

28/11

QEL

Java version -> 1.1

python -> object & function oriented

in industrail version -> 1.8 -> possible interface contract concrete method

interface Demo1

Java -> function oriented code

object oriented

Java supports non portability

{
void test();}

{
class Demo2 implements Demo1

{
public void test()
{

s.o.p ("Babbar is smiling");

}
System.out.println("Smiling");

{
Demo2 d1 = new Demo2();
d1.test
}

prog:- package com.issdekar.P;
interface Demo

{
void void test();

{
class Demo2 implements

(System.out.println("Smiling"));
ARUN'S PAGE NO.

1

demo di = () ->

s.o.p ("test implemented in lambda expression");

) :

di. test();

}

)

interface name def = (arguments) -> { } ;

Functional Interface

- * an interface which has @ the max one abstract method is known as functional interface
- * lambda expression works only for functional interfaces

non portable proj :- package com. isipworld. p1

interface demo

{

public void test (int a);

)

→ default void def();

private class mainclass3

{

psm (String[] args)

{

demo. di = (int a) ->

)

s.o.p ("test implemented in lambda expression" + a);

) :

d. test(56);

→ it's should pass Boolean Value

{

proj. class program

{

psm (String[] args)

)

boolean learning = false;

only

else

s.o.p (" went to office class
here ");

)

)

if (learning == true)

s.o.p (" learned ");

ARUN'S
PAGE NO.

DATE

for (initialization; condition; increment)

{
 }
 {
 }

prg1 :- class program

{

 random string on s

{

 int i=0;

 for (; ;)

{

 s.o.p ("Hello world!");

}

 // s.o.p ("main method ended");

}

prg1 :- class program

{

 num (char* ans)

{

 int p=0;

 for (; i<5 ; i++)

{

 s.o.p ("i=" + i);

}

// s.o.p ("main method ended");

}

{

1) write a program to divide 2 numbers without using division operator

prg1 :- class program 3

variable statement

function random (int div)

{
 int count=0;

 // write logic (ans = 2 div)
 // ans = num - div
 // count++

num = num - div;

return count

}

random (string)

2

s.o.p ("

0.8 10

)

2

1) final true

2) program

3) array

4) strings

prg1 :- print

class pr

{

 return

 std::endl +

 int a;

 a + "uc

)

 s.o.p

 a + "

 test

)

 PSV.R

 4

 s.o.p

 bc

 sc

)

return count;

}

PSVM C (string () exgs)

2

int res = division (2, 5);

S.O.P ("res = " + res);

3

4

1 find the remainder of two no without % operator

2 program to multiply two no without multiplication operator

1) patterns

decorater design.

2) numerical

3) array

4) strings

Decursion:-

3/5/19

problem :- printing a recursive loop
class program

loop recursive define

a

c

} to avoid infinite loop recursive define
is need deep member

int a=0

a=0

a<5

static void test ()

int a=0;

if (a<5)

l

S.O.P ("test ");

att : } → fact (c); } No fact

fact (c); } att ; }

l

2

PSVM (---)

4

Script "startic"

test (0);

S.O.P ("end");

l

l

prog1 :- c/c++s program 4

public static void test (int a)

if (a <= 5)

{ s.o.p ("test"); }

 a++;

 test (a);

}

 psum (- - -)

 s.o.p ("

 program starts");

 test (0);

 s.o.p ("program ends");

}

prog1 :- class program 4

{

public static void test (int a)

 }

 if (a <= 0)

 {

 s.o.p ("a");

 a++;

 test (a);

 }

 psum (- - -)

}

 s.o.p ("program starts");

 test (1);

 s.o.p ("program ends");

}

}

a <= 5
a = 1

test()

a <= 5
a = 2

test()

a <= 5
a = 3

test()

a <= 5
a = 4

test()

a <= 5
a = 5

test()

a <= 5
a = 6

test()

a <= 5
a = 7

test()

a <= 5
a = 8

test()

a <= 5
a = 9

test()

a <= 5
a = 10

test()

a <= 5
a = 11

test()

a <= 5
a = 12

test()

a <= 5
a = 13

test()

a <= 5
a = 14

test()

a <= 5
a = 15

test()

a <= 5
a = 16

test()

a <= 5
a = 17

test()

a <= 5
a = 18

test()

a <= 5
a = 19

test()

a <= 5
a = 20

test()

a <= 5
a = 21

test()

a <= 5
a = 22

test()

a <= 5
a = 23

test()

a <= 5
a = 24

test()

a <= 5
a = 25

test()

test
test

test +
test

prog1 :- c/c++s

3
public s

2
if (a <

5
s.o.p (a

u - ;
test (

3
2
Public st

1
if (c <

2
s.o.p (

at t
test (

3
2
prog (

2
s.o.p (

test (

prog1:- class program1

in single method

public static void test1 (int a) → public static void test3

(int a, int b)

if (a>=1)

System.out.println

a+ " "

a--;

test1(a)

}

public static void test (int a)

if (a>=9)

System.out.println

a+ " "

a++;

test2(a);

}

return (- - -)

2

System.out.println ("programmer111");

test1(9);

test2(2);

System.out.println ("programmer Sodha");

}

1 write a program to print the following pattern

out for loop → printing rows decide

inner for loop → → ← column ←

prog1:- class program1

3

return (-)

{

for (int row = 0; row < 8; row++)

→ make 10

→ by fix

for (int col = 1; col < 3; col++)

→ make 5

2

200030 023

row 1

1 < 3

S.O. P($\text{col} + " "$);	$\text{col} = 1$	$1 \leq 3$	$\text{col} = 1$	$1 \leq 3$
$\}$	$\text{col} = 2$	$2 \leq 3$	$\text{col} = 2$	$2 \leq 3$
S.O. P($\text{col} + " "$);	$\text{col} = 3$	$3 \leq 3$	$\text{col} = 3$	$2 \leq 3$
$\}$	$\text{col} = 4$			
	$\text{row} 03 : 2 \leq 3$		$\text{row} 03 : 3 \leq 3$	

