Looping :→   → while
                 for
                 do while

Entry Control loop   Exit Control loop

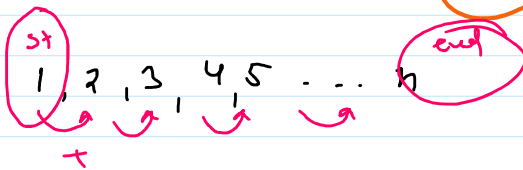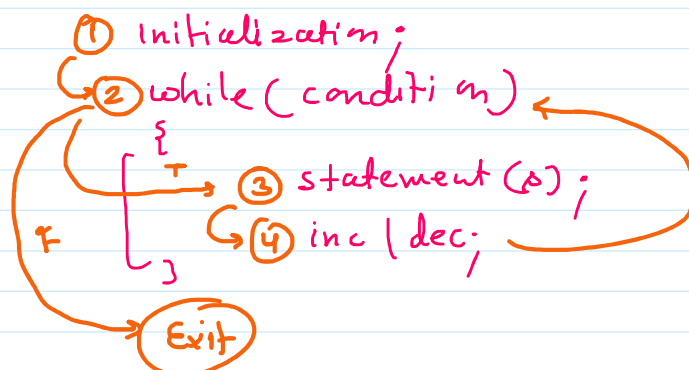 — while   — do while
  — for

Ⓐ Entry Control loop :→

        steps →   ① Initialization (start)
                     ② Condition (stop)
                     ③ statement (Repeat)
                     ④ inc | dec
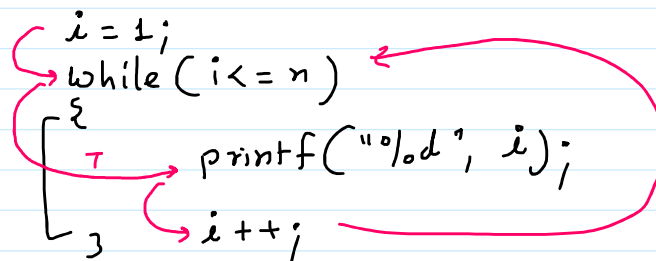   ++   --

while loop :—

① Initialization;
② while (condition)
   {
   T   ③ statement (s);
        ④ inc | dec;
   F
   }

Exit

end

st
1, 2, 3, 4, 5 . . . n

n $\boxed{5}$

i $\boxed{1\ 2\ 3\ 4\ 5\ 6}$

output

$\boxed{1, 2, 3, 4, 5}$

i = 1;
while (i <= n)
{
T   printf("%d", i);
     i++;
}

1, 3, 5, 7, 9

n $\boxed{10}$

i = 1;
while (i <= n)
{
     printf("%d", i*2-1);
     i++;
}

$\boxed{i * 2 - 1}$

1*2-1 → 1
2*2-1 → 3
3*2-1 → 5
4*2-1 → 7

5
6
7

```
                                    printf("%d", i*2-1);    5
                                                            6
                            ]       i++;                    7
                                                            8
                                                            9
                                                            10
                                                            ✗
```

```
   1  2  3   4  5  ... n
6, 6,
   1, 0, 1, 0, 1  ... n
```

```
              i=1;                      if (i%2 == 0)
              while (i<=n)
              {                          else              0
                  printf("%d", i%2)
                                                           1
              i++;
              }
```

```
   initialization                        ___        Error
   while (condition)                              while()
   {                                              {
          statement                                  ___
          inc/dec;                                   ___
   }                                              }
```

```
              Always true
   while (1)
   {
   }    ___
                 Always false
   while (0)
   {
       ___
   }
```

```
   n ... 3, 2, 1
```

```
                          j=n;              (n!=0)
                          while (n>=1)         (n)
                          {          (n)
                              n--;
                          }
```

```
#include<stdio.h>
int main()
{
    int i,n;
    printf("Enter number of terms:");
    scanf("%d",&n);
    while (n)
    {
        printf("%d ",n);
        n--;
    }
```

```
n  5 4 3 2 1 0
```

```
5, 4, 3, 2, 1
```

```
        printf("%d ",n);
        n--;
    }
    return 0;
}
```

$$\boxed{5\,,4\,,3\,,2\,,1}$$

factorial of a num →

n $\boxed{5}$! → 5 * 4 * 3 * 2 * 1 → 120

(with annotation: 4 3 7 120 over the box)

fact $\boxed{1\ 20\ 120 \atop 5\ 60\ \boxed{120}}$

while (n)
{
    fact *= n;   →   fact = fact * n
    n--;
}
printf("%d", fact);   → $\boxed{120}$

```
#include<stdio.h>
int main()
{
    int fact=1,n;
    printf("Enter a number:");
    scanf("%d",&n);
    while (n)
    {
        fact *= n;
        n--;
    }
    printf("%d",fact);
    return 0;
}
```

