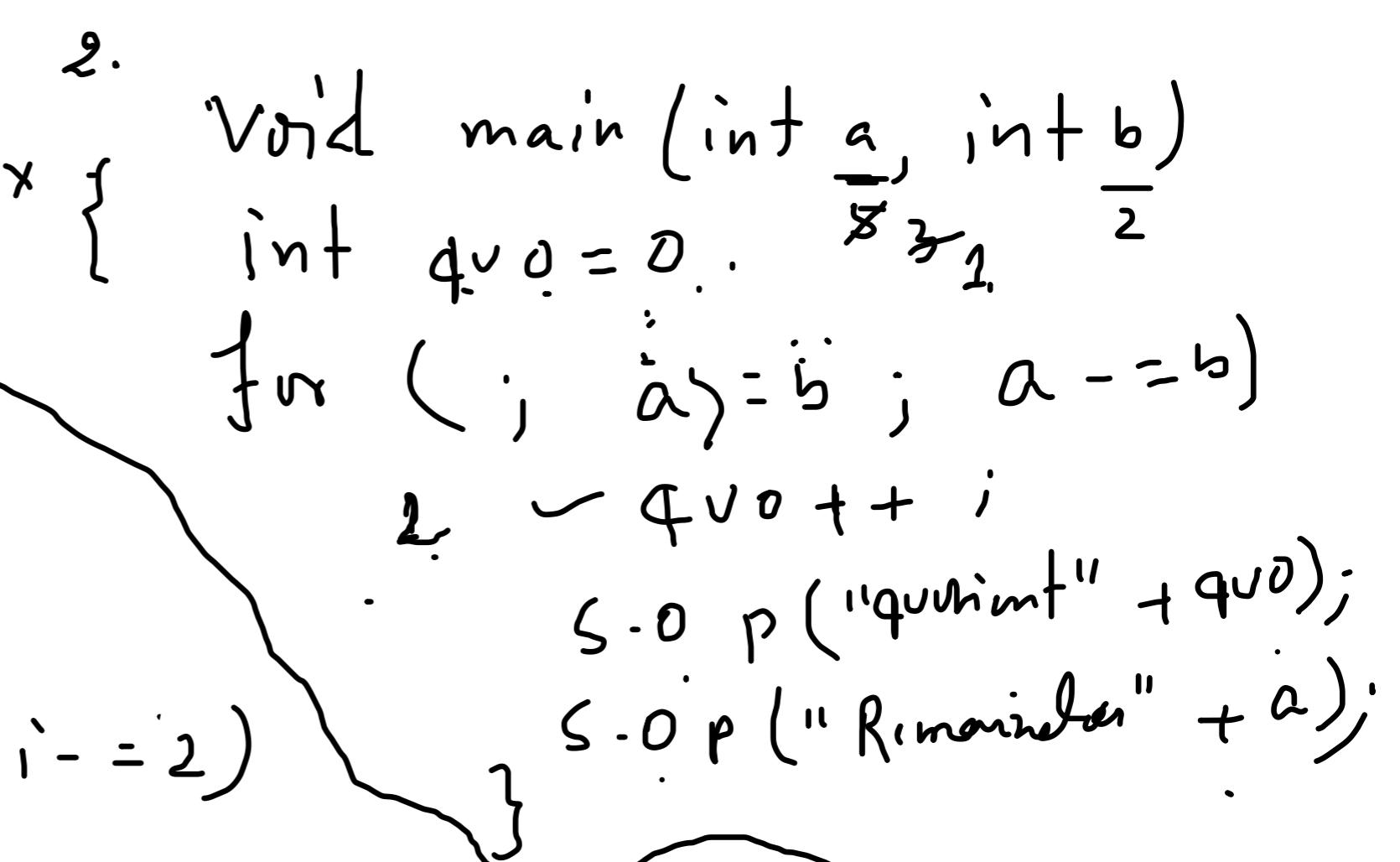


19.09

1. WAP to accept a number and check whether the number is odd or even without using %, / and any if construct.
2. WAP to find the quotient and remainder of a division without using %, /
3. WAP to print 1st 10 odd fibonacci terms
4. WAP to print each term and sum of the following series. $5 + 8 + 12 + 17 + \dots + \text{upto } n \text{ terms}$.

```
1. if (num % 2 == 0)
    s.o.p("even");
else
    s.o.p("odd");
```

```
{ void main(int num)
    int i;
    for (i = num; i >= 2; i -= 2)
        j
        s.o.print(i == 0 ? "Even" : "odd");
}
```