Prgrm :-

1 1	1 1
2 2	2 2
3 3	3 3

class program

{ psum(--) }

{ }

for (int row = 0 ; row <= 3 ; row++)

{ }

for (int col = 1 ; col <= 4 ; col++)

{ }

S.O. P($\text{col} + " "$);

col++;

{ }

S.O.P(02());

class program

{ }

psum(--)

{ }

for (int row = 1 ; row <= 3 ; row++)

{ }

for (int col = 1 ; col <= 4 ; col++)

{ }

S.O.P($\text{row} + " "$);

{ }

S.O.P(01());

{ }

{ }

Prog :-

5 4 3

5 4 3

class program

{ }

psum(--)

{ }

for (int row = 1 ; row <= 2 ; row++) { }

{ }

for (int col = 5 ; col >= 3 ; col--) { }

{ }

S.O.P($\text{col} + " "$);~~██████████~~

S.O.P(03());

asym diff

12 3 4

12 3 4

part

pre - program

class prog

Prgrm - class

{ }

PSUM (med)

{ }

for (int

{ }

for (int

{ }

Pf (row)

{ }

S.O.P(

size

{ }

S.O.P(

{ }

{ }

2

{ }

Prgrm :-

class

{ }

psum(

{ }

69

{ }

12 34	44 44	1352	X X X X
12 34	33 33	1352	X X X X

pattern 9

~~program~~

~~class program~~

& has 1st priority

1 = 0 0 1 4

→ Prgrm - class program |

{
psum (main (string > args))

ff / 0 == 0 || 0 == 3 & (0 == 0)

true || false

for (int row = 0; row < 4; row++) if (0 == 0 || 0 == 3 & 1 == 0)
true || false

{
for (int col = 0; col < 4; col++)

row = 0 col = 0

if (row == 0 || row == 3 || col == 0 || col == 3)

s.o. p (" * ");

X X X A

size

0 1 2 X

s.o.p (" ");

X * X

5. o. p ();

X X X X

2

row = 2 col

3

psum :- * X X A X

* X X X X

X 1 1 X (1,3) X

* X X X X

X (5,1) X X (3,3) X

class program |

{

psum (string > args)

4

for (int row = 0; row < 5; row++)

5

int col = 0; int i, col++]

if ($\tau_{00} = 20$ || $\tau_{00} \text{ and } 2 = 4$ || $\text{col} = 20$ || $\text{col} = 4$ || $\tau_{00} = 2 \text{ col}$) $\tau_{00} + 10 = 24$

If $\log_{10} x = 0$

S.O.P (n > u) :

EIGE

S.O.P (" ")

{} s.o.p() :

3

2

Pray3:-	*	*	x	x
	*	*	x	x
	*	*	x	x
	*	*	x	x
	*	*	x	x

class programs

۲

psom (smiling args)

2

for (int row = 0; row < 5; row++)

colts

~~for (int col = 0; col < r)~~

2

if ($\text{row} \geq 0$ || $\text{row} = 2$ || $(\text{col} = 0 \text{ || } (\text{col} = 4) \text{ || } (\text{row} = 2 \text{ || } (\text{col} = 2))$

S.O.L (* * *)

slice

S. 2.8 ("W")

S. S. P. ~~C~~ C.

40

PSVM (--)

2

int n=15;

`bar (int max = 0; max < n ! good {`

~~for (int i = 0; i < col; i++)~~

$|col| = 24$ if $row \geq 0 \text{ or } row + col = n - 1$ $col = 0 \text{ or } col = n - 1$ $row = 0 \text{ or } row = n - 1$

$|col| = n/2 \text{ or } row = col \cdot 1 \text{ or } row + col = n - 1$

S.O.P ("X"):

$n = 4$ for row/col

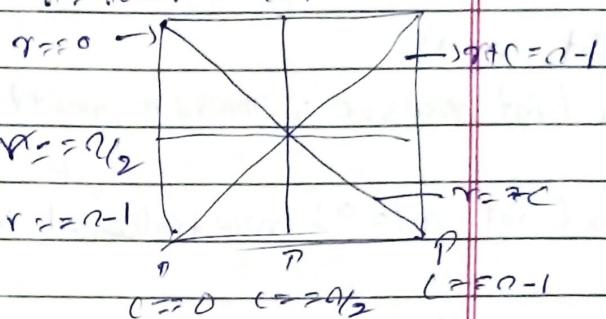
SOP

S.O.P (" "):

S.O.P (1):

)
)

Triangle



25/19

$h = 4$

x

x			
(0,0)	(0,1)	(0,2)	(0,3)

$r=0 \quad 0 \leq t$

x x

$c=0 \quad 0 \leq t \checkmark$

x x x

x x		
(1,0)	(1,1)	(1,2)

$r \neq 0 \quad 0 \leq t \checkmark$

x x x x

x x x		
(2,0)	(2,1)	(2,2)

$c=1 \quad 1 \leq t \checkmark$

$\text{if } (c \geq 1) \times$

x x x x		
(3,0)	(3,1)	(3,2)

$c=2 \quad 2 \leq t \checkmark$

$\text{if } (c \geq 2) \times$

$c=3 \quad 3 \leq t \checkmark$

$\text{if } (c \geq 3) \times$

$c=4 \quad 4 \leq t \times$

Day 1:- class triangle1

{

$n = 4$

rows/cols/units/rows)

{

$r=0 \quad 0 \leq t$

int p=4;

