

UNIT - III

ARRAYS AND STRINGS

ARRAY

Collection of items of similar datatype.

Syntax: datatype var_name [size];

Initialization

- * Compile Time Initialization
- * Run Time Initialization.

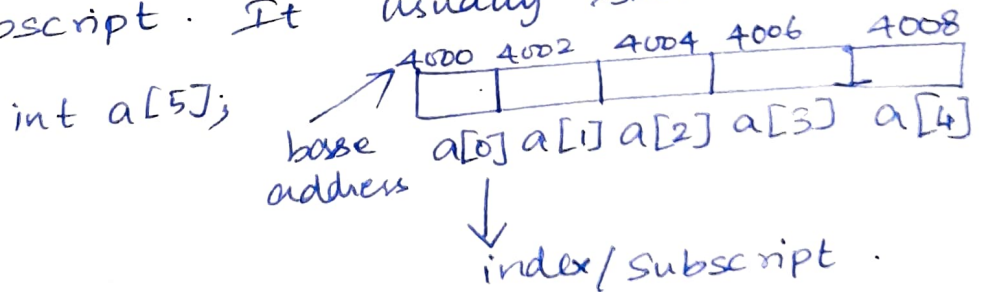
Compile Time:

```
int a[5] = {5, 10, 15, 20, 25};
```

Run Time

```
int a[5];  
for (int i=0; i<5; i++)  
scanf ("%d", a[i]);
```

* Array elements can be accessed through Index/subscript. It usually starts with 0.



* Memory allocation for array is done contiguous. The address of a[0] is the base address.

* Types of Array:

- 1 Dimensional array
- 2 Dimensional array
- Multi Dimensional array.

① Searching an element in given array

main()

{

int a[5], i, k

printf("\n Enter the array Elements");

for (i=0; i<5; i++)

scanf("%d", &a[i]);

printf("\n Enter the element to search");

scanf("%d", &k);

printf("\n ... searching");

for (i=0; i<5; i++)

{

if (k == a[i])

{

found = 1;
break;

printf("\n Element found at %d position", i+1);

exit(0);

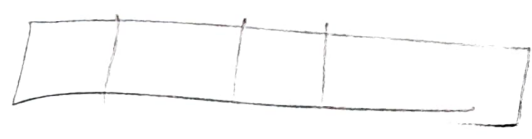
}

if (found == 1)
pf("found");

printf("\n Element Not found");

else }

pf("Not found");



② Sorting the elements in given array.

main ()

{

int a[5], i, j

printf ("In Enter array elements");

for (i=0; i<5; i++)

scanf ("%d", &a[i]);

printf ("\n.... Sorting");

for (i=0; i<4; i++)

{

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

{

if (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

} }

}

printf ("Sorted array is:");

for (i=0; i<5; i++)

{

printf ("%d", a[i]);

}

}

③ Sum the elements in an array.

main ()

{

int a[5], i, sum = 0;

printf ("In Enter the array elements");

for (i = 0; i < 5; i++)

scanf ("%d", &a[i]);

for (i = 0; i < 5; i++)

sum = sum + a[i];

printf ("In sum of array ele is %d", sum);

}

Note: example.

Find the longest consecutive even numbers
in the given array.

int a[] = {1, 2, 3, 4, 6, 8, 9, 10, 12, 14, 16, 1, 3, 5, 7};

④ Minimum element in the given array.

```
main ( )
```

```
{
```

```
int a[10], i, min;
```

```
printf ("Enter the array elements");
```

```
for (i=0; i<10; i++)
```

```
scanf ("%d", &a[i]);
```

```
min = a[0];
```

```
for (i=1; i<10; i++)
```

```
{
```

```
if (min > a[i])
```

```
min = a[i];
```

```
}
```

```
printf ("The min ele. in array is %d", min);
```

```
}
```

⑤ Maximum element in the given array.

```
max.
```

```
max < a[i]
```

Two Dimensional array.

$$\begin{matrix} & \begin{matrix} 0 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \end{matrix} & \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} \end{matrix}$$

```
int a [ ] [ ];
```

To get the values from user @ run time:

```
main ()
```

```
{
```

```
    int a [3] [3], i, j
```

```
    printf ("In Enter the matrix elements")
```

```
    for (i=0; i<3; i++)
```

```
    {
```

```
        for (j=0; j<3; j++)
```

```
        {
```

```
            scanf ("%d", &a [i] [j]);
```

```
        }
```

```
    }
```

```
}
```

Matrix Addition/Subtraction

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 4 \\ 6 & 8 \end{bmatrix}$$

main ()

{

int a[3][3], b[3][3], c[3][3], i, j;

printf ("Enter the elements of A matrix");

for (i=0; i<3; i++)

{

for (j=0; j<3; j++)

{

scanf ("%d", &a[i][j]);

}

}

printf ("Enter the elements of B matrix");

for (i=0; i<3; i++)

{

for (j=0; j<3; j++)

{

scanf ("%d", &b[i][j]);

}

} printf ("C matrix is");

for (i=0; i<3; i++)

{

for (j=0; j<3; j++)

{

c[i][j] = a[i][j] + b[i][j];

printf ("%d", c[i][j]);

}

}

}

Matrix Multiplication

```
main()
```

```
{ int a[3][3], b[3][3], c[3][3], i, j, k
```

```
printf("\n Enter the A matrix elements");
```

```
for (i=0; i<3; i++)
```

```
{ for (j=0; j<3; j++)
```

```
{ scanf("%d", &a[i][j]);
```

```
    }
```

```
printf("\n Enter the B matrix elements");
```

```
for (i=0; i<3; i++)
```

```
{ for (j=0; j<3; j++)
```

```
{ scanf("%d", &b[i][j]);
```

```
    }
```

```
}
```

```
printf("\n Matrix Multiplication");
```

```
for (i=0; i<3; i++)
```

```
{
```

```
    for (j=0; j<3; j++)
```

```
    { c[i][j]=0;
```

```
      for (k=0; k<3; k++)
```

```
      { &c[i][j]=c[i][j]+a[i][k]*b[k][j];
```

```
        printf("%d", c[i][j]);
```

```
      } printf("\n");
```

```
    }
```

```
}
```