# Array

Comp 103: structure programming

# Array

```
int x ;
x = 5 ;
x = 10 ;
printf ( "\n x = %d", x ) ;
```

How can we store more than one value at a time in a single variable?

# Array

- Array is collection of similar items
- These items can be int, char or float.
- All items in array must be of same data type.
- Array must be declare before use.

# Array declaration

- Integer array

```
int   a [64];
```

Data type

Array Dimension / array size

- Character array

```
char alpha[10];
```

- Float array

```
float num[5];
```

# Assigning value to array

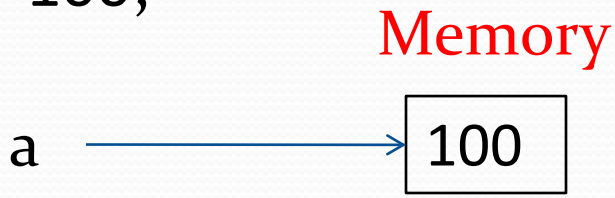
```
int a[4];
a[0]=50;
a[1]=60;
a[2]=20;
a[3]=10;
```

- Note :-Elements of array has a name which is called array index

# Assigning value to array

- int a;

  a=100;



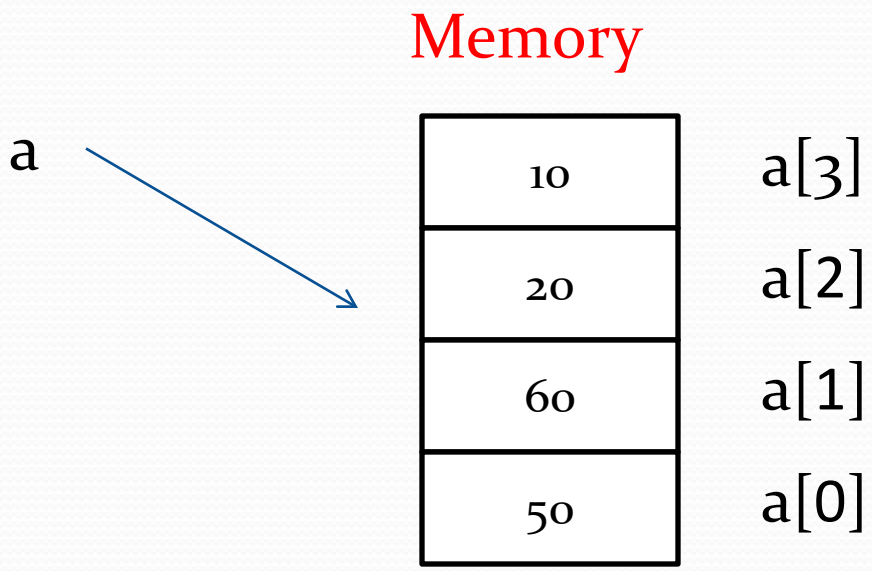

- int a[4];

  a[0]=50;

  a[1]=60;

  a[2]=20;

  a[3]=10;



# Entering Data into an Array

```
int marks[64];
int i;
for ( i = 0 ; i <= 63 ; i++ )
{
   printf ( "\n Enter marks  for  number %d :" , i ) ;
   scanf ( "%d", &marks[i] ) ;
}
```

# Reading Data from an Array

```
for ( i = 0 ; i <= 63 ; i++ )
{
   printf ( "\n Marks of no %d :  %d" , i , marks[i]) ;
}
```

- This will print marks of all 64 students

# Initializing An Array

- Initializing integer

  int a=10;

- Initializing array

  int a[4]={50,60,10,40};

  int n[ ] = { 2, 4, 12, 5, 45 } ;

# Program to store marks of 5 students and get average

```
int marks[5];
int i,avg,sum=0;

for(i=0; i<5; i++)
{
        printf("Mark of student %d : ",i);
        scanf("%d", &marks[i]);
}

for(i=0 ; i<5 ; i++)
{
        sum=sum + marks[i];

}
avg= sum/5;
printf("Average Mark of 5 students  = %d\n ",avg);
```

- Write program to find largest and smallest element of array **myarray**

int **myarray**[**5**] ={11, 5, 16, 25, 19}

# Write program to find largest element of array

```
int myarray[6]={11,5,16,25,55};
int i,large,small;
large=myarray[0];
 small= myarray[0];
for(i=1;  i<5 ;  i++)
{
          if(myarray[i]>large)
                    large=myarray[i];
         else if(small>myarray[i])
                    small=myarray[i];
}
printf("Largest element = %d \n",large);
printf(“Smallest element = %d \n",small);
```

# Static Array

```
main()
{
   int a[2];
   static int b[2];
   printf("a : %d  and %d",a[0],a[1]);
   printf("b : %d  and %d",b[0],b[1]);

}
```

# String

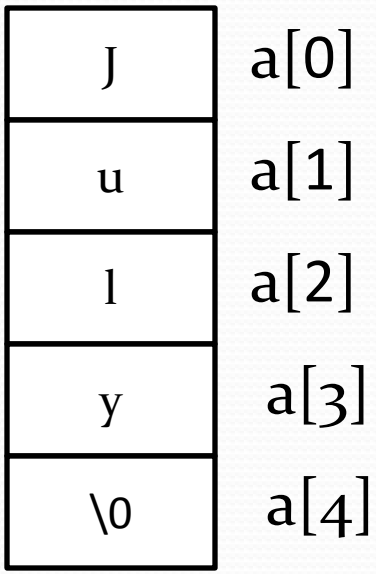
- Strings are arrays of character.
- They are enclosed with in quotes.