$c=0 \quad 0 \leq t \checkmark$

for (int rows=0; rows < n; rows++)

$c=1 \quad 0 \leq t \times$

{

$r=1 \quad 1 \leq t$

for (int col=0; col < p; col++)

 \ otherwise

 S.O.P (" * ");

 if (rows >= col)

)

 S.O.P (" ");

)

size

```
s.o.p(" ");
```

}

```
s.o.p();
```

}

}

FLOYD'S triangle

prog:- class triangle

{

```
psum (int n) {
```

}

```
int n=4;
```

```
int count=1;
```

```
for (int rows=0; rows<n; r++)
```

{

```
for (int col=0; col<rows; col++)
```

{

```
s.o.print( count++ + " " );
```

}

```
s.o.p();
```

}

prog:- class triangle

{

```
psum ( int n ) {
```

{

```
int n=4;
```

```
for ( int rows=0; rows<n; r++)
```

```
{ int count;
```

```
for ( int col=0; col<rows; col++ )
```

{

```
s.o.p( count++ + " " ),
```

}

```
s.o.p()
```

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

prg:- class triangle

psw (string [] arr)

int a=5

for (int row=0; row<5; row++)

{

for (int col=0; col<row; col++)

{

if (row+col==5)

s.o.p("1")

size

s.o.p("n")

}

s.o.p();

}

prg:- class triangle

{

psw (string [] arr)

{

int a=5; char count='A';

for (int row=0; row<5; row++)

{

for (int col=0; col<row; col++)

{

if (s.o.p(count++ + " "));

5

s.o.p();

}

Aa

Bb Cc

Dd Ee Ff

Gg Hh Ii Jj

1

0 1

1 0 1

1 0 1 0 1

ASC

0511 → A-2-G5-90

A-2-47122

'A' - 'A' → 18-87

'1' + '2' → 51

'1' + '2' → 99

prg:- class

5

psw (str

)

int n=5

char c1

char c2

for (in

for (

c.0

3

s.0

})

prg:- cl

1

psw (

)

int a=9

int count

int tem

for (int

1

for (

3

s.o.

count

temp

)

prg:-
class triangle9

81519

5
return (string(7 args))

Aa

1
int n=5;

Bb Cc

char c1='A';

Dd Ee Ff

char c2='a';

Gg Hh Ii Jj

for (int rows=0; rows<n; rows++)

4

for (int col=0; col<rows; col++)

2

s.o.p (c1++ +" " + c2++ + " ");

3

s.o.p ('');

2

prg:- class triangle10

1

return (string(7 args))

y=0 0L4

1→y

(=0 0L=0

←3 2 Ⓛ

4

int a=4;

count = 1

6

(=1 1L=0X

←10 9 8 7

int count=1

count = 0+3 = 3

5+4 Ⓛ

int temp=3;

temp = 3+2 = 5

1→y

for (int r=0; r<n; r++)

y=1 1L=4

←3 2 Ⓛ

1

for (int c=0; c<r; c++)

(=0 0L=1

←10 9 8 7

2

for (int c=0; c<r; c++)

count+=3

6+5 Ⓛ

3

s.o.p (count -+ " ");

c=1 1L=1

7

4

s.o.p ();

count = 2

6+5 Ⓛ

5

count = count + temp;

c=2 2L=1X

←3 2 Ⓛ

6

temp = temp +2;

(=2 count = 1+5 = 6

6+5 Ⓛ

7

s.o.p ();

temp = 5+2 = 7

7

8

count = count + temp;

c=2 2L=3

7

9

temp = temp +2;

(=3 count = 6+9 = 15

7+6 Ⓛ

10

s.o.p ();

c=3 3L=3

7+6 Ⓛ

11

count = count + temp;

(=4 count = 6+11 = 17

7+6 Ⓛ

12

temp = temp +2;

c=4 4L=4X

7+6 Ⓛ

ARUN'S PAGE NO. 1/1

PROBLEM :- class triangle

param (string (>args))

int n = 4;

int temp = 3;

int count = 1;

for (int row = 0; row < n; row++)

for (int col = 0; col < row; col++)

S.O.P (count-- + " ");

if (col < row)

S.O.P (" ");

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

count = count + temp;

temp = temp + 2;

1
3 * 2

6 * 5 * 4

10 * 9 * 8 * 7

0 * 3

1 * 2

7 * 6

4 * 0

PROBLEM :-

class prob

2 5

param (string

int n = 5

int count

for (int i = 0;

i < n;

for (int j = 0;

j < i;

S.O.P (" ");

count = count + 1;

if (j < n - 1)

else

count = count + 1;

}

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

1

2 5

3 6 8

4 7 9 10

count = count + 1;

else

count = count + 1;

}

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

PROBLEM :- class triangle

param (string (>args) /

int n = 5;

for (int row = 0; row < n; row++)

int count = row + 1;

int temp = n - i;

for (int col = 0; col <= row; col++)

S.O.P (count + " ");

count = count + temp - 1;

S.O.P (n + " ");

1

2 5

3 6 8

4 7 9 10

count = count + 1;

else

count = count + 1;

}

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

count = count + 1;

if (j < n - 1)

else

$n=3$

1 * 2 * 3 *
 2 * 5 * 9
 4 * 5 * 6

program $n=5$

class program {

```
int n=5;
int count = 1;
```

```
for (int i=0; i<n; i++)
    for (int j=0; j<i; j++)
        if (j+1 == 0)
```

S.O.P (Count++ " ")

s.o.p ():

if ($i < n/2$)

Count = Count + 1;

else

Count = Count - (x * n);

}

.