Sum of digits →

n $\boxed{1578}$ →   1 + 5 + 7 + 8 → $\boxed{21}$ sum

(annotations: 15 + 15, %10)

sum $\boxed{21}$

                                    1578 % 10 → 8

                                    1578 / 10 → 157

while (n)
{
    sum += n % 10;   →   sum = sum + n % 10
    n /= 10;   →   n = n / 10;
}
printf("%d", sum);

$\hookrightarrow$ printf ("%d", sum);

Rev of a num

n | 1578 $\longrightarrow$ Rev | 8751

rev | 0

rev | 0

while (n)
{
    rev = rev * 10 + n%10;
    n /= 10;
}

1578

87 * 10
870 + 5
875

8   7

87
8 * 10
80 + 7 $\to$ 87
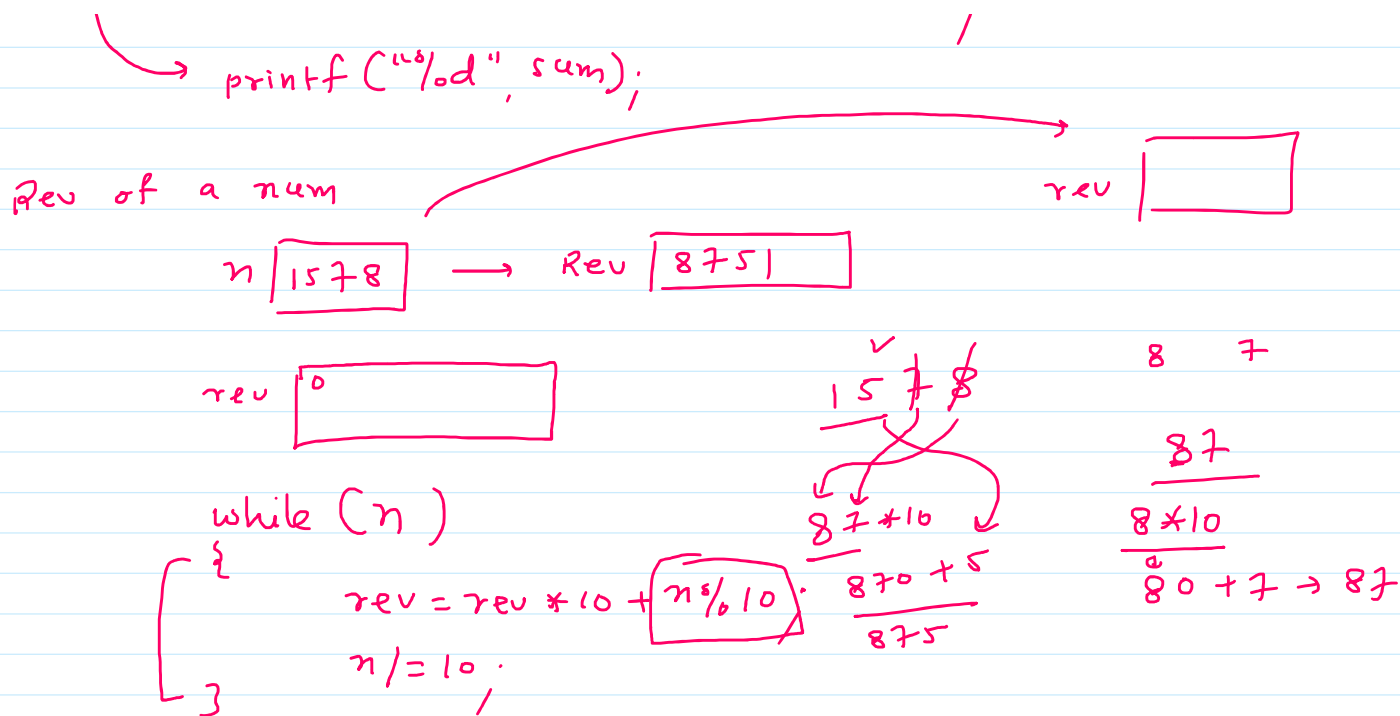
is palindrome :$\to$

