Arrays

What is an array?

- An array is a consecutive series of variables that share one variable name
- The individual data items in an array must all be of the same data type, accessed using an index
- Often used when dealing with multiple data items possessing common characteristics
- Ex. 24 hourly temperature readings might be stored in array named **temperature**

1-D Array Declaration

- To declare a 1-D Array, we specify 1 size

  data_type array_name[size];

- For Eg.

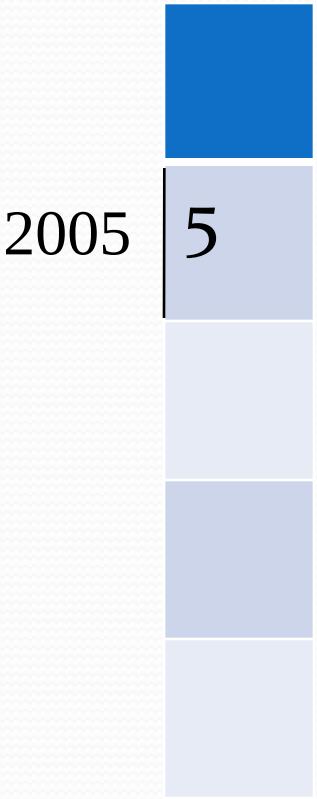
  int number[30];

  float marks[60];

  char name[15];

Single Integer

a=5;



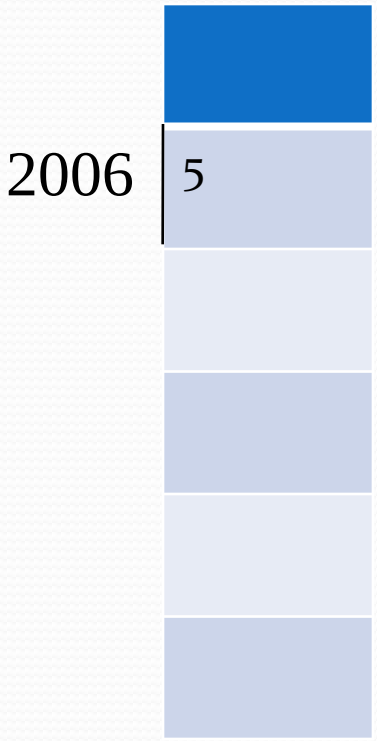
integer Array


```
int   my_array[20]={1,2,3,4,7,12,13,15,89};
```

These are the memory locations in which integer type data elements are stored.

char a;

a= ‘5’;