(Ans)

u have to write separate logic for

$n=5$

1 * 2 * 3 * 4 * 5 * 6
 11 * 12 * 13 * 14 * 15 * 16
 21 * 22 * 23 * 24 * 25 * 26
 16 * 17 * 18 * 19 * 20 * 21
 6 * 7 * 8 * 9 * 10 * 11

$n=1$ count = 1

$n=2$ count = 13

$n=3$ count = 35

$n=4$ count = 14

$n=5$ count = 15

$n=6$ ~~16 + 5 = 21~~

$n=7$ 25

$n=8$ 0

$n=9$ 11

$n=10$ 22

$n=11$ 33

$n=12$ 44

$n=13$ 55

$n=14$ 66

$n=15$ 77

$n=16$ 88

$n=17$ 99

$n=18$ 1010

$n=19$ 1111

$n=20$ 1212

$n=21$ 1313

$n=22$ 1414

$n=23$ 1515

$n=24$ 1616

$n=25$ 1717

$n=26$ 1818

$n=27$ 1919

$n=28$ 2020

$n=29$ 2121

if ($x > 2$) $c=2$ 225

else $c=0$ 025

Count = $28 - (2 \times c)$ $c=3$ 325

Count = 16 $c=1$ 125

$x=3$ 325 $c=4$ 425

$x=0$ 625 $c=0$ 025

Count = 16 $c=1$ 125

$x=1$ 125 $c=2$ 225

Count = 17 $c=3$ 325

Count = $21 - (3 \times 5)$ $c=4$ 425

Day 1:- class pyramid1

9/5/19

PSUMC contains args

int n=3;

```
for (int rows=0; rows<n; rows++)
```

```
for (int col=0; rows==col; col++)
```

2

```
s.o.p ("* ");
```

3

```
s.o.p ("* ");
```

```
for (int col=0; rows+col<n-1; col++)
```

4

```
s.o.p ("* ");
```

5

```
s.o.p ("* ");
```

6

Day 2:- class pyramid2

7

```
psum (string args)
```

8

```
int n=3;
```

```
for ( int rows=0; rows<n; rows++)
```

9

```
for (int col=0; col<n; col++)
```

10

```
if (rows+cols>=n-1)
```

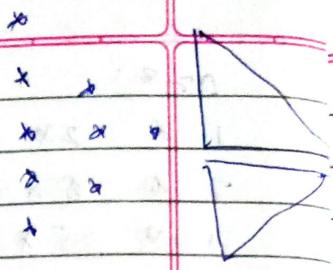
```
s.o.p ("* ");
```

size

```
s.o.p (" " );
```

11

```
for (int row=1; row<n; row++)
```



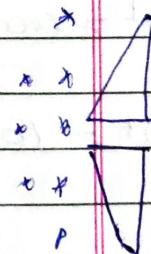
for (int
2
Pf ("
C.O.P
size
S.O.P

3
S.O.P (")

4
psum :- class
5
PSUMC (---)

6
int a=3
7
if (a>1)
for (int row

8
if (a>1) {
for (int col=0; col<a; col++)
for (int row=0; row<a; row++)
if (row+col>=a-1)
s.o.p ("* ");
else s.o.p (" ");



9
S.O.P (" ");
10
if (a>1) {
for (int col=0; col<a; col++)
for (int row=0; row<a; row++)
if (row+col>=a-1)
s.o.p ("* ");
else s.o.p (" ");

11
for (int st

12
S.O.P (" ");
13
}

14
S.O.P (" ");
15
}

16
}

for (int col=0; col < n; col++)

 Pf (space <= col)

 S.O.P ("* ");

 S18e

 S.O.P (" ");

}

prnt :- class program

PCUD (---)

int a=3

if(a) loop decides no of spaces

 for (int row=0; row < n; row++)

}

 if(a) loop decides no of space in col

 for (int space=0; space < n-1-row; space++)

 cout << " ";

 S.O.P (" ");

}

 if(a) loop decides no of stars

 in col

 for (int star=0; star < col; star++)

}

 S.O.P ("* ");

}

 S.O.P (" ");

}

}

 star=4 as=4

 --*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

--*

</div

PY5: Reversed pyramid

PSUM(string) contd

int n=3;

|| true loop decide no of rows

for (int rows=n-1; rows>0; rows--) → rows-1; rows++; rows--

|| true loop decide no of spaces in col

for (int space=0; space<n-1; space++)

↓

S.O.P (" ") ;

|| true loop decide no of stars in col

for (int star=0; 2*star >= n; star++)

↓

S.O.P (" *") ;

S.O.P (" ") :

return

assignment → run one not before for loop

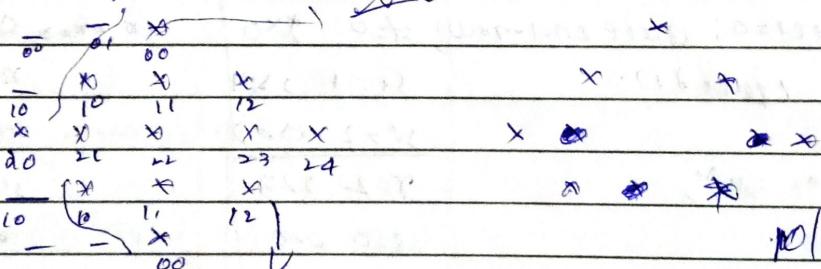


Diagram - class pyramid

public static void main (String args)

↓

int n=3;

|| true loop decides no of rows

for (int row=0; row<n-1; row++)

↓

|| true loop decides no of spaces in col

for (int space=0; space<n-1-rows; space++)

↓

S.O.P (" ") ;

decide no of stars in col

* * * *

* * *

*

when r=0 → c=2 & st=1 } previous

when r=1 → sp=1 & st=3 } pg 4

when r=2 → sp=0 & st=5

for (int s=

S.O.P (" ")