121 $\xrightarrow{\text{rev}}$ 121

```c
#include<stdio.h>
int main()
{
    int rev=0,n,org;
    printf("Enter a number:");
    scanf("%d",&n);
    org = n;
    while (n)
    {
        rev = rev * 10 + n%10;
        n /= 10;
    }
    if(org == rev)
        printf("Palindrome");
    else
        printf("NOT palindrome");
    return 0;
}
```

method

pow (5,3)

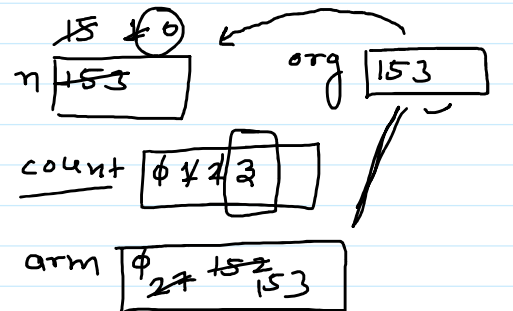is armstrong :$\to$

① count

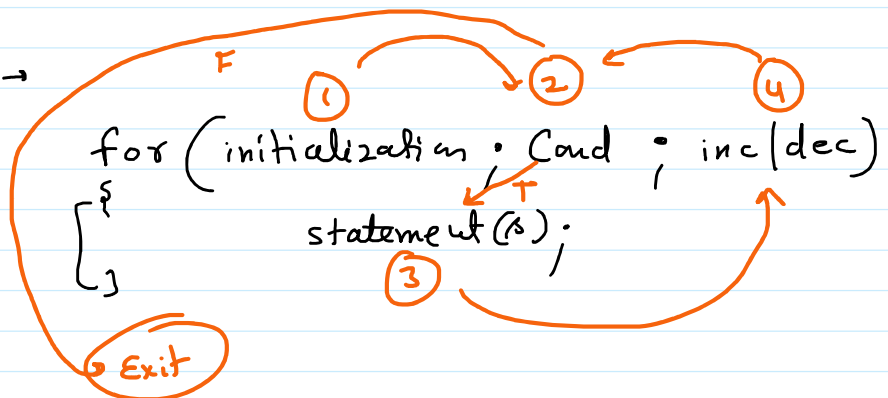153 $\to 1^3 + 5^3 + 3^3$

② scan

2 $\to 2^1$

4   4   4   4

② scan

digit count

$2 \rightarrow 2^1$

$1234 \rightarrow 1^4 + 2^4 + 3^4 + 4^4$

```c
#include<stdio.h>
#include<math.h>
int main()
{
    int n;
    double count=0,arm,org;
    printf("Enter a number:");
    scanf("%d",&n);
    org = n;
    while(n)
    {
        coaunt++;
        n/=10;
    }
    n = org;
    arm = 0;
    while(n)
    {
        arm += pow(n%10,count);
        n/=10;
    }
    if(org == arm)
        printf("Armstrong");
    else
        printf("Not Armstrong");
    return 0;
}
```

15 ≠ 0

n 153

org 153

count ∅ ∤ ⅓ 3

arm ∅ 27 ∤ 52 153

1
+25

1 8   3

125

27 -2
25

2

for loop :→

for (initialization ; Cond ; inc(dec))
{
}

① ② ④

F

T

statement (s);

③

Exit

No Error

Always True

for (i=1 ;   ; i++)
{

printf ("%d", i);

optional

for (; ↓ ;)

Compulsory

$$\text{for } (\underbrace{i=1, \ j=10}_{\text{initialization}} \ ; \ i <= j \ ; \ \underbrace{i++, \ j--}_{\text{inc/dec}})$$

```
{
    printf("%d  %d \n", i, j);
}
```

