Date Page 28 Jan 20

UNIT NO. 1 ARRAY

Array :- It is a Collection of Similar type of Data elements or it is the Collection of Similar type of Data types.

It is a group which define the Collection of Similar elements.

It is a Collection of Homogenious types of element.

eg. int a, b, c

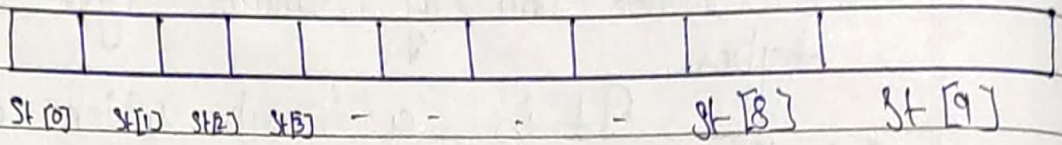
int a [3]

Here, array main is a User define Name, [ ] is a array bracket which is also called Subscript bracket.

Subscript bracket define the Location of Array element in the array.

Define an array name Student, Size/Number of array element 10, array data type int also draw their memory location.

int Student [10]



Array Declaration :- Array element can be declared as an Singal Variable.

Syntax :-

datatype arrayname [array length]

eg. int z[5]

int a[10]

int c[9]

Array Initialisation :- An array element can initialise as a single Variable but difference is there are need of more than One Value at a time for the element of Array.

Datatype arrayname [array length]

{ Value }

Date Page 25 Jan 20

int st [5] = { 6, 7, 3, 2, 1 }

6	7	3	2	1
st [0]	st [1]	st [2]	st [3]	st [4]

Define a array name employee Contain 10 element
↳ Contain the value 10 to 19
form for there locations.

int employee [10] ;

10	11	12	13	14	15	16	17	18	19
em [0]	em [1]	em [2]	em [3]	em [4]	em [5]	em [6]	em [7]	em [8]	em [9]

Type of Array Array are Two Type.

1) Single Dimancial Array
2) Multi Dimancial Array

Single Dimancial Array :- Single Dimancial array are those Array which Contain only one Subscript.

eg. int a [ ]

Multi Dimancial Arrays:- Those Array which contain More than One Subscript is called Multi dimancial Array. It may be 2D, 3D or so on.

eg. int a[ ][ ]...[ ];

Declaration One Dimancial Arrays:- An Array can be declared same as a Single Variable.

eg. int a[5] , int a[3]

int a[7] , int z[4]

Here, Array Name denote the Name of the array it can be any Valid Identifire. The Size of Array specifie the Number of element that can be store in the Array. It can be a positive integer constant. When array element declare Compiler allocate space in Memory Sufficient to Hold out all the element of the Array.

Here Compiler should Know the Size the array at the Compile time.

Initialisation of One Dimancial Array:-

After Declaration the element of array Contain the Garbage Value So

Date Page 29/01/20

There are need to some value of the element. For this purpose we use the Initialisation.

Syntax :- Data type array name [array length] = { Value }

int a[5] = { 1, 2, 3, 6, 7 };

- If the size is not define during initialization the compiler assume the size of array equal to the number of Initializer.
- During a initialization if the Number of initializer is less than the size of array. All the remaining element of the array or assign the value zero.
- If the Number of initializer more than the size of array. then Compiler will Generate error (Run time).

Processing on One Dimantional Array :-

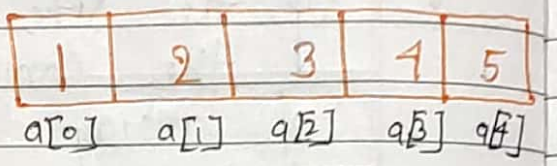
For Processing array we use a for loop & loop variable is used with in the Subscript. There are two type of Variable.

1). Reading or take Value.
2). Display or print Value.

```
int a[5] = {1,2,3,4,5}
For (i=0; i<5; i++)
{
 Scanf ("%d", &a[i]);
}
```

Que. WAP to input the values into an array.