S.O.P ("");

|| true loop

for (int row=

|| true loop

for (int space=

S.O.P ("")

|| true loop

for (int star=

S.O.P ("")

S.O.P ("")

S.O.P ("")

S.O.P ("")

PSUM(string)

n=3 0 1 2

r=0

sp=0 0 1 2

st=1 1 2 3

sp=2 2 3 4

st=0 0 1 2 3

st=1 0 1 2 3

int n=7

for (int r=

for (int s=

for (int st=

for (int sp=

for (int n=

for (int r=

for (int s=

for (int st=

for (int sp=

for (int n=

for (int r=

for (int s=

for (int st=

for (int sp=

for (int n=

1 2 3

0 1 2

1 2 3

2 3 4

3 4 5

4 5 6

5 6 7

6 7 8

7 8 9

0 1 2 3 4 5 6 7

1 2 3 4 5 6 7 8

2 3 4 5 6 7 8 9

3 4 5 6 7 8 9 0

4 5 6 7 8 9 0 1

ARUN'S

DATE / / 3
PAGE NO. 1
ARUN'S

```

for (int star=0; 2*rows >= star; start++)
    r=2 2>0
}
s.o.p ("*");
s.o.p ("");
}
// true loop decides no of rows
for (int rows = n-1; rows >= 0; rows--)
    st=0 2>0
}
if false loop decides no of spaces in col
for (int space=0; space<n-1-rows; space++)
    st=1 2>0
}
s.o.p ("  ");
}
// false loop decides no of stars in col
for (int star=0; 2*rows >= star; start++)
    s.o.p ("*");
}

```

prg :- class pyramid

```

}
s.o.p ("");
}
if false loop decides no of stars in col
for (int star=0; 2*rows >= star; start++)
    s.o.p ("*");
}

```

prg :- class pyramid

```

}
psvm (string > crs)
{

```

int n=7

```

for (int rows=0; rows<n-1; rows++)

```

2

```

for (int space=0; space<n-1-rows; space++)

```

~~else or~~

```

s.o.p ("  ");

```

3

```

for (int star=0; 2*rows >= star; start++)

```

if (star >= 20) if star == 2*rows

ARUNI'S	PAGE NO.	DATE
28/2/2023	1	1.8.2023

```

s.o.p ("  ");
}

```

S.O.P (1)

```
for (int rows=0; rows<=n-1; rows++)
```

{

```
for (int space=0; space<n-1-rows; space++)
```

{

```
s.o.p(" ")
```

{

```
for (int star=0; stars<=rows; star++)
```

{

```
if (star==0 || star==2*rows)
```

```
s.o.p(" *")
```

else

```
s.o.p(" ")
```

{

```
s.o.p("
```

{

prog3 - class pyramid6

{

```
psum(stars) ans
```

{

```
int n=4
```

```
for (int rows=0; rows<n; rows++)
```

{

```
for (int space=0; space<n-rows; space++)
```

{

```
s.o.p(" ")
```

{

```
for (int star=0; stars<=rows; star++)
```

{

```
s.o.p(" " + "
```

{

```
s.o.p()
```

{

prob:- class

2

psum(stars)

3

int n=4;

for (int i=0; i<n; i++)

int count=0;

for (int space=0; space<i; count; space++)

{

s.o.p(" ")

{

for (int star=0; star<=i; star++)

{

if (rows>5)

s.o.p(count)

{

size

{

s.o.p(" .")

{

psum()

{

prog5 - class

{

psum(stars)

{

int n=4;

for (int i=0; i<n; i++)

{

int count=0;

for (int star=0; star<=i; star++)

{

s.o.p(" ")

{

if (star==0)

{

for (int star=1; star<=i; star++)

{

s.o.p(" " + "

{

s.o.p()

{

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

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

if

else

if

else

else

else

else

else

prg:- class pyramid?

2
for(string < i < n)

3
int n=4;

for(int i=0 ; i<n ; i++)

4 int count=1;

for(int space=0 ; space<n-i-row ; space++)

5 s.o.p (" ");

6

for(int star=0 ; star<row ; star++)

7

if(rows>star)

s.o.p(count++ " ");

8

size

9 s.o.p(count-- + " ");

10

s.o.p();

11

prg 5 :- class pyramid

12

for(string < i < n)

13

int n=4;

for(int rows=0 ; rows < n ; rows++)

14

int count=1;

for(int space=0 ; space<n-1-row ; space++)

15

s.o.p(" ");

16

for(int star=0 ; star<row ; star++)

17

if(star+space == 0)

s.o.p(count++ + " ");

18 s.o.p(" ");

19 s.o.p(" ");

929	x=0	0 4
	st=0	0 5=0
	if(0 3=0)	
	st=1	0 5=1
	x=1	1 4
	st=0	1 5=6
	if(1 5=6)	
	count=2	
	st=1	2 5=1
	if(1 5=1)	
	count=1	
	st=2	2 5=2

1	*	2	
10	11	12	
1	*	2	3
20	21	22	23 24
1	*	2	3
30	31	32	33
3	4	5	6

even odd

24 1

2

325 326

if(star+space == 0)

s.o.p(count++ + " ");

18 s.o.p(" ");

19 s.o.p(" ");

DATE 28/3/19
PAGE NO. 3
NAME _____

a=3

b=4

c=5

3 1 3	1 4 1 4 4
3 2 3	4 4 2 4 4
3 3 3	4 0 3 4 4

55 155

55 255

55 355

55 455

55 555

prog 6:- class Scanner

{

num convert to any()

{

int n=3;

int temp=0;

if (n>2)

temp=n+1;

for (int rows=0; rows<n; rows++)

{

for (int col=0; col<temp; col++)

{

if (col==n-1)

s.o.p ("row"+rows+" ");

else

s.o.p (" "+");

}

s.o.p ();