1	10
2	9
3	8
4	7
5	~~6~~

print fibonacci Series :—

ing ✓ ✓ ✓
0   1   1   2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 ...
a + b → c

a + b → c

a + b → c
    a   b

```c
#include<stdio.h>
int main()
{
    int i,n,a,b,c;
    printf("Enter number of terms:");
    scanf("%d",&n);
    for(i=1,a=0,b=1;i<=n;i++)
    {
        c = a+b;
        printf("%d ",b);
        a=b;
        b=c;
    }
    return 0;
}
```
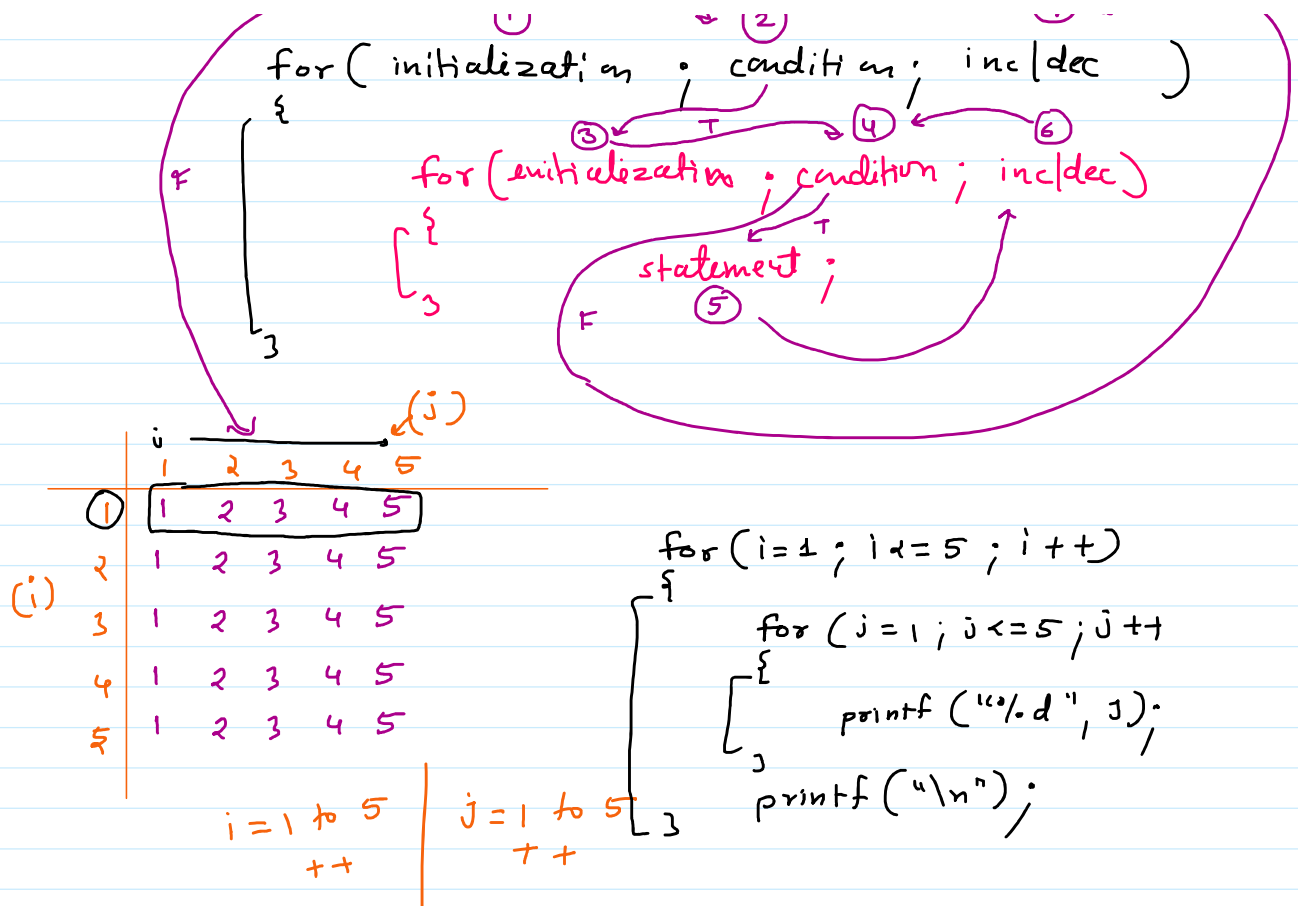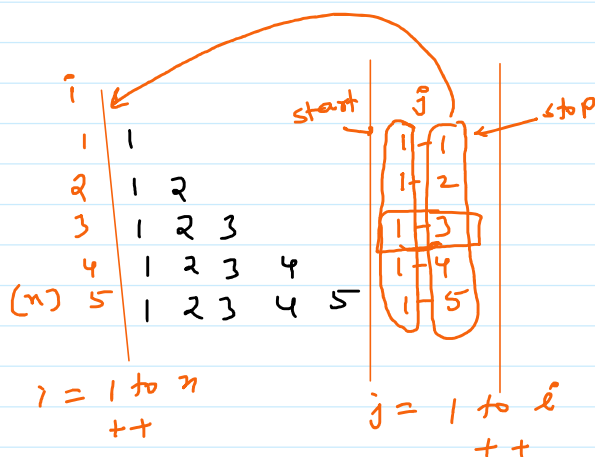
while (n--)

