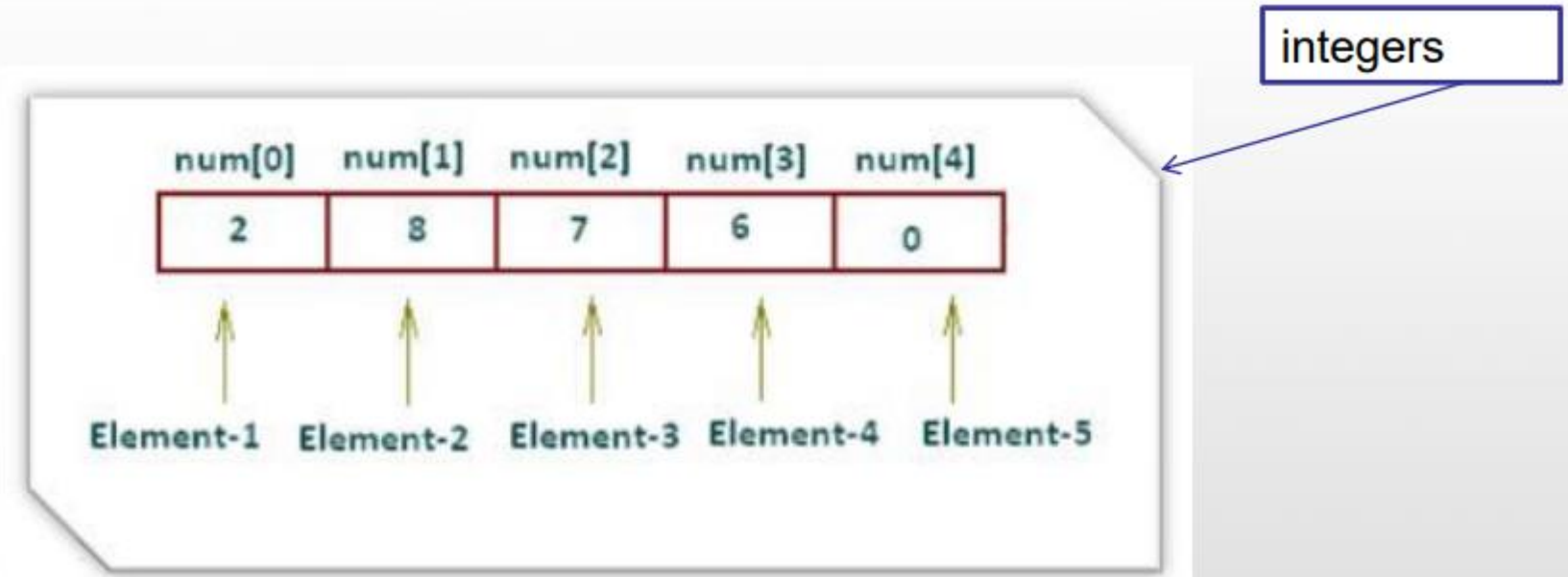
Arrays

Comp230

Arrays

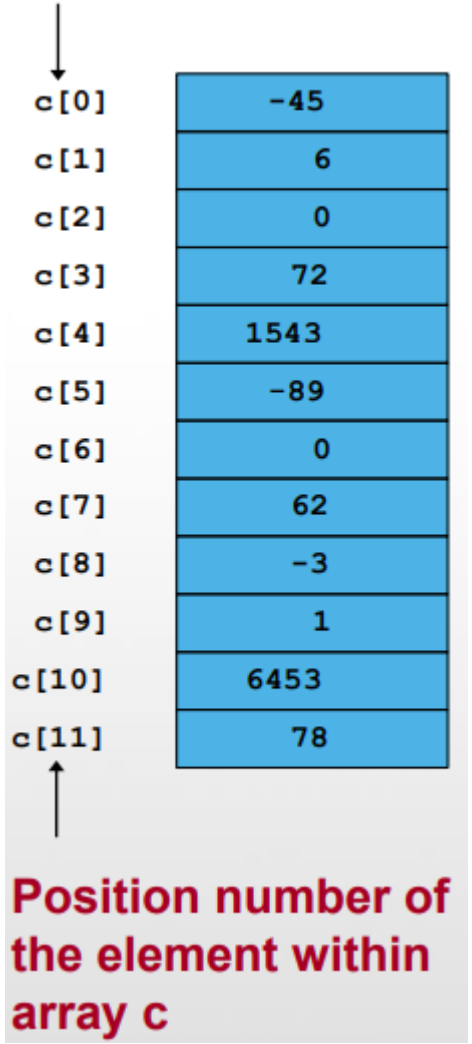
- Array is a collection of data items of the same type.
- Array element is a data item that is part of an array.