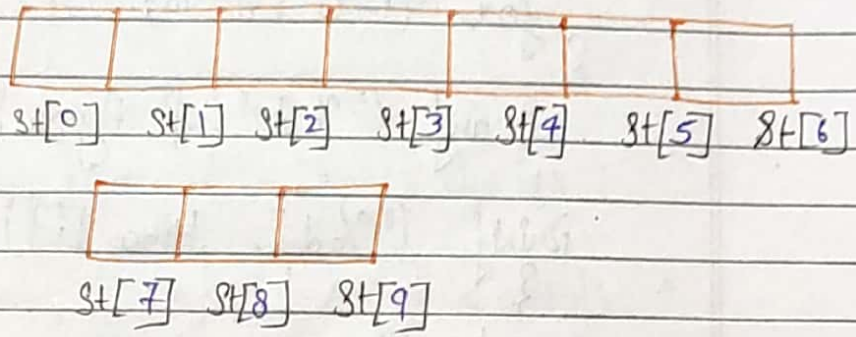
```
#include <Stdio.h>
#include <Conio.h>
  Void main ()
{
  int i; a[5];
 Printf ("Input");
  for (i=0; i<5; i++)
  {
 Scanf ("%d", &a[i]);
  }
  getch ();
  }
```



Que. WAP to declare an array name - Student, Size - 10 & input the values in the array.

```
#include <Stdio.h>
#include <Conio.h>
  Void main ()
{
  int St[10];
 Printf ("input");
 for (i=0; i<10; i++)
```

```
{
  scanf ("%d" & st[i]);
}
getch ();
}
```



Que WAP to declare an array name - Student, size - 10 input the value in the array.

```
#include <stdio.h>
#include <conio.h>
  Void main( )
  {
    int  i, st [10];
    printf (" input ");
    for (i=0; i<10 ; i++)
    {
      scanf ("%d", &st [i]);
    }
    getch ();
  }
```

Que WAP to declare an array of size 10, name Ayush & print the value of that array.

```
# include <stdio.h>
# include <conio.h>
```

```
Void main ( )
{
int Ayush [10];
print ("input");
for (i=0 ; i<10; i++)
{
Scanf ("%d", & Ayush [i]);
printf ("%d", Hiba [i]);
}
getch ();
}
```

Que. WAP to declare an array name = employee, Size = 20, data type int input value.

```
# include <stdio.h>
# include <conio.h>
Void main ()
{
int emp [20];
printf ("input");
for (i=0 ; i< 20; i++)
{
Scanf ("%d", & emp [i]);
printf ("%d", emp [i]);
}
getch ();
}
```

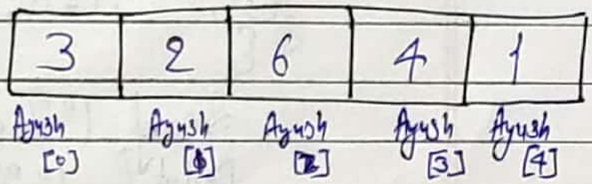
Que. WAP to declare an Array of Size-10, name employee, input the value in the array &

Que Print the addition of the element of the array.

```
# include <Stdio.h>
# include <Conio.h>
  Void main ()
{
int emp[10], Sum = 0;
for (i=0; i<10; i++)
{
  Scanf ("%d", & emp [i]);
Sum = Sum + emp [i];
}
 Printf ("%d", Sum );
 getch ();
}
```

Que.

```
# include <Stdio.h>
# include <Conio.h>
  Void main()
{
  int i, Ayush[5];
  for (i=0; i<5; i++)
  {
  Scanf ("%d", & Ayush [i]);
  printf ("%d", Ayush [i]);
  }
  getch ();
}
```



```
#include <Stdio.h>
#include <conio.h>
  Void  main ();
{
  int i, Ayush 2 [10];
for (i=0 ; i<10 ; i++)
  {
  Scanf ("%d", & Ayush 2 [i]);
  Printf ("%d" , Ayush 2 [i]);
  }
  getch();
}
```

1	2	3	4	5	6

7	8	9	10

Que.

```
#include <Stdio.h>
#include <conio.h>
  Void main ()
{
  int i, Ayush [3][20] ;
  for     (i=0; i<20 ; i++)
  {
    Scanf ("%d", & Ayush3 [i] );
  Printf ("%d", Ayush 3 [i] );
  }
  getch ();
}
```

