ARRAYS

Situation

- You are required to store the following data for the T1 Marking system:
  - » Name of students along with En. numbers
  - » T1 Marks
  - » Batch

- Compute and display the name of student with highest T1 marks and his/her batch.

Notion of an array

- Array
  - Homogeneous collection of variables of same type.
  - Group of consecutive memory locations.
  - Linear and indexed data structure.
- To refer to an element, specify
  - Array name
  - Position number (Index)

Arrays - Why?

```
main()
{
    float sal1,sal2,sal3;
    scanf (“%f”,&sal1);
    scanf (“%f”,&sal2);
    scanf (“%f”,&sal3);
}
```

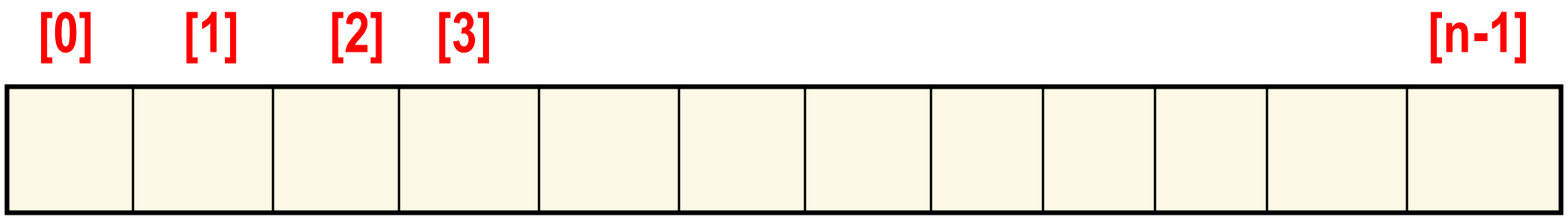
```
main()
{
        int sal[3], i=0;
        while (i < 3)
        {
           scanf (“%f”,&sal[i];
        }
}
```

