DSA-3

Program: Print Even Numbers:-

Approach-1

i = 2 4 6 8 10 12

```
for (int i = 2; i <= 10; i = i+2) {
    cout << i << " ";
}
```

Algorithm:-

1. Start
2. initialize i = 2
3. Repeat loop while i ≤ 10
4. Print i
5. Increment i by 2
6. Stop

Approach-2

1 % 2 ≠ 0
2 % 2 = 0
3 % 2 ≠ 0
4 % 2 = 0

```
for (int i = 1; i <= 10; i++) {
    if (i % 2 == 0)
        cout << i << " ";
}
```

Algorithm:-

1. Start
2. initialize i = 1
3. Repeat loop while i ≤ 10
4. if i % 2 == 0, print i
5. Increment i by 1
6. Stop

Program: Print Alphabets from a to z:-

i = 'a'
i++ → 97+1 → 98 → 'b'

```
for (char i = 'a'; i <= 'z'; i++) {
    cout << i << " ";
}
```

Algorithm:-

1. Start
2. initialize i = 'a'
3. Repeat loop while i ≤ 'z'
4. Print i
5. Increment i by 1
6. Stop

Program: Sum of first 'n' natural numbers:-

sum=0
int=1, sum=1
int=2, sum=3
int=3, sum=6

```
int sum = 0;
for (int i = 1; i <= 10; i++) {
    sum += i;
}
cout << sum;
```

Algorithm:-

1. Start
2. initialize i = 1, sum = 0
3. Repeat loop while i ≤ 10
4. Add i to sum
5. Increment i by 1
6. Print sum
7. Stop

Note

Why do we initialize sum = 0?
It contains a random value from memory.

Program: Sum of Squares of first 'n' Natural Numbers:-

```
int sum = 0;
for (int i = 1; i <= 5; i++) {
    sum += i*i;
}
cout << sum;
```

i = 1
sum = 1
i = 2
sum = 5
i = 3
sum = 14

Algorithm:-
1. Start
2. Initialize i = 1, sum = 0
3. Repeat loop while i ≤ 5
4. Calculate i×i and add to sum
5. Increment i by 1
6. Print sum
7. Stop

Program: Reverse Counting :-

```
for(int i = 10; i >= 1; i--) {
    cout << i << " ";
}
```

i = 10 9 8 7 6 5 4 3 2 1 0

Algorithm:-
1. Start
2. Initialize i = 10
3. Repeat loop while i ≥ 1
4. Print i
5. Decrement i by 1
6. Stop

Program: Multiplication Table :-

```
int num = 5;
for (int i = 1; i <= 10; i++) {
    cout << num << " x " << i << " = " << (num * i) << endl;
}
```

5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20

→ Nested for Loop

```
for(initialization ; condition; update) {
        for(initialization ; condition; update) {
            // Code to repeat
        }
}
```

Pattern 1:
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

i = 1
j = 1 2 3 4 5
i = 2
j = 1 2 3 4 5

```
for(int i = 1; i <= 5; i++) {
    for(int j = 1; j <= 5; j++) {
        cout << j << " ";
    }
    cout << endl;
}
```

Pattern2:

```
a b c d e
a b c d e
a b c d e
a b c d e
a b c d e
```

```
i=1
i=2
j=a b c d e
```

```
for(int i=1; i<=5; i++){
    for(char j='a'; j<='e'; j++){
        cout<<j<<" ";
    }
    cout<<endl;
}
```

Pattern3:

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

```
i=1
j=1 to 5
Print*
j=2
Print*
j=3
Print*
```

```
for(int i=1; i<=5; i++){
    for(int j=1; j<=5; j++){
        cout<<"*"<<" ";
    }
    cout<<endl;
}
```

Pattern4:

```
*
* *
* * *
* * * *
* * * * *
```

```
i=1
j=1 to 1
Print*
i=2
j=1 to 2
Print*
```

```
for(int i=1; i<=5; i++){
    for(int j=1; j<=i; j++){
        cout<<"*"<<" ";
    }
    cout<<endl;
}
```

Pattern5:

```
* * * * *
* * * *
* * *
* *
*
```

```
i=5
j=1 to 5
Print*
i=4
j=1 to 4
Print*
```

```
for(int i=5; i>=1; i--){
    for(int j=1; j<=i; j++){
        cout<<"*"<<" ";
    }
    cout<<endl;
}
```

Pattern6:

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
i=1
Print space→4
j=1 to 1
Print*
i=2
Print space→3
j=1 to 2
Print*
```

```
for(int i=1; i<=5; i++){
    for(int j=1; j<=5-i; j++){
        cout<<" ";
    }
    for(int j=1; j<=i; j++){
        cout<<"*"<<" ";
    }
    cout<<endl;
}
```