2. 

```
void main(int a, int b)
int quo = 0;
for (; a > b; a -= b)
    2. quo++;
s.o.p("quotient" + quo);
s.o.p("Remainder" + a);
```

3. 1, 1, ~~2~~, 3, 5, ~~8~~, 13, 21, ~~31~~, 55

```
{ void main()
    int i, a=1, b=0, c;
    for (i=1; i<=10; )
        c = a+b;
        if (c % 2 == 1)
            cout << c;
            i++;
    a = b;
    b = c;
}
```

i	a	b	s.o(c)
1	1	0	1
2	0	1	1
3	1	1	2
3	1	2	3
4			

4. $5 + 8 + 11 + 17 + \dots$

```
void main (int n)
{
    int diff = 3, num = 5, i, lmn = 1;
    for (i = 1; i <= n; i++)
    {
        cout << num; sum = lmn + num;
        num = num + diff; cout << " ";
        diff = diff + 1;
    }
    cout << endl ("sum of the series" + lmn);
}
```

Write a program to accept a sum of money and calculate the different number of denominations of Rs. 100, 50, 10, 2, 1 notes.

Example : Amount = 568

Output : $100 \times 5 = 500$

$$50 \times 1 = 50$$

$$10 \times 1 = 10$$

$$2 \times 4 = 8$$

```
-Void main(int amt)
```

```
{  
    int n100, n50, n10, n2, n1 ;
```

```
n100 = amt / 100 ; if (n100 > 0)
```

```
amt = amt % 100 ;
```

```
s.o.p ("100 x " + n100 + "=" + (n100 * 100));
```

```
n50 = amt / 50 ; if (n50 > 0)
```

```
amt = amt % 50 ; s.o.p ("50 x " + n50 + "=" + (n50 * 50));
```

```
n10 = amt / 10 ; - . . . .
```

```
amt = amt % 10 ; - . . . -
```

```
n2 = amt / 2 ; }
```

```
amt = amt % 2 ; ↗
```

```
n1 = amt ;
```

26.09 1) WAP to accept a number and check whether the number belongs to fibonacci series or not. Example : Input 6 output : Not in fib series
Input 8 output : Found in fib series.

2) UAP to accept a number and display the prime decomposition of the number. Example

Input 8 output - 2, 2, 2
Input 30 output - 2, 3, 5

3) WAP to display the highest and lowest factors excluding 1 and the number itself.

INPUT 10 output - 5, - highest

4) WAP to display the multiplication table of a number x up to y times.

Solution.

i) void main (int n) n = 6 / n = 8
 { int i, a = 1, b = 0, c ;
 for (i = 1; c <= n; i++)
 { c = a + b; → 1, 1, 2, 3, 5, 8
 if (c == n) w
 s.o.p ("Found in fib series"); w
 return; w
 }
 }
 }
 {
 a = b;
 b = c;
 s.o.p ("Not found in fib series");

 }
}

Q) Prime decomposition.

```
Void main (int num)
{
    int i;
    for (i = 2; i <= num; i++)
        if (num % i == 0)
            S.o. printf (.i + " ");
        num /= i;
    else
        i++;
}
```

$$\begin{array}{r} \text{num} = 8 \\ \hline \text{num} = 30 \end{array} \rightarrow 2, 2, 2$$
$$2, 3, 5$$

$$15 / 3 = 5$$

```
8. void main ( int num )
{
    int i ;
    for ( i = 2; i < num ; i++ )
        if ( num / i == 0 )
    {
        cout << "Lowest factor = " + i ;
        cout << endl ;
        return ;
    }
    cout << "Highest factor = " + (num / i) ;
    break ;
}
```

The handwritten annotations provide a visual guide to the flow of control:

- A green arrow points from the `return` statement to the `break` statement, indicating that the loop exits after the first factor is found.
- A green bracket groups the two `cout` statements, likely indicating they are part of the same output line.
- A green curly brace groups the entire `if`-block, starting from the opening brace and ending at the closing brace of the `if` statement.
- A green arrow points from the bottom curly brace back up to the closing brace of the `if`-block, forming a loop that highlights the entire conditional structure.

```

4. void main (int x, int y)
{
    int i;
    for (i = 1; i <= y; i++)
    {
        s.o.println(x + "x" + i + "=" + (x * i));
    }
}

```

$x = 6$
 $y = 10$
 $6 \times 1 = 6$
 $6 \times 2 = 12$
 $6 \times 3 = 18$

 \dots

03.10

Jump Statement.