3	4	5	8	9	2	6	2	8	1	3	5	4	5

0	0	8	9	2	1

Que. W.A.P to declare of array size 5 array name is employee (. Input the values in the array & count the even or odd Number in the Array.

```
for (i=0 ; i<5 ; i++)
{
scanf ("%d", &st[i]);
if (st[i] % 2 == 0).
{
    even = even + 1;
else
 odd. = odd + 1;
 }
 }
```

Que. WAP to declare an array Size - 5, Initialise the array element and find the maximum Value & or minimum value in the array.

Date Page 3 Feb 20

```
#include <stdio.h>
#include <conio.h>
  Void main()
{
  int a[5], min, max, i;
   for (i=0; i<5; i++)
  {
  Scanf ("%d", &a[i]);
  }
   min = max = a[0];
  for (i=0; i<5; i++)
  {
  if (a[i] < min)
  min = a[i];
  if (a[i] > max)
   max = a[i];
  }
   printf ("%d %d", min, max);
   getch();
   }
```

Array and function :- Array can be work with the function by two way.

1). **Passing Individual array element to the function.**

Array element is treated as a single variable in the program. we can pass separate array element as a argument

Date Page 3 feb 20

to a function.

```
#include <stdio.h>
#include <conio.h>
    void main ()
{
int a [5], i;

for (i=0; i<5; i++)
{
  scanf ("%d", &a [i]);
  check (a [i]);
  getch ();
}
}
Void check (int num)
{
if (num % 2 == 0)
 printf ("even");
 else
 printf ("odd");
}
```

void check (int);

6	5	4	3	2
a[0]	a[1]	a[2]	a[3]	a[4]

2). Passing whole One Dimencial Array to the function.

We can pass whole array as an actual Argument to a function the Corresponding Formal Argument should be declared as an array variable of same Data type.

```
Void fun (a[]);
# include <stdio.h>
# include <conio.h>
   Void main ()
{
   int a [6] = { 6,5,4,3,2,1 };
   fun (a);
   getch ();
}

   Void fun (int a[])
{
   int sum = 0, i;
   for (i = 0; i<6; i++)
{
   Sum = Sum + a [i];
}
   printf ("%d", Sum);
}
```

Multi Dimancial Array :- Those Array which Contain more than One Subscript like 2, 3 is called multi-dimancial Array.

eg. 2D array contain 2 Subscript.

3D array contain 3 Subscript. & so on...

Declaration of 2D (2 Dimancial) Array -

Syntax :- datatype arrayname [row] [Column];

eg. int mat [3] [3];

Initialization of 2D Array -

A two dimancial Array can be initialise same as Single dimancial Array.

Syntax :- datatype arrayname [row] [column] = {Values};

eg. int mat [3] [3] = {1, 2, 3, 6, 7, 8, 9, 10, 2};

Memory Allocation of 2-D Array :-

eg. int mat [3] [3] = {1, 2, 3, 6, 7, 8, 9, 10, 3};

	0	1	2
0	00 1	01 2	02 3
1	10 6	11 7	12 8
2	20 9	21 10	22 3

Que. WAP to declare of Array of size 10 An input the value in the array & Search.

```
# include <stdio.h>
# include <conio.h>
   Void main ()
{
   int a[10], i, n;
Scanf ("%d", &n);

for (i=0; i<10; i++)
{
   if (n== a[i])
{
   Printf ("Found");
}
   Continue;
}
   getch();
}
```

Processing on 2D Array

There are two type of operation on the array
1) reading values to the array
2) printing the values of the array

WAP to input the values in 2D array of size 4x4, type integer, name - student input the values in the array & display the value of the array.

```
# include <Stdio.h>
# include <conio.h>
void main()
{
int st[4][4], i;
st[4][4] = {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};
```

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16

```
for(i=0; i<4; ++i)
  { for(int j=0; j<4; ++j)
    { printf("%d", st[i][j]);
  }
~~printf("%d", st[4][4]);~~ }
getch();
}
```