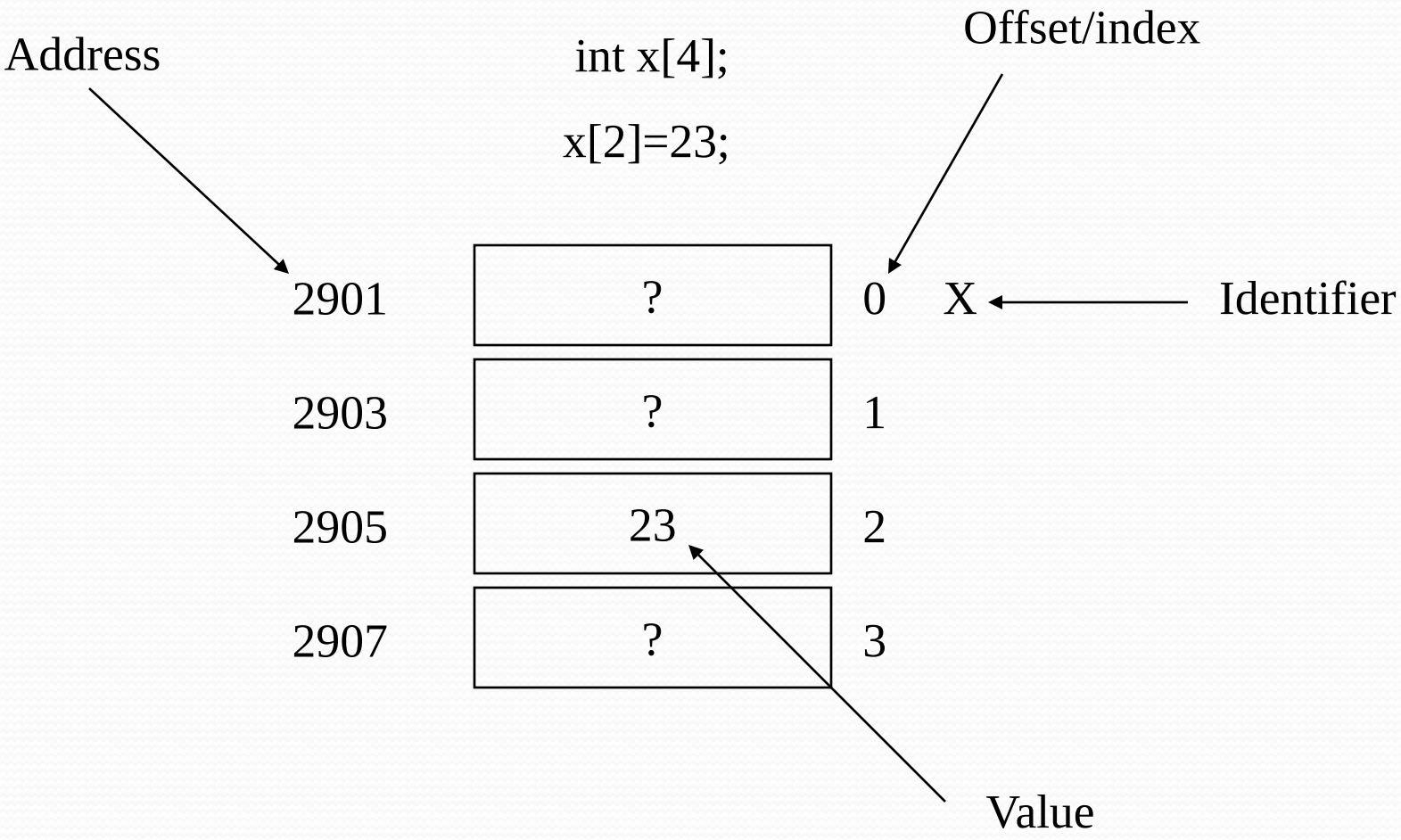
Character Array

N	I	T	R	K	L	\0
[2001]	[2002]	[2003]	[2004]	[2005]	[2006]	[2007]

char college_name[7]={‘N’,’I’,’T’,’R’,’K’,’L’,’\0’};

Each character occupy 1 byte in memory.

Visual representation



Array example

```
//Find the lowest number in the array num[10]
    main()
    {
    int num[10]={4,9,11,2,13,77,8,19,21,1};
    int min, i;
    min = num[0];
       for (i=1; i<10; i++) {
          if (num[i] < min)
                        min = num[i];
       }
    printf("The minimum is %d", min);
    }
```

Examples

int x[5] = { 11,12,13,14,15 }; size 10 bytes

- creates array with elements 0-4 values 11-15

int x[5] = {4,3}; size 10 bytes

- creates array with elements 0-4 values 4,3,0,0,0

int x[ ] = {1,2,3}; size 6 bytes

- creates array with elements 0-2 values 1,2,3

char c[4] = {‘M’, ‘o’ , ‘o’ }; size 4 bytes

- creates array with elements 0-3 values M o o NULL

Dimensionality

- Determined by the number of subscripts of an array.
  - x[i]
  - y[i][j]
  - z[i][j][k]

Some more tips..

double a[n]; ✗ Size can not be a Variable

int a[-5.2], a[-6] ✗ index always +ve integer

Convinient to use symbolic constants
#define N 100
int array_num[N];

Single vs Two-dimensional

Represented by one row of data	Represented by a table of data
Declaration: datatype name_of_array[20];	**Declaration:** datatype name_of_array[20][20];
Initialization: name_of_array[20]={1,2,3,4……16};	**Initialization:** name_of_array[20][20]={{1,2,3,4}, {2,3,4,5},{5,6,7,8}};

Examples

- Assuming we have the following array b:

	0	1
0	1	0
1	3	4

```
printf ("%d", b[0][0]);          /* prints 1 */
printf ("%d", b[1][0]);          /* prints 3 */
printf ("%d", b[1][1]);          /* prints 4 */
```

2D-Array example

```
/*print Even and Odd no's among 3x3 matrix/
 #include<stdio.h>
  main( )
{
    int x, i, j, a[i][j], even=0, odd=1;
    for( i=0;i<3; i++ )
     {
        for(j=0;j<3;j++ )
         {
                    printf( "enter a no");
                    scanf ( "%d", &a[i][j]);
           }  //end of inner for loop
     }   //end of outer for loop
```

Continue …

```
for( i=0;i<3; i++ )
     {
        for(j=0;j<3;j++ )
         {
                if( a[i][j]%2== 0 )
                     printf( " even= %d", a[i][j]  );
               else
                     printf( " odd= %d",a[i][j] );
             }
       }        // end of outer for loop
}     // end of main function
```

Strings

What is string?