{

Conversion programs

1) binary -> 0,1 → 0b (DB) Value (0,1)

2) octal -> 0-7 -> 0 value

3) decimal -> 0-9

4) Hexadecimal -> 0-9 / A-F -> 0x value

class program

{

psum()

{

int binary = 0b101;

s.o.p ("binary=" + binary);

int octal = 025;

s.o.p ("octal=" + octal);

int hexa =

s.o.p ("hexa")

2

2

DWAP to C

1) num = 43

out = '1'

while no

class decision

{

public static

String out = "

while (a >

{

out + num =

num = num +

2

return out

{

return ()

scanner sc

s.o.p ("enter

int num = sc

String res = de

s.o.p ("binary

3

3

2) WAP to

class D

{

public static

{

String s

+

String w

+

while (w >

{

out + w =

out

return out

num = num/c;

}

return out;

}

pcosm (char* c> args)

{
Scanner in = new Scanner (System.in);

S.O.P ("Enter the number to converted : ");

int num = in.nextInt();

string res = decToOct (num);

S.O.P ("The converted octal no is : " + res);

}

}

DECOA To convert decimal number to hexadecimal

class DECOAHex

{

public static void Hexd (int num)

{

String out = "" ;

while (num != 0)

{

int rem = num % 16;

switch (rem)

{

case 10: out = 'A' + out;

break;

case 11: out = 'B' + out;

break;

case 12: out = 'C' + out;

break;

case 13: out = 'D' + out;

break;

case 14: out = 'E' + out;

break;

case 15: out = 'F' + out;

break;

08

143 -> de

815 F

8F.

default : out = zero + out;

break;

}
num = num / 10;

}
defout out;

}
pcorn (string) > urys)

}
scanner pn = new scanner (urys);

if (pn.laeter number "1")

put num = pn.nextInt();

string yrs = dectohexa (num);

if (p.laeter "1" + neg);

}

}

4) WAP to convert decimal to other no system which is
selected by user.

WAP to convert binary to decimal

$$(1101)_2 \rightarrow ()_{10}$$

$$1) \text{rem} = 1101 \% 10 = 1$$

$$\text{out} = 1 + 1 * 2^0 \Rightarrow 1$$

$$\text{num} = 11001 / 10 = 1100$$

$$2) \text{rem} = 1100 \% 10 = 0$$

$$\text{out} = 1 + 1 * 2^1 \Rightarrow 2$$

$$\text{num} = 1100 / 10 \Rightarrow 110$$

$$3) \text{rem} = 11 \% 10 = 1$$

$$\text{out} = 1 + 1 * 2^2 \Rightarrow 1$$

$$\text{num} = 11 / 10 = 1$$

$$4) \text{rem} = 1 \% 10 = 1$$

$$\text{out} = 1 + 1 * 2^3 \Rightarrow 9$$

$$\text{num} = 1 / 10 = 1$$

$$5) \text{rem} \% 10 = 1$$

$$\text{out} = 9 + 1 * 2^4 \Rightarrow 25$$

$$\text{num} = 1 / 10 = 0$$

class binary to decimal

{
public static int binToDec

{
int out = 0;

int a = 0;

while (num != 0)

)

int rem = num % 10;

out = out + rem * power(2, a++);

num = num / 10;

}

return out;

DATE
PAGE NO.
ARUN'S

Binary to dec
1st

(D110)

1) $x = 1210 \times 10 \Rightarrow 0$

LSI

$1210 / 10 \Rightarrow 121$ public static boolean validate (int num)

2) $x = 1211 \times 10 \Rightarrow 1$

LSI

$1211 / 10 \Rightarrow 121$

3) $x = 121 \times 10 \Rightarrow 1$

LSI

$121 / 10 \Rightarrow 12$

\rightarrow Bin to Dec
2nd

4) $x = 12 \times 10 \Rightarrow 2$

2 > 1

return false

1) write a program to determine power of given number without using any built method

prog:- import java.util.Scanner

class power

public static power(int num, int pow)

int out = 1;

for (int i = 0; i < pow; i++)

{

out = out * num;

}

return out;

}

main (String args)

Scanner sc = new Scanner (System.in);

s.o.p ("Enter the number");

int num = sc.nextInt();

s.o.p ("Enter the power");

int pow = sc.nextInt();

int res = power (num, pow);

s.o.p ("The " + num + " power of " + num + " = " + res);

}

Binary -> decimal

(1011)₂ → (11)₁₀

$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

= 8 + 0 + 2 + 1

= 11

import java.util.Scanner

class BinaryToDecimal

{

public static int binToDec (int num)

int num, out = 0; // output variable

int a = 0; // power variable

while (num >= 0) { repeat till num = 0

1011

10

11

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

10

1

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

1011

<p

loop till 1st least digit of num > 1.10^21

and stores in result

{ if end of while return result
return out; // terminates loop early

else - 3 + 10^23 out = 3 + 8 = 11
num = 1/10 = 0 a = A

returning to args

Scanner sc = new Scanner(System.in);

int num = sc.nextInt();

int res = binToDec(num);

S.O.P("the decimal equivalent of " + num + " is " + res);

) // end of main

} // end of class

main (int[])

import java.util.Scanner;

class BinaryToDecimal

{

public static boolean isBinary(int num)

{
while (num != 0) if repeat till num == 0

end num = num % 10; // extract last digit

if (res > 1) // condition for invalid binary number

return false; // invalid binary num

} // end of method

public static int binToDec (int num)

{
int out = 0; // output variable

int quo; // lower variable

while (num != 0) if repeat till num == 0

out = out + num % 10; // extract last digit

num = num / 10; // remove last digit

out = out + num * power(2, i++);

unmultiply

num = num / 10;

if end

return

else end

sum (

2

Scanner

sistem

int

boolea

if (

2

int

S.O.P

3

else

4

S.O.P (

5

recom

6

hexad

(1AB)