WAP to input the values in 2D array size 3x3 type int name st. input the values in the array & display.

```
# include <Stdio.h>
# include <conio.h>
void main()
```

```
{
int a[3][3] i,j;
for(i=0; i<3; i++)
{
for (j=0 ; j<3; j++)
{
scanf ("%d", &a[i][j]);
}
}
for(i=0; i<3; j++)
{
for(j=0; j<3; j++)
}
printf("%d", a[i][j]);
}
getch();
}
```

WAP to declare an array 5x5 name employee initialized the value in array & print

```
# include <stdio.h>
# include <conio.h>
void main()
{
int emp[5][5], i,j;
for(i=0; i<5; i++)
{
for(j=0; j<5; j++)
{
scanf("%d", & emp[i][j]);
}
}
```

```
for (i=0; i<5 ;j++)
{
for(j=0; j<5; j++)
{
printf ("%d", emp[i][j];
}
}
getch();
}
```

WAP size, 4x3, int give value to array & print them.

```
# include <Stdio.h>
# include <conio.h>
void main()
{
int a[4][3], i,j;
for(i=0; i<4; i++)
{
for(j=0; j<4 ;j++)
{
fo scanf("%d", &a[i][j];
}
}
for(i=0 ; i<4; i++)
{
for(j=0; j<4; j++)
{
printf("%d", & a[i][j];
}
}
getch();
}
```

WAP to declare an array, size 5x6, st (name) & print them.

```
# include <stdio.h>
# include <conio.h>
void main()
{
int st[5][6], i, j;
for(i=0; i<5; i++)
{
for(j=0; j<6; j++)
{
scanf("%d", &st[i][j]);
}
}
for(i=0; i<5; i++)
{
for(j=0; j<6; j++)
{
printf("%d", st[i][j]);
}
}
getch();
}
```

WAP to print the addition of two matrices of same size.

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

```
int a[4][4], b[4][4], c[4][4], i,j;
for (i=0 ; i<4; i++)
{
for (j=0 ; j<4 ;j++)
{
scanf ("%d", &a[i][j]);
scanf ("%d", &b[i][j]);
}
}for( i=0; i<4; i++)
{
for (j=0 ;j<4; j++)
{
c[i][j]= a[i][j] + b[i][j];
}
}
printf ("%d", c[i][j]);
getch();
}
```

WAP to declare an array 4x5, name - employee type - integer, initialize the value in the array & print them.

```
# include <Stdio.h>
# include <Conio.h>
void main()
{
int emp[4][5], i,j;
for(i=0; i<4; i++)
{
for ( j=0; j<5 ; j++)
{
```

AYUSH

```
scanf ("%d", &emp[i][j]);
}
}
for (j=0; j<4; i++)
{
for (j=0; j<5; j++)
{
printf ("%d", emp[i][j]);
}
}
getch();
}
```

WAP to declare an array, 6x6, name - student input the value in the array & print them.

```
#include <stdio.h>
#include <conio.h>
void main()
{
int stu[6][6],i,j;
for(i=0; i<6; i++)
{
for(j=0; j<6; j++)
{
scanf ("%d", &stu[i][j]);
}
}
for(i=0; i<6; i++)
{
for(j=0; j<6; j++)
{
printf ("%d", stu[i][j]);
}
}
```

getch();

WAP to the transpose of the matrix.

```
# include <stdio.h>
# include <conio.h>
void main();
{
int mat1[r][c], mat2[r][c], i, j;
scanf ("%d%d", &c, &r);
for(i=0; i<r; i++)
{
for(j=0; j<c; j++)
{
scanf ("%d", &mat1[i][j]);
}
}
for(i=0; i<c; i++)
{
for(j=0; j<r; j++)
{
mat2[i][j]= mat1[i][j];
}
}
printf ("transpose");
for(i=0; i<c; i++)
{
for(j=0; j<r; j++)
{
printf("%d", mat2[i][j]);
}
}
getch();
}
```

Difference between Array & Structure.

Array	Structure
Array is collection of Homogenous data.	Structure is collection of Heterogeneous Data.
Array data are access using index	Structure element are using 'access using Operator
Array Allocates Static memory	——Dynamic memory
Time less	Time Taking