while (--n)

Nested for loop :→

for (initialization ; condition ; inc/dec)
①   ②   ⑦

for (initialization ; condition ; inc/dec)
{

}
F

for (initialization ; condition ; inc/dec)
{

}
T
statement ;
F

	1	2	3	4	5
1	1	2	3	4	5
2	1	2	3	4	5
3	1	2	3	4	5
4	1	2	3	4	5
5	1	2	3	4	5

(i)

i = 1 to 5
++

j = 1 to 5
++

```
for (i=1 ; i<=5 ; i++)
{
    for (j=1 ; j<=5 ; j++)
    {
        printf("%d ", j);
    }
    printf("\n");
}
```

```
#include<stdio.h>
int main()
{
    int i,n,j;
    for(i=1 ; i<=5 ; i++)
    {
        for(j=1 ; j<=5 ; j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```

i
1 | 1
2 | 1 2
3 | 1 2 3
4 | 1 2 3 4
(n) 5 | 1 2 3 4 5

start   j   stop
1 1
1 2
1 3
1 4
1 5

i = 1 to n
++

j = 1 to i
++

#include<stdio.h>
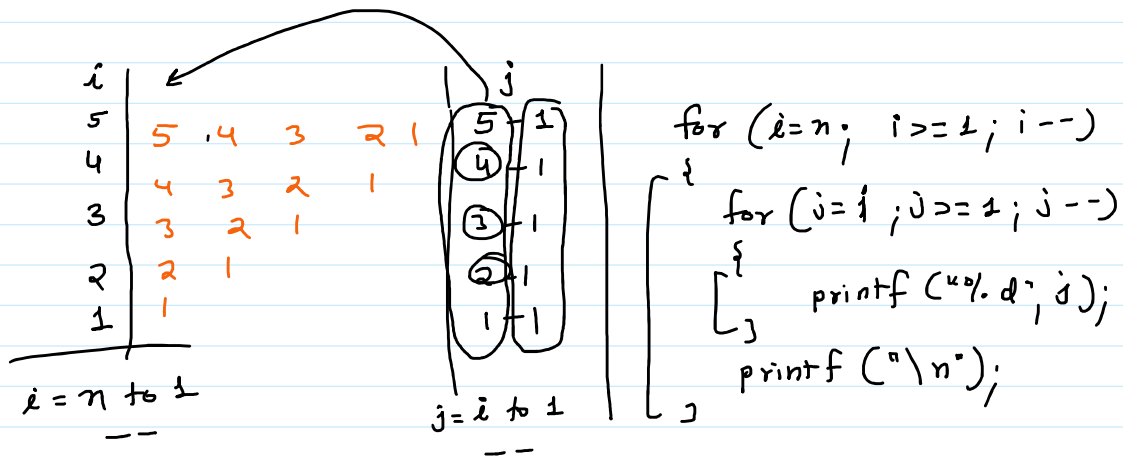
```
int main()
{
    int i,n,j;
    printf("Enter no of rows:");
    scanf("%d",&n);
    for(i=1 ; i<=n ; i++)
    {
        for(j=1 ; j<=i ; j++)
        {
            printf("%d ",j);
        }
        printf("\n");
    }
    return 0;
}
```
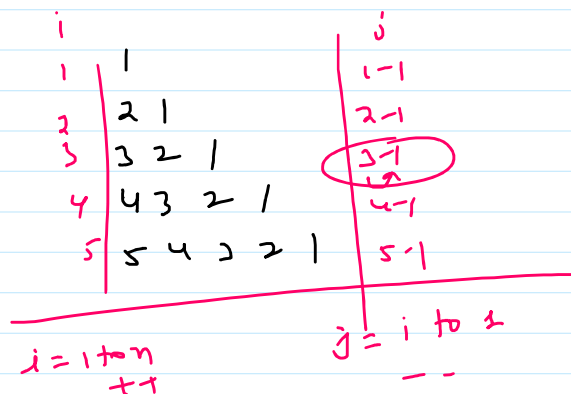
$--\ \rightarrow\ >=$
$++\ \rightarrow\ <=$

i
5   5 .4   3   2 1
4   4   3   2   1
3   3   2   1
2   2   1
1   1

i = n to 1
$--$

j
⑤ 1
④ 1
③ 1
② 1
1 1

j = i to 1
$--$