- Array of characters.
- ‘\o’ in string signifies end of the string in memory.
- Strings is enclosed within double quotes.

Initializing Strings

- 2 ways
- char month1[]={‘j’,’a’,’n’,’u’,’a’,’r’,’y’,’\0’};
- char month1[]=“january”;

Example

```
#include<stdio.h>
  main()
{
   char strc[]="this is a string literal";
   printf("%s",strc);
}
```

o/p:

this is a string literal

Array of strings

```
#include<stdio.h>


main()
{
int j;
char Names[5][9]={“Tejaswi”,”Prasad”,
                  ”Prashant”,”Anand”, “Swati”};
for(j=0;j<5;j++) printf(“%s\n”,Names[j]);
}
```

Names[0]	T	e	j	a	s	w	i		
Names[1]	P	r	a	s	a	d			
Names[2]	P	r	a	s	h	a	n	t	
Names[3]	A	n	a	n	d				
Names[4]	S	w	a	t	i				

- Output:

Tejaswi

Prasad

Prasanth

Anand

Input/output

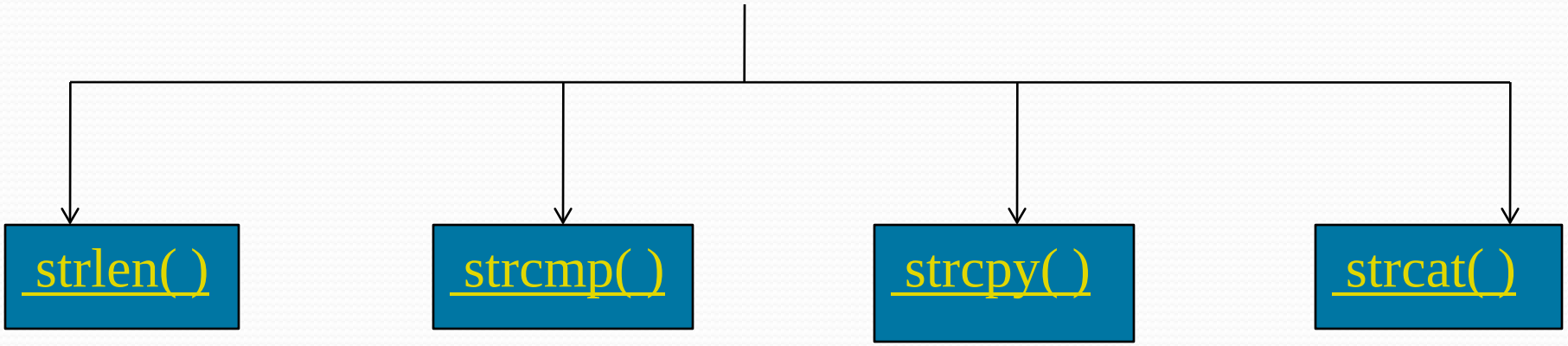
- gets(string) performs input operation .

- puts(string) performs output operation.

Example

```
#include<string.h>
#include<stdio.h>
main()
{  char numstr[20];
    puts("enter the string");
    gets(numstr);
    puts(numstr);
}
```

String functions



Read This:

strncpy()

strncat()

strstr()

```
#include<stdio.h>
main()
{
    char ar[20]=“Arvind”;
    printf(“%d”,strlen(ar));
 }
```

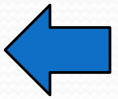
```
#include<stdio.h>
#include<string.h>
void main()
{  int i;
  char name[20]=“home";
  char name1[20]="sweet";
  i=strcmp(name,name1);   // name<name1
  printf("%d”,i);
}
```

o/p: -11

Example

```
#include<stdio.h>
#include<string.h>
void main()
{   char name1[20];
    char name2[20]=“harry”;
    strcpy(name1,name2);
    printf(“%s”,name1);
 }
```

o/p: harry



Example

```
#include<stdio.h>
#include<string.h>
main( )
{int i;
  char name1[20]="home";
  char name2[20]="sweet";
  strcat(name2,name1);
  printf("%s",name2);   }
```

o/p: sweethome