Date 12/8/2020

Types of Statements:

There are 3 types of statements:

1. Linear/Sequential statements: which are executed one after other
2. Conditional: which are executed on the basis of a condition
3. Loop: which are executed more than once.

Conditional statements:

There are two types of conditional statements:

1. If statement
2. Switch statement

If statement:

There are four forms of if:

1. if
2. If – else
3. If –else if—else(Ladder if)
4. If within if(Nested if)
5. Simple if:

Syn: if (condition)

{

//statements

}

For example:

Int N = 56;

If (N > 20)

{

Sop(“No. is greater than 20”);

}

1. If else

Syn: if (condition)

{

//statements

}

else

{

//statements

}

For example:

Int N = 56;

If (N > 20)

{

Sop(“No. is greater than 20”);

}

else

{

Sop(“No…”);

}

13/8/2021

Loops:

Hello

Hello – 100times

Types of Loops:

1. Entry controlled
2. Exit controlled

Entry controlled: are the loops which are checked first, if the condition is true only then the statements will be executed otherwise not.

1. For loop
2. While loop

Exit controlled: are the loops which are checked at the time of exit

1. Do---while loop

Things to look after:

1-100

1. Initialization
2. Condition
3. Modification

1 – 100 , 100 - 1

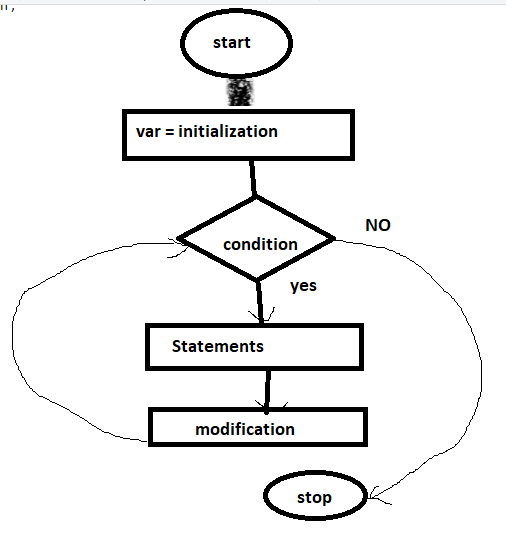
1. For loop:

Syntax: for (var = initialization ; condition ; modification)

{

//statements

}

I = i+1 or i+=1

For (i = 1 ; i<=100 ; i++) i= 2, 3,,,,,,,100,101

{

Sop(i);

}

1

2

3

-

- 99

100

I++ , ++i

B = ++i

C = i++

While loop:

Syn :

Var = initialization;

While(condition)

{

//statements;

Modification;

}

Date 16/8/21:

1. WAP to input ‘n’ terms from the user and display its Fibonacci series

N = 7

0,1 ,1,2,3,5,8

1. WAP to input ‘n’ terms from the user and display Lucas series

N = 5

0,1,2,5,10

0+1+1

1+2+2

2+5+3

5+10+4

1. WAP to input ‘n’ terms and a value for x from the user:

Val = 2, n=4

22/2! + 23/3!+24/4!+25/5!

Operations:

1. Power
2. Factorial
3. Divide
4. Add

**While loop:**

Syn :

Var = initialization;

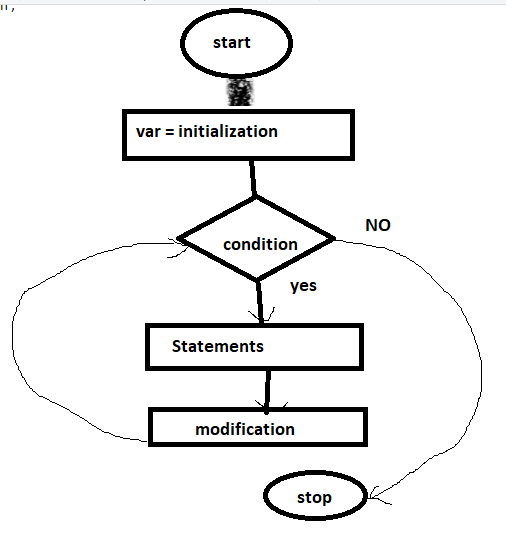
While(condition)

{

//statements;

Modification;

}



For example: display numbers from 1-100

WAP to input any number and display sum of digits

N = 2345 , r, sum =0;

Step 1:

Num = 2345 >0

R = 2345 % 10 🡺 5

Sum = sum + r; 🡺 0+5 🡺5

Num = 2345 /10 🡺234

Num = 234

Step 2: 234 > 0

R = 234 % 10 🡺 4

Sum = sum + r 🡺5+4 🡺9

Num = num /10 🡺 234 /10 🡺23

Step 3: Num = 23 > 0

R = 23 % 10 🡺 3

Sum = 9+3🡺12

Num = num/10 🡺23/10🡺2

Step 4: num = 4>0

R = 4%10 = 4

Sum = sum + r🡺12+4🡺16

Num = num/10 🡺4/10 🡺0

Num = 0

2+3+4+5 = 14

1234 , 6789

1. WAP any 4 digit number and reverse it.

Num = 123 🡺 “Invalid number”