```
for (i=n;  i>=1; i--)
{
    for (j=1 ;j>=1; j--)
    {
        printf("%d ", j);
    }
    printf("\n");
}
```

1
0 1
1 0 1
0 1 0 1
1 0 1 0 1

odd 1
even 2   odd 1
odd 3   even 2   odd 1
even   odd   even   odd
4   3   2   1

i   j
1 | 1 | i-1
2 | 2 1 | 2-1
3 | 3 2 1 | 3-1
4 | 4 3 2 1 | 4-1
5 | 5 4 3 2 1 | 5-1

i = 1 to n   j = i to 1
++   --

```
#include<stdio.h>
int main()
{
```

```c
int i,n,j;
printf("Enter no of rows:");
scanf("%d",&n);
for(i=1 ; i<=n ; i++)
{
    for(j=i ; j>=1 ; j--)
    {
        printf("%d ",j%2);
    }
    printf("\n");
}
return 0;
}
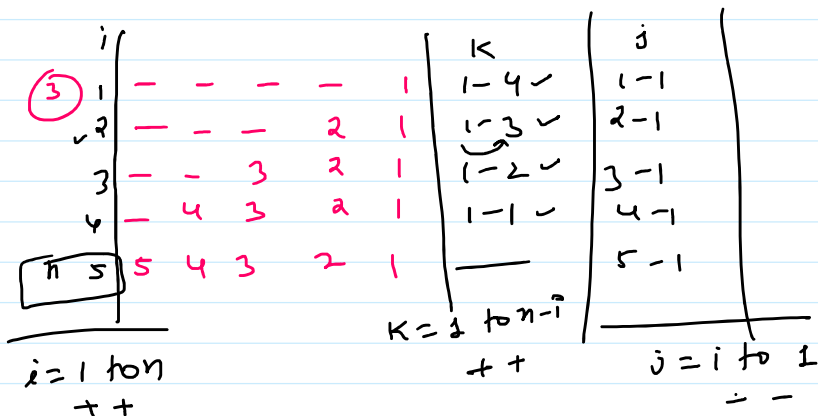```

H.co

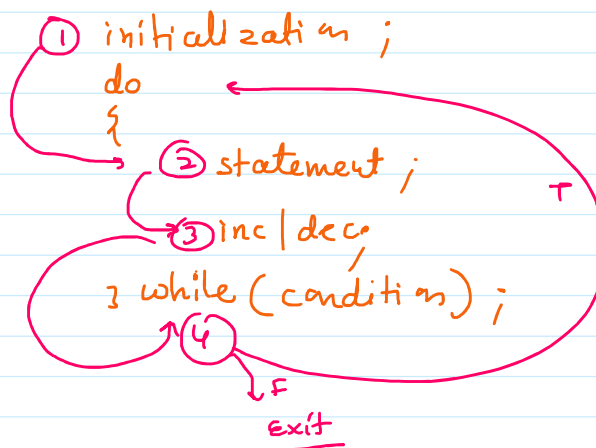① 
```
    1
  2   3
  4   5   6
  7   8   9   10
  11 12 13 14 15
```

② 
```
  1
  2   3
  4   5   6
  7   8   9   0
  1   2   3   4   5
```

③

| i | | | | | | K | j |
|---|---|---|---|---|---|---|---|
| 1 | — | — | — | — | 1 | 1-4 ✓ | 1-1 |
| 2 | — | — | — | 2 | 1 | 1-3 ✓ | 2-1 |
| 3 | — | — | 3 | 2 | 1 | 1-2 ✓ | 3-1 |
| 4 | — | 4 | 3 | 2 | 1 | 1-1 ✓ | 4-1 |
| n 5 | 5 | 4 | 3 | 2 | 1 | | 5-1 |