- i) break → It is used in loop or switch block to take the control out of the block.
- ii) continue → It is used in loop structure to skip the part of the loop beyond it and proceed for next iteration.
- iii) return → It is used to stop the execution of a program and to pass the control to the calling function.

```
for (i = 1; i <= 8; i++)  
{  
    if (i % 3 == 0)  
        break;  
    s.o.p(i);  
}
```

```
for (i = 1; i <= 8; i++)  
{  
    if (i % 3 == 0)  
        continue;  
    s.o.print(i);  
}
```

break can be used in switch or loop only }
continue " " in loop only } NOT
 with
 if only

return can be used anywhere in a program within or outside loop.

Output:

```
for(i=2; i <= 7; s.o.print(++i))  
{  
    if(i%3 == 0)  
        continue;  
    s.o.print(i); } u
```

2 3 4 5 5 6 7 8 8

1). WAP to accept a number and check the number is triangular number or not.

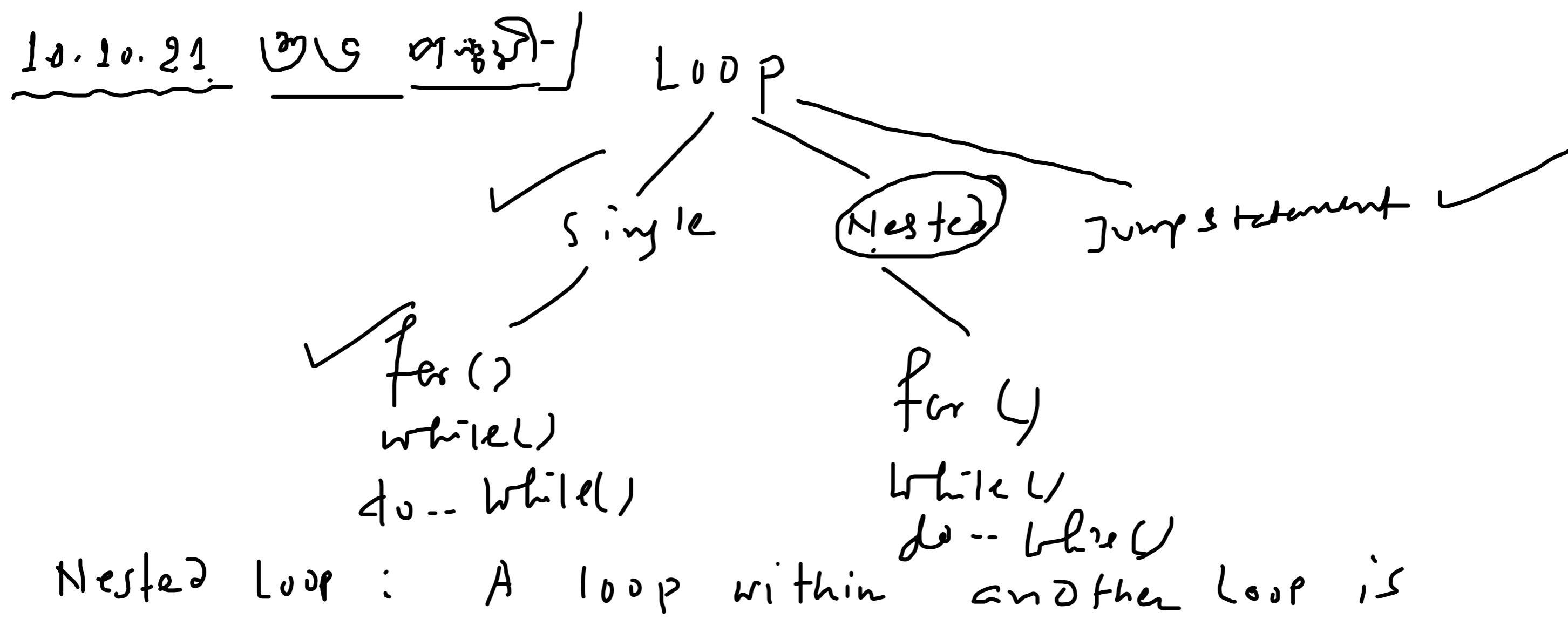
Example: $3 \rightarrow 1+2$ } if a number is the sum of all the natural numbers from 1, then it
 $6 \rightarrow 1+2+3$ }
 $10 \rightarrow 1+2+3+4$ } is triangular number.