Imagine having 100 variables……

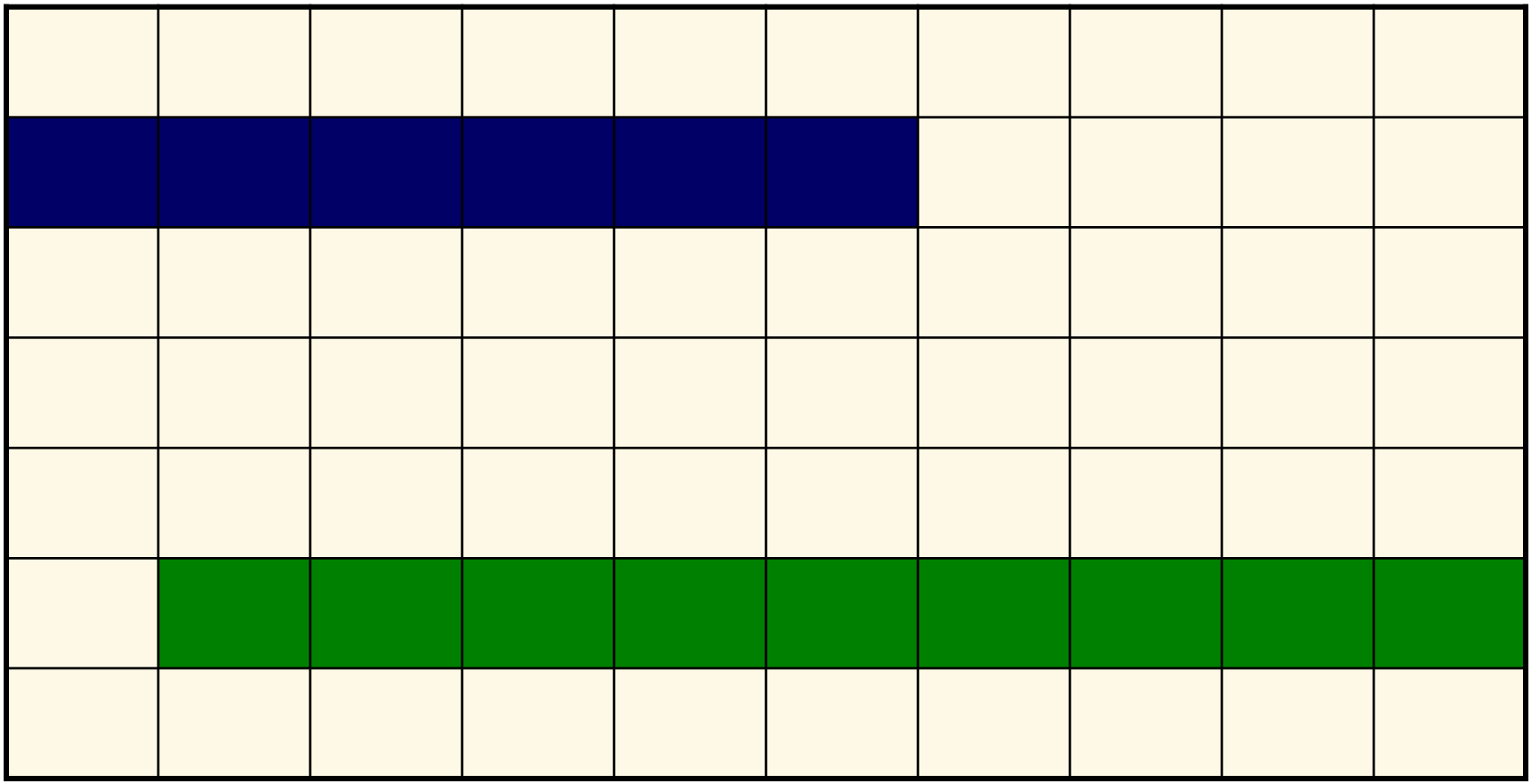
Single Dimension Arrays

Array Format

- Elements of an array can be integers, floats, characters etc.
- All the elements share a common name with an index called subscript.
- In an array of n elements:



Memory Layout



Declaration

- When declaring arrays, specify
  - Data type of array (integers, floats , characters…..)
  - Name of the array.
  - Size: number of elements

```
array_type   array_name[ size ]   ;
```

  - Example:
    - » int student[10] ;
    - » float my_array [300] ;

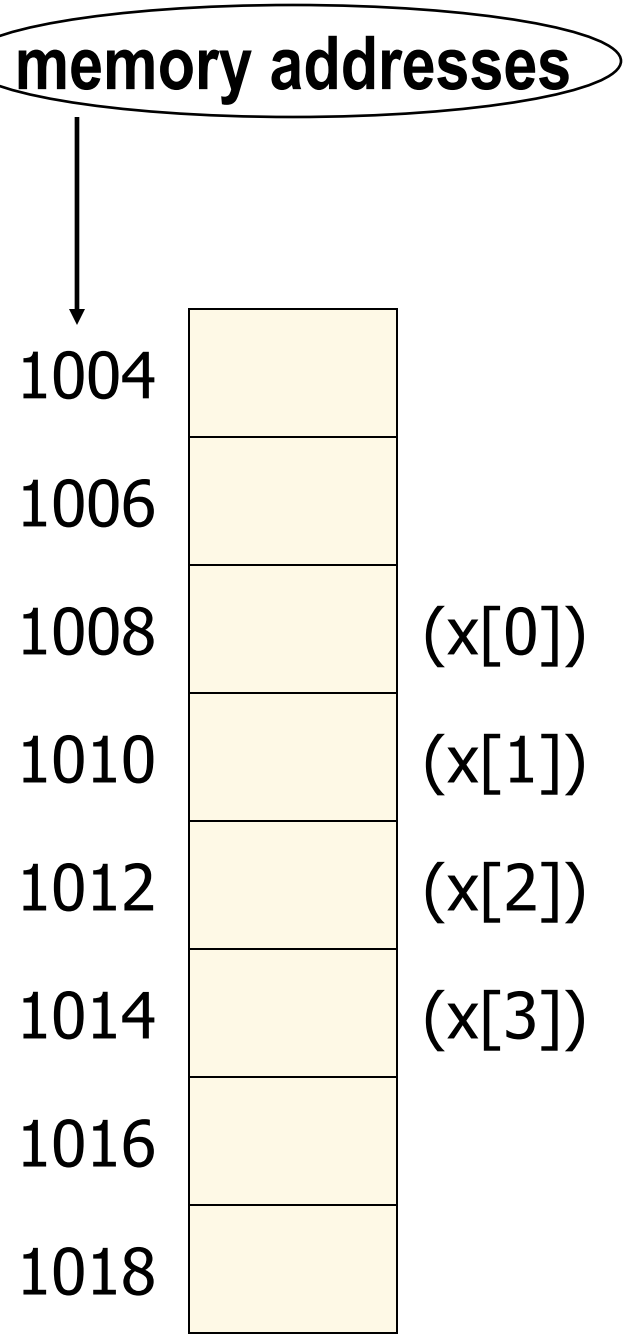
Example

- For example,

  int x [4] ;