7

1x10^2

1x10^2

256

8

10

10

7

10

10

10

multiply last digit with 2 power n and store in result

num = num / 10; // eliminate last digit

{ in end of while

return out; // return result

} // end of method

public static void main(String[] args)

{

Scanner sc = new Scanner (System.in);

System.out.print("Enter the number");

int num = sc.nextInt();

boolean result = validate(num);

If (result)

{

int res = binToDec(num);

S.O.P ("the decimal equivalent of "+num+" = "+res);

}

else

{

S.O.P ("invalid binary number");

↳ end of main

↳ end of class

hexadecimal to decimal

$(1AB)_{16} \rightarrow (?)_{10}$

or = '1b'

out = 0 + 'b' * 16^0
 $45 * 16^0$

$$1 \times 16^2 + A \times 16^1 + B \times 16^0$$

$$(151 - 87) \times 16^0$$

$$1 \times 16^2 + 10 \times 16^1 + 11 \times 16^0$$

$$(100 - 87) \times 16^0$$

$$256 + 160 + 11$$

$$11 \times 16^0 = 11$$

$$\Rightarrow 427$$

↳ hex to decimal
 $10^2 = 0 - 48$
 $10^1 = 1 - 40$

char

Hexa code

ASCII

char

'2'

'-

'50'

'A' - 65

- 10

'@'

- a

'3'

'-

'51'

'B' - 66

- 11

'B'

- b

'4'

'-

'52'

'C' - 67

- 12

'9'

- c

'5'

'-

'53'

'D' - 68

- 13

'0'

- d

'6'

'-

'54'

'E' - 69

- 14

'1'

- e

'7'

'-

'55'

'F' - 70

- 15

'2'

- f

'8'

'-

'56'

ANS

PAGE NO.

DATE

15/5/19

SOP

{ Chars

import java.util.Scanner;

class ScannerDecimal

String s1 = "1AB";

int out = 0;

public static int hexaToDec(String s1)

{

int oct = 0;

int a = 0;

int n = s1.length() - 1;

while (n >= 0)

{

char c1 = s1.charAt(n - 1);

if (c1 >= 47 & & c1 <= 58)

{

oct = oct + (c1 - 47) * power(16, a++);

}

condition to lower case

condition to upper case else if (c1 >= 65 & & c1 <= 90)

{

oct = out + (c1 - 55) * power(16, a++);

}

else if (c1 >= 48 & & c1 <= 57)

{ out = out + (c1 - 48) * power(16, a++);

}

else

{

return -1; // invalid character

}

}

}

}

}

take a input from keyboard sc = new Scanner (System.in);
System.out.println("Enter the number");

local method String num = sc.next();

int res = hexaToDec(num);

if (res == -1)

{

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

}

5/5/19

SLR

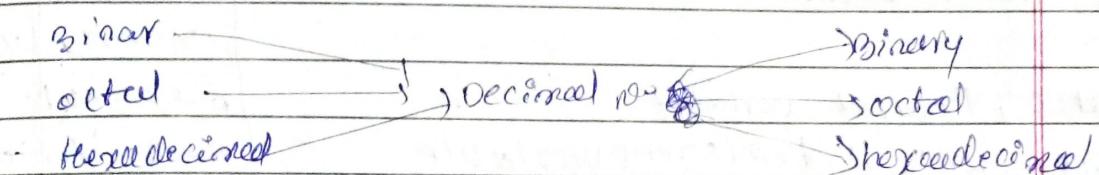
4

S.O.P ("invalid input")

7/3

Assignment

- 1) WAP to convert binary no to octal no
- 2) ——— to binary no
- 3) ——— octal no to binary no



(Q) (16, a++): 1) WAP to calculate factorial of a given Number
 Import `Scanner class`;
 Class Factorial

(Q) (16, a++): public static int fact (int num)
 {
 int out = 1;
 for (int i = num; i > 1; i--)
 {
 out = out * i;
 }
 return out;
}

RS VM (String[] args)

except
 {
 Scanner sc = new Scanner (System.in);
 S.O.P ("enter the number");
 int num = sc.nextInt(); // take input from keyboard;
 int res = fact (num);
 S.O.P ("the factorial of " + num + " = " + res);
}

2

strong number
 $145 = 1! + 4! + 5!$
 $= 1 + 24 + 120$
 $= 145$

int out = 0
 int num = 145
 out = out + factorial(isfact(num))

out = 145
 rem = num % 10;
 num = num / 10;

if (out == 145) write(1 to same)

2).
 rem = 145 % 10 = 5
 out = 5 + isfact(5) ;
 out = 145
 rem = 145 % 10 = 1
 3) rem = 145 % 10 = 1
 out = 145 * 1 = 145
 num = 145 / 10 = 0

WAP
 calculate factorial of last digit and store
 out = out + factorial(isfact(rem));
 num = num / 10; // eliminate last digit

return (out == temp); // return result

psos (Scanner)

Scanner sc = new Scanner(System.in);
 S.O.P("Enter the number");
 int num = sc.nextInt(); // take input from keyboard

boolean res = isStrong(num); // call method

if (res)

S.O.P("it is strong number");

else

S.O.P("it is not a strong number");

$m = 145 / 10 = 5$

for (int i = 1; i <= 5; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

$m = 145 / 10 = 14$

for (int i = 1; i <= 14; i++)

cout << m;

write a program to display all the strong numbers from 1 to 1000

same as previous program

psum (string > args)

for (int num = 1; num < 100000; num++)

Scanner sc = new Scanner (System.in)

boolean res = isStrong (num);

if (res)

2

S.O.P (num);