- 2) WAP to display all the triangular number upto a given number
- 3) WAP to print 1st n triangular number.

```
1) void main (int num)
{
    int i, sum=0 ;
    for (i= 1; i<= num; i++)
    {
        sum = sum + i;
        if (sum == num)
        {
            s.o.pn ("Triangular");
            exit(0);
        }
    }
    s.o.pn ("Not Triangular");
}
```

2) void main (int num)
{
 int i ; sum=0 ;
 for (i= 1; sum<num; i++)
 {
 sum = sum + i;
 s.o.p (sum + " ");
 }
}

3) void main (int n)
{
 int i, sum=0 ;
 for (i= 1; i<= n; i++)
 {
 sum = sum + i;
 s.o.p (sum + " ");
 }
}



Nested Loop : A loop within another loop is called nested loop.

```
for (i = 1; i <= 5; i += 3).
```

```
{  
    for (j = 1; j <= 7; j += 4)  
        s.o.print(j);
```

```
→ s.o.print(j);  
    s.o.print(j);  
    s.o.print(j);  
}
```

Find the output ↑

i	j	Print
1	1	1
4	5	5
	9	9
	1	1
	1	1
	8	8
	4	4

1. WAP to print each term and sum of the following series.

$$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots n \text{ terms.}$$

2. WAP to print 1st n prime numbers

3. WAP to print 1st n terms of fibonacci Series in reverse order.

```

1. void main ( int x, int n )
{
    int j, i, k = 1, f;
    double term, sum = 0;
    ;
```

$\text{sum} + \text{term}$
 $\text{sum} = \text{sum} + \text{term}$

```

for ( i = 1; i <= n; i++ ) → 1, 2, 3
    {
        f = 1;
        for ( j = 1; j <= k; j++ ) →
            f = f * j;
```

x^k / f → $x^k / 1$
 x^3 / f → $x^3 / 6$

```

        term = Math.Pow ( x, k ) / f;
        sum += term;
    }
    cout << sum;
}

```

```
void main( int n )  
{    int i, j, num = 2 ;  
    for ( i = 1 ; i <= n ; )  
    {        int count = 0 ;  
        for ( j = 1 ; j <= num ; j ++ )  
        {            if ( num % j == 0 )  
                count ++ ; }  
        if ( count == 2 )  
        {            S. O. P ( num ) ;  
            i ++ ; }  
        num ++ ; }  
}
```

$n = 4$

prime check

```

3.
void main (int n)
{
    int i, j, a, b, c = 0;
    for (i = n; i >= 1; i--)
    {
        a = 1;
        b = 0;
        for (j = 1; j <= i; j++)
        {
            c = a + b;
            a = b;
            b = c;
        }
        cout << c;
    }
}

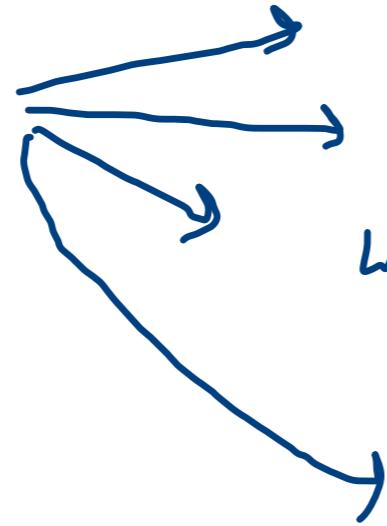
```

$$\begin{array}{c}
 \eta = 5 \\
 \underline{5, 3, 2, 1, 1} \\
 \hline
 \end{array}$$

21.16

WAP for 21Y.