Num = 12345 🡺”Invalid number”

Num = 2345

Output = 5432

1. WAP to any number and check if it is Armstrong or not

Num = 1234

Digits = 4

1^4+2^4+3^4+4^4

1+16+81+256 = 354

Not an Armstrong number

153🡺armstrong number

Do –while loop

Syn:

var = initialization;

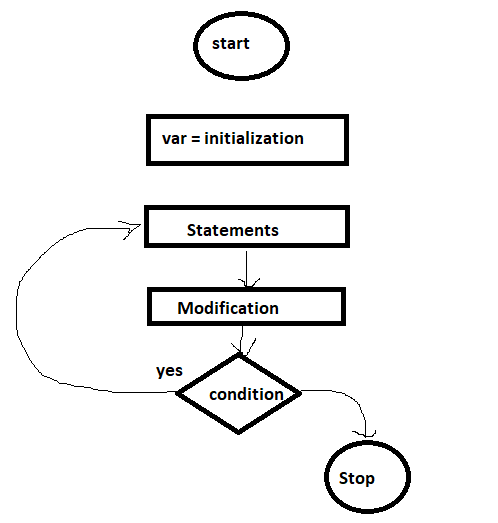
do

{

//Statements;

Modification;

}while(condition);



NOTE: Exit controlled loops are executed at least once.

WAP to display numbers from 1-10;

1. WAP to input any number and display its factorial using do—while.

Date 20/8/2021

Input

Display

Sorting

Searching

Searching: to find an element from set of elements

Complexity:

Best case:

Average case

Worst case

Time complexity:

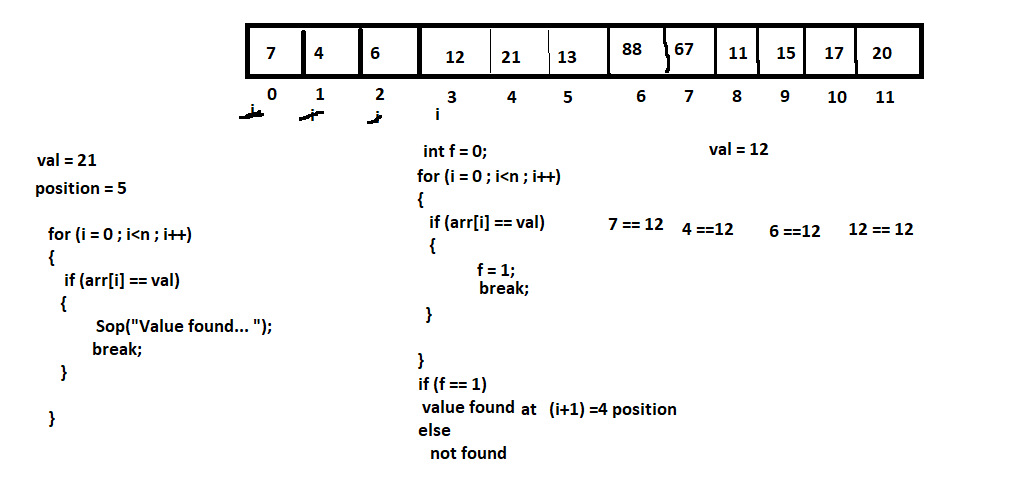
Space:

There are two types of searching :

1. Linear / Sequential Search
2. Binary Search

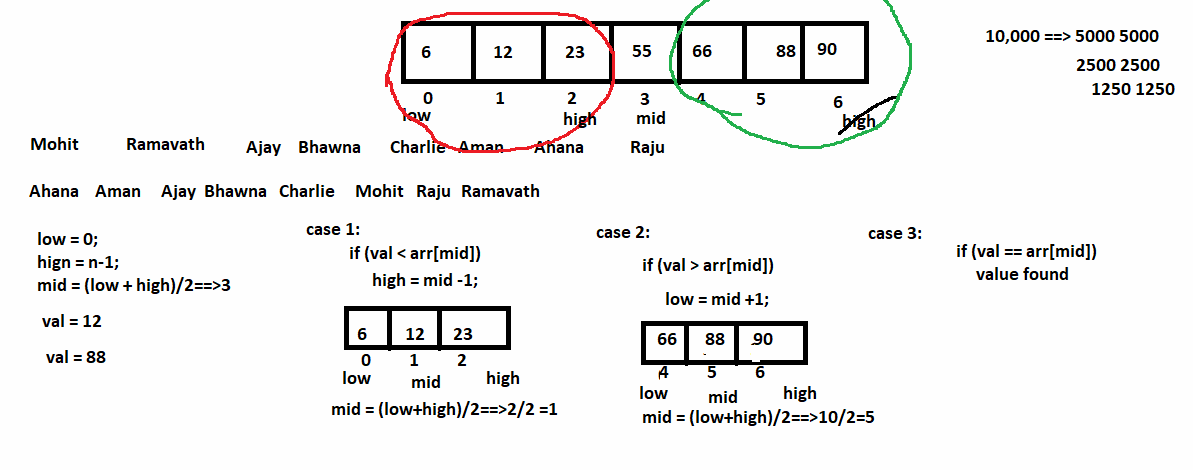
Linear / Sequential Search: Sorting is not required in this search.