  Square bracket

- An array of integers of 4 elements.
- Note that the starting memory address is determined by the operating system (just like that of simple variable).
- Contiguous memory locations are allocated.



Examples

- Integer array of 20 elements:

```
int  array_1  [ 20 ];
```

- Character array of 50 elements:

```
char array_2 [ 50 ];
```

- Float array of 100 elements:

```
float array_3 [ 100 ];
```

sizeof()

- The amount of storage required to hold an array is directly related to its data type and the size.

- Example

  Total size in bytes for a 1 D array:

  total bytes = sizeof(data type) * size of array

  int a [7] ;

  total_bytes = 2 * 7

  = 14 bytes in memory

Exercise

```
int main( void) {
float f1[10] ;
char c1[10] ;
printf("%d", sizeof(f1));
printf("%d", sizeof(c1));
return 0;
}
```

Output:

```
40
10
```

Example 1

Write a program to read 10 integers from the user and display them.

```
int main( void) {
 int  digit[ 10 ] , t  ;
 printf(“ \n Enter the value of 10 integers “ ) ;
 / * for reading 10 integers from the user using scanf   * /
            for( t = 0, t < 10 ; t + + )
                scanf( “ %d”,  &digit [ t ] );
 /* for displaying the integers using printf */
            for( t = 0, t < 10 ; t + + )
                 printf( “ %d”, digit [ t ] );
return 0;}
```

Example 2

- Write a program to read 10 integers and add 5 to odd elements and 10 to even elements.

Solution

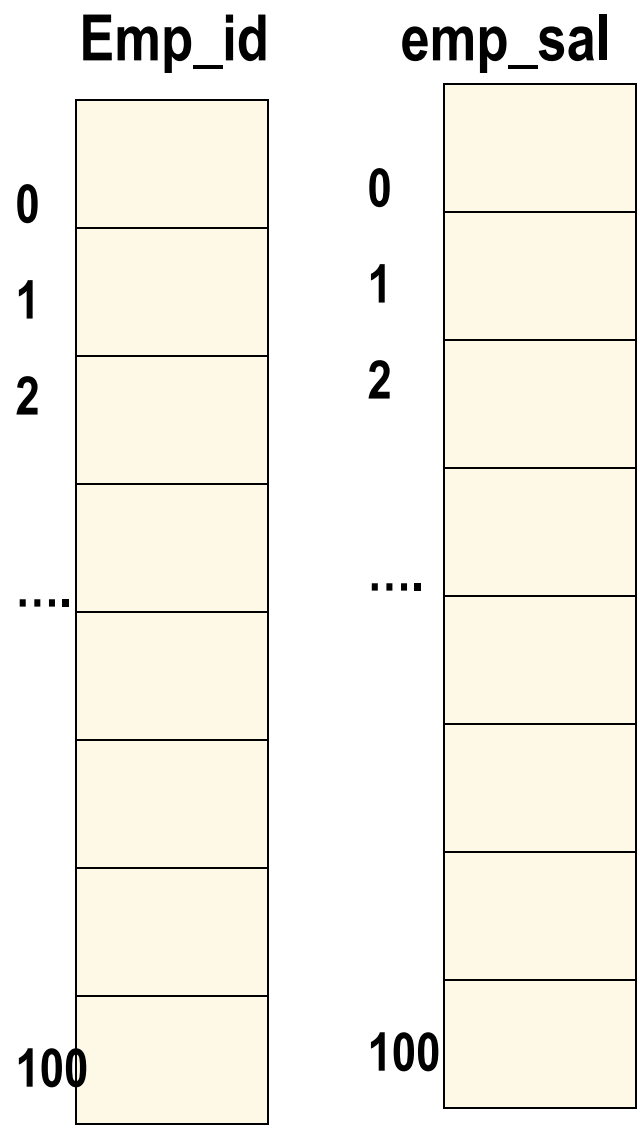
```
#include<stdio.h>
main( ) {
int digit[10 ] , t ;
printf("\n Enter the value of 10 integers" ) ;
for(t=0;t<10;t++)
{ scanf( "%d", &digit[t] );
if(digit[t]%2==0)
digit[t]=digit[t]+10;
else
digit[t]=digit[t]+ 5;
}
for( t=0; t<10; t++ )
printf( " %d", digit[t] );
return 0;
}
```