K = 1 to n-i          j = i to 1

i = 1 to n            ++          - -

++

## do while loop       Exit Control loop

① initialization ;
    do
    {
②  statement ;
③  inc | dec
    } while (condition) ;
④
    F
    Exit

T

1. Add

2. sub

3. mult

4. div

```c
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
    int i,n,a,b;
    do{
        system("cls");
        printf("1. Add\n");
        printf("2. Sub\n");
        printf("3. Mult\n");
        printf("5. Div\n");
        printf("0. Exit\n");
        scanf("%d",&n);
        switch(n)
        {
            case 1:
                printf("Enter 2 numbers:");
                scanf("%d%d",&a,&b);
                printf("%d",a+b);
                break;
            case 2:
                printf("Enter 2 numbers:");
                scanf("%d%d",&a,&b);
                printf("%d",a-b);
                break;
            case 3:
                printf("Enter 2 numbers:");
                scanf("%d%d",&a,&b);
                printf("%d",a*b);
                break;
            case 4:
                printf("Enter 2 numbers:");
                scanf("%d%d",&a,&b);
                printf("%d",a/b);
                break;
            case 0:
                printf("Bye Bye");
                break;
            default:
                printf("Invalid choice");
        }
        getch();
    }while(n!=0);
    return 0;
}
```

Jump statement :→

break ⎤
continue ⎦ → loop

goto

return → func

$$\boxed{\text{return} \rightarrow \text{func}}$$

for (i=1 ; i<=5 ; i++)   i $\boxed{\cancel{\times} \cancel{2} \cancel{3}}$

{
   if (i==3)
     break;
   printf ("%d ", i);
} $\boxed{1, 2}$

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

{
   if (i==3)
     continue;
   printf ("%d ", i);   1 2, 4, 5
}

int a = 1;
ab:
  printf ("%d ", a);
  a++;
  if (a <= 5)
    goto ab;

{
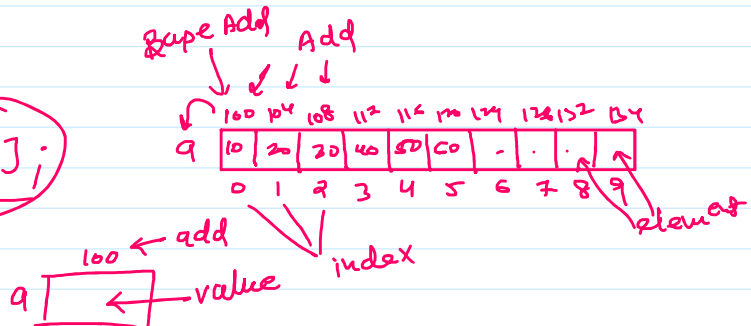 goto ab;
}

func ( )
{
}
a:

Array :→

$\boxed{\text{int a[10] ;}}$

       Base Add   Add

      100 104 108 112 116 120 124 128 132 By

a | 10 | 20 | 30 | 40 | 50 | 60 | . | . | . |
    0 1 2 3 4 5 6 7 8 9   element

int x ;

    100 ← add

a | | ← value   index

printf ("%d ", a[0]) ; → 10

$$\text{int } arr[5] = \{10, 20, 30, 40, 50\};$$

n | 5

arr | 10 | 20 | 30 | 40 | 50
0 | 1 | 2 | 3 | 4

```
for(i=0; i<n; i++)
{
    printf("%d", arr[i]);
}
```

```c
#include<stdio.h>
int main()
{
    int arr[10],n,i;
    printf("Enter number of elements:");
    scanf("%d",&n);
    //input
    for(i=0 ; i<n; i++)
    {
        printf("Enter value of %d element:",i+1);
        scanf("%d",&arr[i]);
    }
    //output
    for(i=0 ; i<n; i++)
    {
        printf("%d\t",arr[i]);
    }
    return 0;
}
```

H.W

Reverse array

arr | 1 | 2 | 3 | 4 | 5   Rev   arr | 5 | 4 | 3 | 2 | 1
0 | 1 | 2 | 3 | 4   →   0 | 1 | 2 | 3 | 4