It is good in case if you have lesser numbers of elements



Binary Search: Sorting is must.

Best when you have larger number of elements.



Date 21/8/2021

int arr[] = {2,34,5,6,7};

for (i = 0 ; i<arr.length ; i++)

{ arr[i] = sc.nextInt();

Sop(arr[i]); ==>2 34 5 6 7

}

For each/Enchanced for loop

1. used for accessing the elements

2. no need to give initialization,condition and modification

for (int i :arr)

{

Sop(i); ==>2 34 5 6 7

}

Functions: dividing the code into small independent modules

Main()

{

Int A b c

Enter three numbers….

Sum = a+b+c;

Sop(sum);

Dafs

Dddd

Ddd

Enter three numbers….

Sum = a+b+c;

Sop(sum);

Aa

Aa

Aa

Enter three numbers….

Sum = a+b+c;

Sop(sum);

}

Println()

Categories of Functions/Methods:

1. Predefined: println(), nextInt(),……
2. User defined

Categories of user defined methods:

1. No return value, no argument
2. No return value, with argument
3. Return value, no argument
4. Return value, with argument

Method Segments:

1. Method declaration
2. Method definition
3. Method calling

Method declaration/ Method prototype: used in abstract classes and interfaces

Syntax:

Return\_type method\_name(data\_type arg1 , data\_type arg2,….);

1. Void sum();
2. Void sum(int a, int b);
3. Int sum( );
4. Int sum(int a , int b);

Method Definition: Set of statements/ definition will be given

Syntax: Return\_type method\_name(data\_type arg1 , data\_type arg2,….)

{

//defintion

}

For example: void sum()

{

//statements }

Method calling:

Syntax: value = method\_name(arg\_list);

Programs using 2nd category:

1. WAP to input any number and display its factorial
2. WAP to input ‘n’ terms and display Fibonacci series

Int A[] = {2,3,4,5,6,7}

Variable: is a name given to a memory location. A variable can have a single value at a time.

A

5

A12AA

Arrays: it is a collection of elements of similar type. It occupies continuous memory location.

It is also called as subscripted variable.

Type of Arrays:

1. Single Dimensional
2. Multi Dimensional

Single (1-D) :

Syntax :

1. Array Declaration:

Data\_type array\_name[];

Or

Data\_type [] array\_name;

For example: int arr[];

1. Memory Allocation:

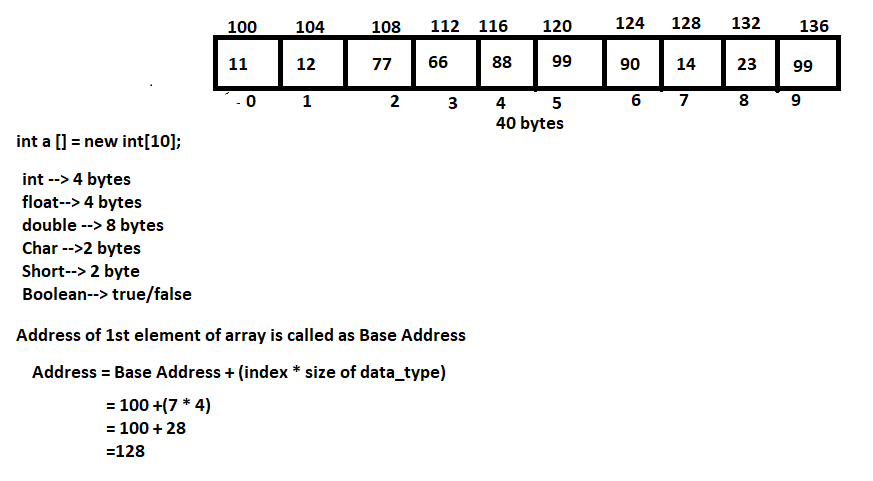
Array\_name = new data\_type[size];

For example: arr = new int[10];

Combine 1 and 2

Int arr[] = new int [10];

Memory Representation:



2-D Array: Array of Arrays. Used to represent data in a tabular/matrix format.

Syn:

1. Array Declaration:

data\_type array\_name[][];

For example: int arr[][];

1. Memory Allocation: array\_name = new data\_type[row\_size][col\_size];

Arr = new int [2][3];

Combine 1 and 2

Int arr [][] = new int [2][3];