Arrays

- Array
  - Group of consecutive memory locations
  - Same name and type
- To refer to an element, specify
  - Array name
  - Position number
- Format:

  arrayname[position number]
  - First element at position 0
  - n element array named c:
    - c[0], c[1]...c[n - 1]



Declaring Arrays

- When declaring arrays, specify

  arrayType arrayName[numberOfElements];

  e.g. int c[10];

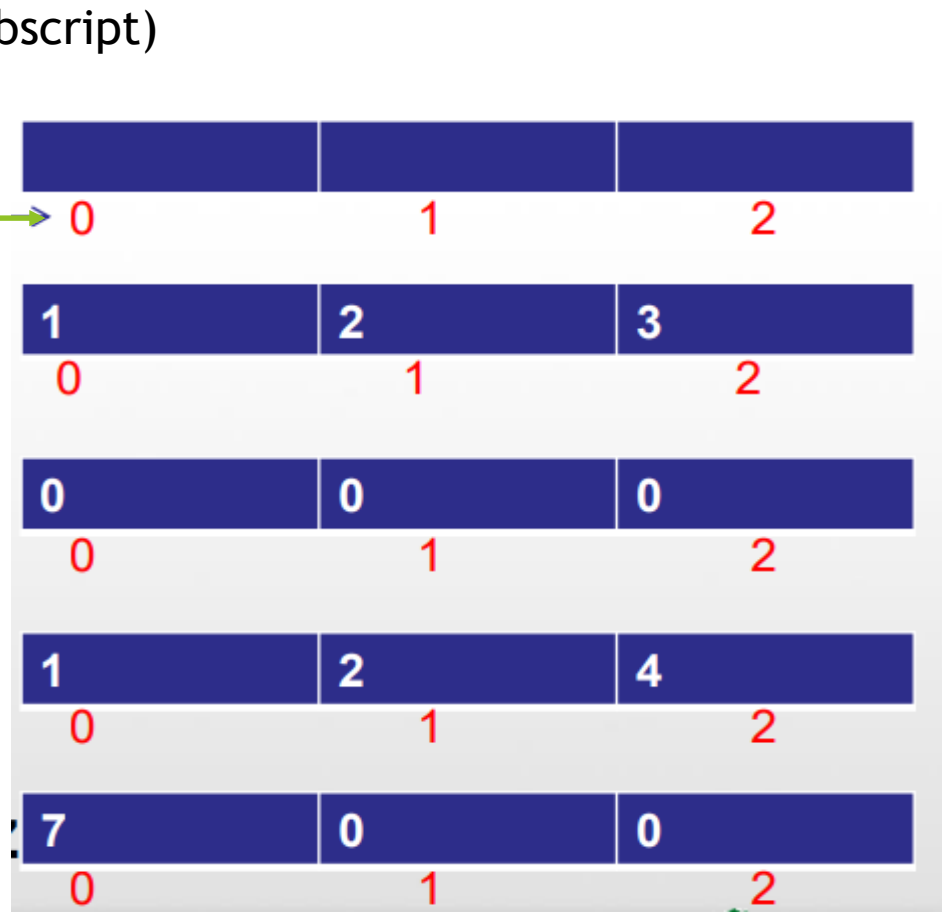
  float myArray[100];

- Declaring multiple arrays of same type
  - Format similar to regular variables

    e.g. int b[100], x[27];

Declaring Arrays

- int x [3];
- int val[3]={1,2,3};
- int y[3]={0};
- int m[ ]={1,2,4};
- int z[3]={7};



Arrays

- Array elements are like normal variables

```
c[ 0 ] = 3;
printf( "%d", c[ 0 ] );
c[1]= c[0]+c[2]
c[3]= c[2]+5
```

- Perform operations in subscript (index).

```
c[ 5 - 2 ] == c[ 3 ] == c[ x ]
```

Examples Using Arrays

- Initializers

  int n[5] = { 1, 2, 3, 4, 5 };

  char alphabet[5] = {‘A’ , ’B’ , ’C’ , ’D’ , ’E’};

- All elements 0

  int n[5] = { 0 }

- If size omitted, initializers determine it

  int n[ ] = { 1, 2, 3, 4, 5 };

  5 initializers, therefore 5 element array

Examples Using Arrays

```
int a [5] = {5,2,9,10,31};
int result = a[3%2] + a[2]+a[4/2];
printf("%d\n",result);
printf("%d",a[5%3]);
```

Output:

20

9

```
int a [5] = {5,2,9,10,31};
int temp;
printf("%d   %d",a[0], a[4]);
temp=a[0];
a[0]=a[4];
a[4]=temp;
printf("\n%d   %d",a[0], a[4]);
```