Exercise 1

- In an organization 100 employee are there. Write a program to store the employee id and salary of all the employees. Then input the employee id, and display the salary of the employee.

Solution

- Steps
  - Declare two arrays int emp_id[100] and float emp_sal[100]
  - Input the value of both the arrays from the user.
  - Input the emp_id for which salary is to be displayed.
  - Search the above id in array emp_id[ ] and get the index if match found
  - In the second array display the salary for the same index.



Initializing Integer arrays

- Another way of initializing the elements of the array is as follows:
  - » int array [ ] = { 2, 20, -30, 10, 100 } ;

•The array size need not be specified explicitly when initial values are included as a part of array declaration.

•Array size will automatically be set equal to the number of initial values specified with in definition.

- int digits [10] = { 3, 30, 300} ;

- All individual array elements that are not assigned explicit initial values will automatically be set to zero.

- digits [0] = 3 digits [1] = 30 digits [2]= 300

  digits [3] = 0 digits [4]= 0 digits[5]= 0

  digits [7] = 0 digits [8] =0 digits [9] =0

Initializing Arrays

The second way is to

Initialize each array element separately

```
id[0]=1234;

id[2]=2883;

id[3]=2322;

id[4]=8888;

id[5]=8237;
```

Basics of character array

- If you are required to store a group of character like your name, city , or your college name, or any word or text you need to define a array of characters.

- A char variable can hold a SINGLE character only like

  char c = ‘A’ ;

  char c1=‘B’;

- What if you need to store “Sachin Tendulkar” or “MUMBAI” a string.

Character Arrays

- To hold a single string you need to declare a single dimension character array
  - » char str [11] ;
- When declaring a character array to hold a string(group of characters), one need to declare the array to be one character longer than the largest string that it will hold
- Example above array str[11] will hold 10 characters and a NULL character ('\0') at the end
- This null character indicates the end of the string.
- Strings are always enclosed by double quotes. Whereas, character is enclosed by single quotes in C.

- Character arrays can be initialized using string literals

```
char string1[] = "first";
```

- Null character '\0' terminates strings
- string1 actually has 6 elements. It is equivalent to

```
char string1[] = { 'f', 'i', 'r', 's', 't', '\0' };
```

- You can access individual characters,

  string1[3] is character ‘s’

How To Enter value ?

```
/* program to read user's name and display it */
int main( )
{
char str [10 ] ;
printf( " Enter your name ");
scanf(" %s ", str);
printf( " Your name is : %s", str);
return 0;
}
```

No need to put &

- In case of character arrays

  Name of the array is the starting address of the array

```
char str[10];
&str [ 0 ]  => str
```

Disadvantage of %s

- While reading a string using %s format specifier, it does not scan the string after the space bar.

- For example

  Sachin Tendulkar

  It will scan only “ Sachin ”.

  It will IGNORE any space or tab space

How to scan a complete text of words?

gets( )

- gets(argument string) : Collects a string of characters terminated by new line character ‘\n’ from the standard input stream (stdin).

- It allows to input spaces, tabs and copies all the character till new line into the argument string and append a NULL character ‘\0’ at the end of it.

- For example

```
char str [ 20 ] ;
gets( str ) ;
```

puts( )

- puts(argument string):

  It displays the string of characters from argument string to standard output till NULL character and appends a new line character at the end.