}

} // end of for loop

)

way to display only 1st three strong numbers :

same as previous program

psum (string > args)

3

Scanner sc = new Scanner (System.in);

int count = sc.nextInt(); // enter the name of strong num

if (int num = 1; count > 0; num++)

)

boolean res = isStrong (num);

if (res)

4

S.O.P (num);

(count--)

5

6

1 determine sum of 1st 3 strong numbers

2 read as privilege

psum (string > args)

7

Scanner sc = new Scanner (System.in);

S.O.P (" enter the number of strong number")

int count = sc.nextInt();

int sum = 0;

for (int num = 1; count > 0; num++)

)

boolean res = isStrong(num);

if (res)

{

sum = sum + num;

count--

}

}

S.O.P (sum);

}

int

1) Determine strong number

16/3/19
int num = 153

2) Display strong number in range 1-n

3) Display first n strong numbers

4) Determine sum of factors

strong numbers

Arm strong number | int num = 153

$$153 = 1^3 + 5^3 + 3^3$$

$$\text{int out} = 0$$

$$= 1 + 125 + 27$$

$$= 153$$

$$\frac{153}{10} = \frac{15}{10} = \frac{1}{10} = 0$$

program:- import java.util.Scanner

class Armstrong

{

public static int counter (int num)

int

int a=0;

while (num!=0)

{

num = num / 10;

a++;

}

return a;

}

public static boolean isArmstrong (int num)

{ int a=0;

int temp = num;

while (num!=0)

{

int rem = num;

out = out + rem;

num = num / 10;

}

return out;

{

PC num (String)

2

Scanner sc

S.O.P ("other")

int num =

boolean res =

if (res)

{

S.O.P ("ans")

{

else

{

S.O.P ("not")

{}

wap to determine

if not

28 = 1+2+4+7

int num

int sum

for (int

{

16 (

{

sum =

int sum =

for (int

} { get

while (num != 0)

}

int rem = num % 10

lcount = lcount + freq[rem].freq; rcount (temp);

num = num / 10;

}

return lcount == temp;

}

psolve (string & args)

2

scanner sc = new scanner (systemin);

s.o.p ("either the number of Armstrong numbers")

int num = sc.nextInt();

boolean res = isArmstrong (num);

if (res)

{

s.o.p ("Armstrong number");

}

else

{

s.o.p ("not an Armstrong number");

}

3) WAP to determine whether a given number is perfect or not

$$28 = 1+2+3+4+5+6$$

$$6 = 1+2+3 = 6$$

$$i=1 \quad i \leq 14$$

$$\text{if } (28/i == 0)$$

int num = 28;

int sum = 0;

for (int i = 1; i <= num / 2; i++)

$$\text{sum} = 1+2+3$$

$$i=2 \quad 2 \leq 14$$

$$\text{if } (28/2 == 0)$$

$$16 \quad (28 \% 2 == 0)$$

$$i=3 \quad 3 \leq 14$$

$$\text{sum} = \text{sum} + i;$$

$$i=4 \quad 4 \leq 14$$

return (num == sum);

$$\text{if } (28/4 == 0)$$

$$\text{sum} = 3+4=7$$

i) WAP to determine whether the given no is prime or not

```
int num=13;  
for(int i=2; i<=num; i++)  
{  
    if (num % i == 0)  
    {  
        return false;  
    }  
}  
return true;
```

13

num \Rightarrow 2 divides 13

psum (str)

<

int res;

cout < "res

>

Fibonacci

class Fibonacci

public static
return (int

'

s.o.p (act "

s.o.p (n2 +

606 (int i

'

n2 + n3 -

s.o.p (n3 +

01 = n2

n2 = n3 -

'

psum (str)

45 + 30 == 45 + 45 = 90

90 + 30 == 45 + 45 == 90

return 90

}

Array pro

class prop

{

psum (s

int (3

s.o.p ("

8nt [

0412

S.N.O.

ii) WAP to determine LCM of 2 numbers

int a=3; \rightarrow 3, 6, 9, 12, 15, 18 | int max=a;

int b=5; \rightarrow 5, 10, 15, 20 | if (b>a)

(- $y \cdot 3 = 0$ & $y \cdot 5 = 0$) | max=b;

if (5/3 == 0 & 5/5 == 0)

while (

if (max%a == 0 & max%b == 0)

return max;

max+=t;

max+=t;

class LCM

{

public static int lcm(int a, int b)

{

int max=a;

if (b>a)

max=b;

int temp=max;

while (true)

{

if (max%a == 0 & max%b == 0)

return max;

max+=temp;

}

max > max+temp;

psum (string) args

```

<
    int res = com(30, 9);
    s.o.p("res = " + res);
    >
    <

```

0 1 1 2 3 5 8 13 21 34

n1 n2 n3

class FibonacciSeries

```

class FibonacciSeries {
    public static void ps fibbo(0, 1);
    feature (int n1, int n2)

```

```

    <
        s.o.p("n1 = " + n1);
        s.o.p("n2 = " + n2);

```

```

        for (int i=0; i<8; i++) {

```

```

            n2 + n3 = n1+n2;
            s.o.p("n3 = " + n3); → s.o.p("n1+i", 1); → for single s.o.p call fair

```

n1 = n2

n2 = n3;

int n1 = 0;

int n2 = 1;

s.o.p(n1);

s.o.p(n2);

ps um [string] args

int count = 0;

while (count < 8) {

int n3 = n1+n2;

s.o.p(n3);

Array programs

class program 1

```

class program 1 {
    ps um [ string ] args
}

```

```

    int arr[] = new int[5];

```

```

    s.o.p("size = " + arr.length);

```

```

    int sum = 0;

```

ANSWER	PAGE NO.
--------	----------

sum[4];

("size = " + arr.length);