Output:

5 31

31 5

Example: Fill and Print Array

```c
#include <stdio.h>

int main ()
{
   int n[ 10 ];  // n is an array of 10 integers
   int i,j;

    // initialize elements of array n _ _ _ (Fill Array)
   for ( i = 0; i < 10; i++ )
   {
      n[ i ] = i + 1; /* set element at location i to i + 1 */
   }

    // output each array element's value (Print Array)
   for (j = 0; j < 10; j++ )
   {
      printf("Element[%d] = %d\n", j, n[j] );
   }

   return 0;
}
```

Output:

```
Element[0] = 1
Element[1] = 2
Element[2] = 3
Element[3] = 4
Element[4] = 5
Element[5] = 6
Element[6] = 7
Element[7] = 8
Element[8] = 9
Element[9] = 10
```

Example: Fill and Print Array

```c
#include <stdio.h>
#define size 5 //   array size= 5
int main ()
{
   int n[ size ]; //  n is an array of 5 integers
   int i,j;

   //  initialize elements of array n (Fill Array)
   for ( i = 0; i < size; i++ )
   {
      scanf ("%d",&n[ i ]);
   }

   //  output each array element's value (Print Array)
   for (j = 0; j < size; j++ )
   {
      printf("Element[%d] = %d\n", j, n[j] );
   }

   return 0;
}
```

Input:

```
1 2 3 4 5
```

Output:

```
Element[0] = 1
Element[1] = 2
Element[2] = 3
Element[3] = 4
Element[4] = 5
```

Examples

- **Fill and print array using function**

```
#include <stdio.h>
void fillArray(int[], int);
void printArray(int[], int);
int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int myArray[size];
    fillArray(myArray, size);
    printArray(myArray, size);
    return 0;
}
```

```
void fillArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("Enter value for element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
 }
void printArray(int arr[], int size) {
    printf("Elements of the array: \n");
    for (int i = 0; i < size; i++) {
        printf("arr[%d] = %d\n",i, arr[i]);
    }
}
```

Examples

- **Inverse Array using function**

```
#include <stdio.h>
void swap(int *, int *);
void inverseArray(int [], int );
void printArray(int [], int );
int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int myArray[size];
    printf("Enter elements of the array:\n");
    for (int i = 0; i < size; i++) {
      printf("Enter value for element %d: ", i + 1);
      scanf("%d", &myArray[i]);
    }
    inverseArray(myArray, size);
    printArray(myArray, size);
    return 0;
}
```

```
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
void inverseArray(int arr[], int size) {
    int start = 0;
    int end = size - 1;
    while (start < end) {
        swap(&arr[start], &arr[end]);
        start++;
        end--;
    } }
void printArray(int arr[], int size) {
    printf("Inversed Array: \n");
    for (int i = 0; i < size; i++) {
        printf("arr[%d] = %d \n", i, arr[i]);
    }
}
```

Examples

- **sum two arrays**

```
#include <stdio.h>
void sumArrays(int [], int [], int [], int);
void printArray(int [], int);
int main() {
    int size;
    printf("Enter the size of the arrays: ");
    scanf("%d", &size);
    int array1[size], array2[size], sumResult[size];
    printf("Enter elements of the first array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array1[i]);
    }
    printf("Enter elements of the second array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &array2[i]);
    }
 sumArrays(array1, array2, sumResult, size);
   printf("First Array: ");
    printArray(array1, size);
    printf("Second Array: ");
    printArray(array2, size);
    printf("Sum Result: ");
    printArray(sumResult, size);
    return 0; }
void sumArrays(int arr1[], int arr2[], int result[], int
size) {
    for (int i = 0; i < size; i++) {
        result[i] = arr1[i] + arr2[i];
    } }
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

Examples

- sort array

```
#include <stdio.h>
void sortArray(int [], int);
void printArray(int [], int);
int main() {
    int size;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int myArray[size];
    printf("Enter elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &myArray[i]);
    }
    sortArray(myArray, size);
    printf("Sorted Array: ");
    printArray(myArray, size);
    return 0;
}
```

```
void sortArray(int arr[], int size) {
    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - i - 1; j++) {
            if (arr[j] > arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;  }
        }
    }
}
void printArray(int arr[], int size) {
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}
```

Example: Finding the Maximum

```c
#include <stdio.h>
#define size 5
int main()
{
    int i,max;
    int list[size];
    //initialize the array
    for (i=0;i<size;i++)
        scanf("%d",&list[i]);
    //find maximum value
    max=list[0];
    for (i=1;i<size;i++)
        if (max<list[i])
         max=list[i];
    printf("Maximum value:%d",max);
    return 0;
}
```

Example: sorting it in descending order