- Initializing strings

char month[5]={‘J’, ‘u’, ‘l’, ‘y’};

char month[]=“July”;

char month[5]={‘J’, ‘u’, ‘l’, ‘y’, ‘\0’};




Character string is terminated by null character ‘\0’

# Reading String with scanf

```
char myarray[20];
printf("Enter text :");
scanf("%s", myarray);
printf("Your Text : %s \n ", myarray);
```

Input :
String class

Output :
String

- scanf( ) is not capable of receiving multi-word strings.

# Reading String with gets

```
char myarray[100];
printf("Enter text :");
gets(myarray);

printf("Your Text : %s \n ", myarray);
```

- gets( ) is capable of receiving multi-word strings.

# Reading String with getchar

```
char myarray[20];
int i=0;
printf(“Enter text : ”);

while((myarray[i]=getchar())!='\n' && i<19)
{
    i++;
}
printf("%s",myarray);
```

# String

- Write program to count number of vowels
- Write program to count number of words in a sentence

# Program to count number of vowels

```
char myarray[100];
int i=0, count=0;
while((myarray[i]=getchar())!='\n' && i<99)
{
        if(myarray[i]=='a' || myarray[i]=='A' || myarray[i]=='e' ||
        myarray[i]=='E' || myarray[i]=='i' || myarray[i]=='I'||
        myarray[i]=='o' || myarray[i]=='O'|| myarray[i]=='u' ||
        myarray[i]=='U')
        {
                count++;
        }
        i++;
}
printf(" %s\n ", myarray);
printf("Number of words :%d \n ",count);
```

# Write program to count number of words in a sentence.

```
char myarray[100];
int i=0, count=0;
while((myarray[i]=getchar())!='\n' && i<99)
{
        if(myarray[i]==' ')
                count++;
        i++;
}
count++;
myarray[i]='\0';
printf(“ %s\n ", myarray);
printf("Number of words :%d \n ",count);
```

# Sorting

- Arranging numbers in ascending and descending order

- Bubble sort
  - A procedure for sorting a set of items that begins by sequencing the first and second items, then the second and third, and so on, until the end of the set is reached, and then repeats this process until all items are correctly sequenced.

# Bubble sort

```
int a[6]={5,7,4,1,8,2};
int i,j,temp;

for( i=0; i<6; i++)
{
      for( j=0; j<(6-1);  j++)
     {
           if(a[j] > a[j+1])
           {
               temp=a[j];
               a[j]=a[j+1];
                a[j+1]=temp;
           }
      }
}
```

```
printf("In ascending order :");

for(j=0;j<6;j++)
{
          printf("%d  ",a[j]);
}

printf("\n");
```

# Two Dimensional Array

- Two dimensional is used to store data of tables

- Declaration:

  type array_name [row_size] [col_size];

- Example :

  int arr[3][2]={1, 1, 2, 2, 3, 3};

  OR

  int arr[3][2]={{1, 1}, {2, 2} ,{ 3, 3}};

# Two Dimensional Array

- Nested for loop is used to read and write values in array

```
int a[2][2];
for(i=0;i<2;i++)
{
   for(j=0;j<2;j++)
      scanf("%d",&a[i][j]);
}
```

# Transpose of matrix

```
int arr[10][10], b[10][10];
int row, col , i , j;

printf("Enter rows and cols :");
scanf("%d %d", &row, &col);

for(i=0; i<row; i++)
{
     for(j=0; j<col; j++)
     {
        printf("arr [%d][%d] :", i, j);
        scanf("%d“ , &arr[i][j]);
      }
}
```

```
for(i=0; i<col; i++)
{
        for(j=0; j<row; j++)
              b[i][j]=arr[j][i];
}
printf("\nTranspose of A:\n");

for(i=0;i<col;i++)
{
       for(j=0; j<row; j++)
             printf("%d", b[i][j] );

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

# Multidimensional Array

- 3 or higher dimensional arrays are also allowed in c
- General form:

  type array_name [s1][s2][s3][s4]...s[n];
- Example :

```
int arr[3][3][2]={
                    {   {1,2} ,{3,4},{5,6}   },
                    {   {6,5} ,{4,3},{2,1}   },
                    {   {8,9} ,{7,0},{1,0}   }
                  };
```