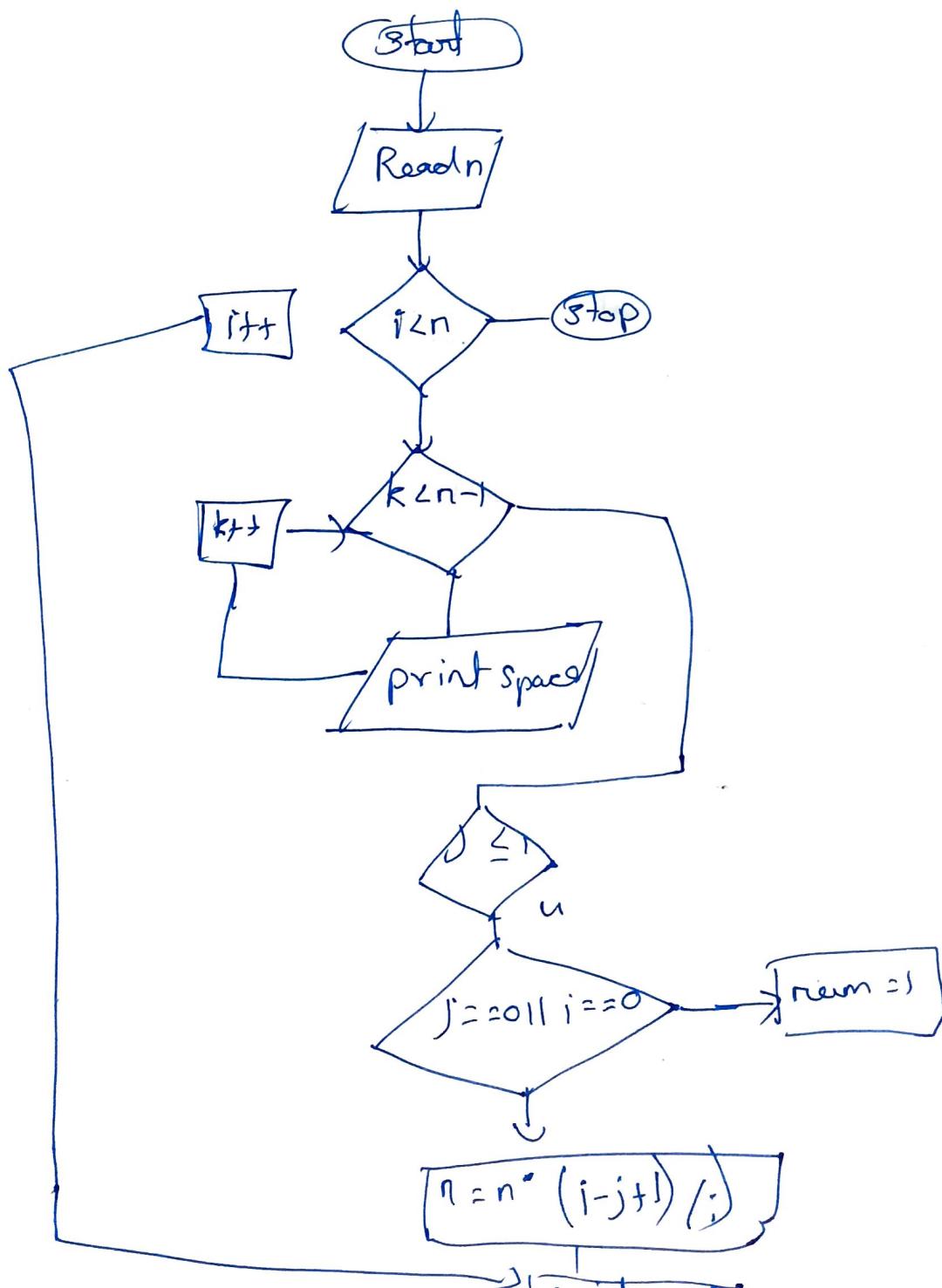


- 1) Start
- 2) input binary no
- 3) binary to decimal a for each bits
- 4) Decimal to octal by dividing by 8 and
- 5) collect reminders
- 6) print octal no
- 7) stop

67) Draw a flow chart to print given pattern

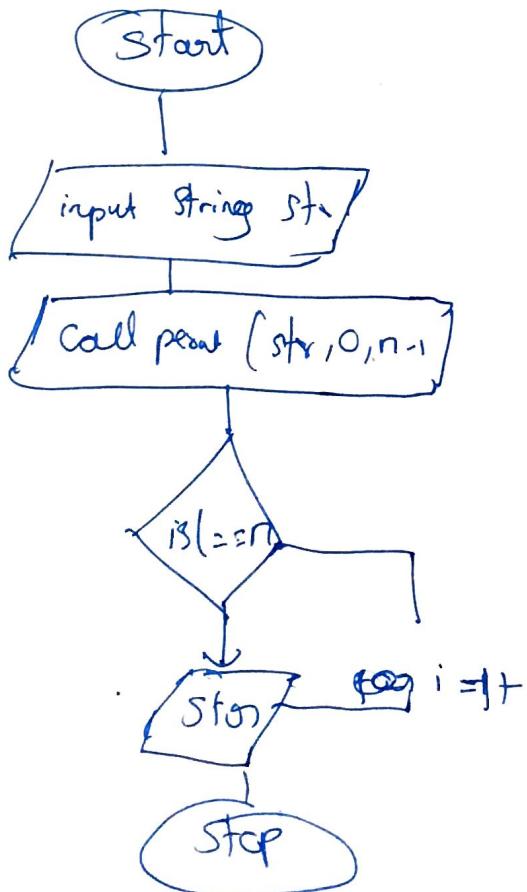
Dascal triangle

```
    1  
   1 1  
  1 2 2 1  
 1 3 4 3 1  
1 4 7 7 4 1
```



- Start
- Read n from user
- Declas loop with condition ($i \leq n$)
- until $i < n$ this loop will run
- Print spaces using for loop with ($k \leq n-i$)
- print the number using for loop ($j \leq i$)
- compute number = $n^*(i-j+i)/j$
- print number
- stop

68) Draw a flow chart to take string from user and print all the combinations.



Algorithm

- Start
- Read input string
- permut (str, l=0, n= length-1)
- condition which is
- $L = :n$ point the string
- The enter for loop with
- $i = 1$ to \rightarrow Then $i+1$ swap
- Recurs
- Stop

b7) Draw a flow chart to print pattern

1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4