Nested Loop



for() within for()

for() within while() / do...while()

while() / do while()

within while() / do while()

while() within for()

Nested Loop \Rightarrow Pattern printing.

- i) WAP to check a number Krishnamurthy or Special number or not.
 $145 \Rightarrow 1! + 4! + 5!$
- ii) WAP to print all the prime number in a range $n, y (y > n)$

iii) WAP to print all the 3 digit SPY number.

$$1124 \rightarrow \frac{1+1+2+4}{8} = \frac{1 \times 1 \times 2 \times 4}{8}$$

iv) WAP to print all the 3 digit Armstrong number.

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

```
Void main( int num )  
{  
    int copy = num, d, sum = 0, fact, l;  
    while (num > 0)  
    {  
        d = num % 10; fact = 1;  
        for (i = 1; i <= d; i++)  
            fact = fact * i;  
        sum = sum + fact;  
        num /= 10; }  
    if (num == copy)  
        S. O. Put ("Special");  
    else S. O. Put ("Not  
        Special"); }
```

```

ii)
void main ( int x, int y ) (ii)
{
    int i, count, j ; ;
    for (i = x; i <= y; i++)
    {
        count = 0 ;
        for (j = 1; j <= i; j++)
            if (i % j == 0)
                count++ ;
        if (count == 2)
            S.o. print ( i + " " ) ;
    }
}

```

```

void main ( )
{
    int i, d, sum = 0, prod = 1, copy ;
    for (i = 100; i <= 999; i++)
    {
        copy = i ; sum = 0 ; prod = 1 ;
        while (copy > 0)
        {
            d = copy % 10 ;
            sum += d ;
            prod *= d ;
            copy /= 10 ;
            if (sum == prod)
                S.o. print ( i + " " ) ;
        }
    }
}

```

iv)

```

void main()
{
    int i, sum, copy;
    for (i = 100; i <= 999; i++)
    {
        copy = i;
        sum = 0;
        while (copy > 0)
        {
            d = copy % 10;
            sum = sum + d * d * d;
            copy = copy / 10;
        }
        if (sum == i)
            cout << i << endl;
    }
}

```

$$125 \div 10 \Rightarrow 5$$

$$\begin{array}{r}
153 \\
\downarrow \\
153 = 1^3 + 5^3 + 3^3 \\
\underline{\quad} \quad \underline{\quad} \quad \underline{\quad}
\end{array}$$

$$\begin{array}{c}
(\text{int}) \text{Math} \cdot \text{pow}(d, 3) \\
\underline{\quad \quad \quad}
\end{array}$$

31.10

- i) WAP to check a number armstrong
number or not.

$$1634 = 1^4 + 6^4 + 3^4 + 4^4$$

- ii) WAP to check a number disarium number or not

$$135 = 1^1 + 3^2 + 5^3$$

- iii) WAP to check a number magic number or not.

$$\underline{7777} \rightarrow \underline{28} \rightarrow \underline{10} \rightarrow 1$$

- iv) WAP to check a number happy number or not

$$68 \rightarrow 6^2 + 8^2 \rightarrow 100 \rightarrow 1^2 + 0^2 + 0^2 \rightarrow 1$$

~~for(n)~~ 1) { void main(int num) 1631
 int count = 0, d, copy = num, ;
 { while (copy > 0)
 count++; → 4
 } copy /= 10;
 copy = num ;
 { while (copy > 0)
 d = copy % 10; → 9 $9^4 + 6^3 + 4^2$
 sum = sum + (int) Math.pow(d, count);
 } copy /= 10;
 if (num == sum)
 System.out.println("Armstrong");
 else System.out.println("Not");
}

2) Disarium
 //
 //
 //
 //
 //
 count --

```
3) { void main ( int num )
```

```
    int sum = 0, d ;
```

```
    while ( num > 9 )
```

```
        sum = 0 ;
```

```
        while ( num > 0 )
```

```
            d = num % 10 ;
```

28

```
            sum = sum + d ;
```

10

```
            num /= 10 ;
```

,

```
        }
```

```
    if ( num == 1 )
```

```
        S.o p( " Magic" );
```

```
    else p( " Not Magic" );
```

7777

↓

28

↓

10

↓

1

4)

Happy number

1,

1,

1,

sum = sum + d * d

1,