- For example

  puts (str) ;

EXAMPLE: Input ten numbers into an array, using values of 0 to 99, and print out all numbers except for the largest number..

```
/* to accept 10 values in the range 0 to 99 */
int size=10;
int value[size], i;
for (i=0; i< size; ++i)
  {  scanf (“%d”,&value[i]);
     if (value[i] > 99 || value[i] <0)
        { printf (“ enter only values within 0 -99 ”);
              i--;    }
  }
```

```
/* to find the greatest among the list */
int maximum= 0;
for (i=0; i< size; i++)
 {
     if (maximum < value[i] )
       maximum = value[i];
    }
```

```
/* to print all the values other than the greatest */
for (i=0; i< size; i++)
 {
     if (value[i] != maximum  )
        printf (“%d”, value[i]);
    }
```

C Program to calculate length of a string.

```
#include<stdio.h>
 #include<conio.h>
void main()
 {
char ch[10];
 int i,count=0;
 clrscr();
printf("Enter the string\n");
gets(ch);
 for(i=0;ch[i]!='\0';i++)
{
 count++;
 }
printf("Length = %d",count);
getch();
 }
```

C Program to reverse a string.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int i,count=0;
 char ch[20];
clrscr();
printf("Enter the string\n");
 gets(ch);
for(i=0;ch[i]!='\0';i++)
{
 count++;
}
printf("Reverse is:");
 for(i=count-1;i>=0;i--)
{
printf("%c",ch[i]);
 }
getch(); }
```

C Program to count number of vowels in a string.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch[10]; int i,count=0;
clrscr();
 printf("Enter the string\n");
gets(ch);
for(i=0;ch[i]!='\0';i++)
 {
if(ch[i]=='a' || ch[i]=='e' || ch[i]=='i' || ch[i]=='o'
|| ch[i]=='u')
 {
count++;
}
}
 printf("Vowels = %d",count);
 getch();
 }
```

C Program to count number of spaces in a string.

```
#include<stdio.h>
#include<conio.h>
void main()
{
char ch[10];
 int i,count=0;
 clrscr();
 printf("Enter the string\n");
gets(ch);
 for(i=0;ch[i]!='\0';i++)
{
if(ch[i]==' ')
 count++;
 }
printf("Spaces = %d",count);
 getch();
}
```

WAP in C to reverse an array(if a[0]=1,a[1]=2,a[2]=3 then after reverse operation the array become a[0]=3,a[1]=2,a[2]=1).

Start here | *array.c

```c
int main()
{
  int n, c, d, a[100], b[100];
printf("Enter the number of elements in array\n");
   scanf("%d", &n);
printf("Enter the array elements\n");
for (c = 0; c < n ; c++)
      scanf("%d", &a[c]);
/* Copying elements into array b starting from end of array a*/
for (c = n - 1, d = 0; c >= 0; c--, d++)
      b[d] = a[c];

   /* Copying reversed array into original.
    * Here we are modifying original array, this is optional.
    */
for (c = 0; c < n; c++)
      a[c] = b[c];
printf("Reverse array is\n");
for (c = 0; c < n; c++)
      printf("%d\n", a[c]);
}
```

WAP in C to find the minimum element in an array along with its position.

Start here | array.c

```c
int main()
{
  int array[100], minimum, size, c, location = 1;

    printf("Enter the number of elements in array\n");
    scanf("%d",&size);

    printf("Enter %d integers\n", size);

    for ( c = 0 ; c < size ; c++ )
        scanf("%d", &array[c]);

    minimum = array[0];

    for ( c = 1 ; c < size ; c++ )
    {
        if ( array[c] < minimum )
        {
           minimum = array[c];
           location = c+1;
        }
    }

    printf("Minimum element is present at location %d and it's value is %d.\n", location, minimum);
}
```

Copy one string to another

Start here | string.c

```c
int main()
{
  char s1[100], s2[100];
   int i;

   printf("\nEnter the string :");
   gets(s1);

   i = 0;
   while (s1[i] != '\0') {
      s2[i] = s1[i];
      i++;
   }

   s2[i] = '\0';
   printf("\nCopied String is %s ", s2);
}
```

Concatenation of two strings

Start here | array.c

```c
int main()
{
  char s1[100], s2[100], i, j;
    printf("Enter first string: ");
    scanf("%s",s1);
    printf("Enter second string: ");
    scanf("%s",s2);
    for(i=0; s1[i]!='\0'; ++i);  /* i contains length of string s1. */
    for(j=0; s2[j]!='\0'; ++j, ++i)
    {
        s1[i]=s2[j];
    }
    s1[i]='\0';
    printf("After concatenation: %s",s1);
}
```

Home Assignment

- Write a program to read a text from console and display the number of words and lines in the text.