```
void Sort(int array[])
{
     int i,j;
     int temp;
     for(i=0;i<Size-1;i++)
     {
          for (j=i+1;j<Size;j++)
          {

            if (array[i]<array[j])
            {
                temp=array[j];
                array[j]=array[i];
                array[i]=temp;
            }
          }
     }
}
```

```
Enter array of integers with size 3
3 4 5
array after sorted :5 4 3
```

Linear Search

- Problem: Given a list of N values, determine whether a given value X occurs in the list.
- For example, consider the problem of determining whether the value 55 occurs in:

1	2	3	4	5	6	7	8
17	31	9	73	55	12	19	7

- Solution:

  start at one end of the list, if the current element doesn't equal the search target, move to the next element, stopping when a match is found or the opposite end of the list is reached.

Code

```
#include <stdio.h>
int linearSearch(int [], int, int) ;
int main() {
    int size, key;
    printf("Enter the size of the array: ");
    scanf("%d", &size);
    int myArray[size];
    printf("Enter elements of the array:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &myArray[i]);
    }
    printf("Enter the key to search: ");
    scanf("%d", &key);
    int result = linearSearch(myArray, size, key);
```

```
 if (result != -1) {
        printf("Key %d found at index %d.\n", key, result);
    } else {
        printf("Key %d not found in the array.\n", key);
    }
    return 0;
}
int linearSearch(int arr[], int size, int key) {
    for (int i = 0; i < size; i++) {
        if (arr[i] == key) {
            return i;
        }
    }
    return -1;
}
```

Example

- Write a program that takes 7 integers as input and prints the number with the smallest sum of digits and its location in the array.

```c
#include <stdio.h>

int sumOfDigits(int);

int main() {
    const int size = 7;
    int numbers[size];
    printf("Enter 7 integers:\n");
    for (int i = 0; i < size; i++) {
        scanf("%d", &numbers[i]);
    }
    int smallestSum = sumOfDigits(numbers[0]);
    int smallestIndex = 0;
    for (int i = 1; i < size; i++) {
        int currentSum = sumOfDigits(numbers[i]);
        if (currentSum < smallestSum) {
            smallestSum = currentSum;
            smallestIndex = i;
        }
    }
    printf("Number with the smallest sum of digits: %d\n", numbers[smallestIndex]);
    printf("Location in the array: Index %d\n", smallestIndex);
    return 0;
}

int sumOfDigits(int num) {
    int sum = 0;
    while (num != 0) {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}
```

Creating a 2D Array

- Create array elements by telling how many ROWS and COLUMNS

Example:

```
int grades[5][3];
```

- grades is a two-dimensional array, with 5 rows and 3 columns. One row for each student. One column for each test.

Example

int a[2][4];

a[1][0]=9;

a[0][3]=5;

a[0][1]=a[0][3]+ a[1][0];

Declare & Initialize

- Example:

```
int grades[5][3] = { { 78, 83, 82 },
                     { 90, 88, 94 },
                     { 71, 73, 78 },
                     { 97, 96, 95 },
                     { 89, 93, 90 } };
```

- A Two-D Array is an array of arrays. Each row is itself a One-D array.

Row, Column Indices

	0	1	2
0	78	83	82
1	90	88	94
2	71	73	78
3	97	96	95
4	89	93	90

Give both the ROW and COLUMN indices to pick out an individual element. The fourth student's third test score is at ROW 3, COLUMN 2

Example : Fill Array

What are the elements of the **array** table?

```
int table[3][4];
int x = 1;
for   (row = 0; row < 3; row++)
      for   (col = 0; col < 4; col++)
      {
            table[row][col] = x;
            x++;
      } //for col
```

Example

- Write a program that adds up two 2x2 arrays and stores the sum in third array.

```c
#include <stdio.h>
void addArrays(int [][2], int [][2], int [][2]);
void printArray(int [][2]);
int main() {
    int array1[2][2], array2[2][2], sumArray[2][2];
    printf("Enter elements of the first 2x2 array:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            scanf("%d", &array1[i][j]);
        }
    }
    printf("Enter elements of the second 2x2 array:\n");
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            scanf("%d", &array2[i][j]);
        }
    }
    addArrays(array1, array2, sumArray);
    printf("First Array:\n");
    printArray(array1);
    printf("Second Array:\n");
    printArray(array2);
    printf("Sum Array:\n");
    printArray(sumArray);
    return 0; }
void addArrays(int arr1[][2], int arr2[][2], int result[][2]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            result[i][j] = arr1[i][j] + arr2[i][j];
        }
    } }
void printArray(int arr[][2]) {
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            printf("%d ", arr[i][j]);
        }
        printf("\n");     }
}
```