Computer Concept and Programming

I semester, B.Sc.CSIT

Ambition Guru College

Compiled by :
Ankit Bhattarai

Unit IV

(6 hrs.)

Arrays and Strings

- Introduction,
- Types of Arrays,
- One-Dimensional Array, Multidimensional Arrays,
- String as Array of Characters,
- String Handling Functions

Introduction to Array

- Array is a set of similar data elements which are stored in consecutive memory location under a common variable name.

So to be an array,

- All elements must be of same data type.
- All elements are stored in consecutive memory location.

E.g. If we want to store 100 integers in a sequence, we can create array for it.

```
int num[100];
```

Note: *The size and type of array cannot be changed after its declaration*

Introduction to Array

- If we want many elements of similar type than it is not feasible to declare all variables and also manipulate these elements. So in this case we use array.

For example, if we want 100 integer variables, then instead of writing all 100 variables, we use array like int num[100]. Here 100 integer elements num[0], num[1], num[2]. . . . . .num[99] are declared.

Introduction to Array

Limitation of Array

- Memory allocation in array are static in nature. Which means memory is allocated before the execution of program begins (During compilation).In this type of allocation the memory cannot be resized after initial allocation.

So it has some limitations.

- Wastage of memory
- Overflow of memory

e.g. int num[100];

- Here, the size of an array has been fixed to 100.If we just enter to 10 elements only, then their will be wastage of 90 memory location and if we need to store more than 100 elements there will be memory overflow.

Types of Array

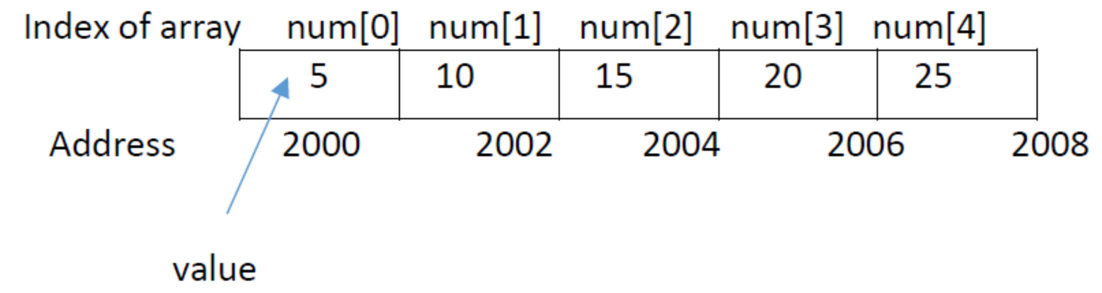
1. One dimensional Array

2. Multidimensional Array

Types of Array

One dimensional array

- Elements of an array can be represented either as a single row or single column.
- There is a single subscript or index whose value refers to the individual array element, which ranges from 0 to n-1, where n is the size of array.
- Eg. int num[5]={5,10,15,20,25};
- Assuming base address 2000, now array elements can be illustrated as:



Index of array	num[0]	num[1]	num[2]	num[3]	num[4]
	5	10	15	20	25

Address 2000 2002 2004 2006 2008

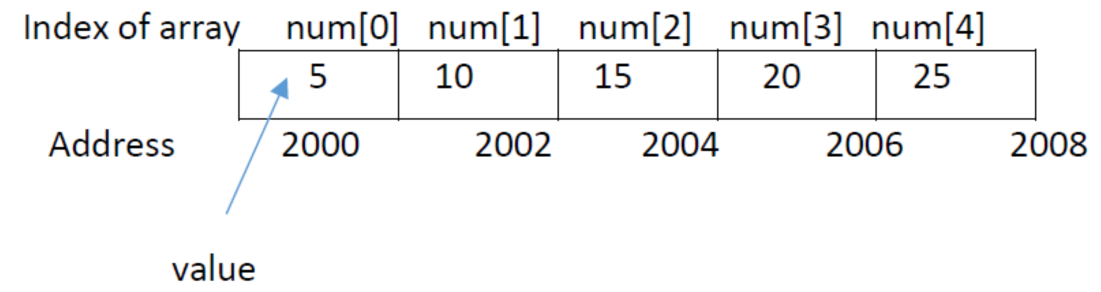
value

Types of Array

Declaration:

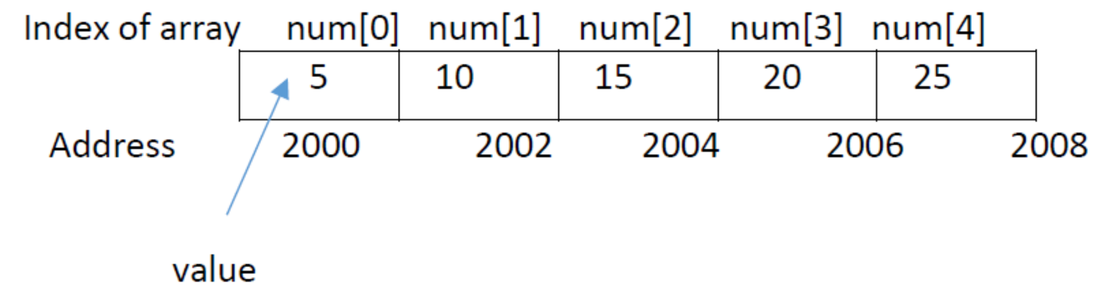
data_type array_name[size];

- eg. int num[10];
- Here, int is a data type and num is the name of the array and 10 is the size of the array. It means num can only contain 10 elements of int type.



Index of array	num[0]	num[1]	num[2]	num[3]	num[4]
	5	10	15	20	25
Address	2000	2002	2004	2006	2008

Types of Array



Index of array	num[0]	num[1]	num[2]	num[3]	num[4]
	5	10	15	20	25
Address	2000	2002	2004	2006	2008

Declaration:

data_type array_name[size];

- eg. int num[10];
- Here, int is a data type and num is the name of the array and 10 is the size of the array. It means num can only contain 10 elements of int type.

Types of Array : Initialization of 1D array

Compile time initialization

General form: `data_type array_name[size]={list of values};`

The value in the list are separated by commas.

- Example:
- `int num[5]={10,20,30,40,50};` //integer array initialization
- `float area[5]={33.4,55.6,88.9,77.8,5.5};` //float array initialization

Note:

It is also possible to initialize array without defining its size.

- `int num[]={67,87,56,24,77};`

Types of Array : Initialization of 1D array

Compile time initialization

Example:

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i;
int num[5]={67,87,56,24,77};
for(i=0;i<5;i++)
{
        printf("\t%d",num[i]
);
}
getch();
return 0;
}
```

Output

```
67 87 56 24 77
```

Types of Array :
Initialization of 1D array

Runtime initialization

- An array can also be initialized at runtime using scanf() function. This Approach is used for initializing large arrays with user specified values.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int num[5];
int i;
printf("\nEnter the array elements");
for(i=0;i<5;i++)
{
            scanf("%d",&num[i]);
}
printf("\nArray elements are");
for(i=0;i<5;i++)
{
            printf("\n%d",num[i]);
}
getch();
return 0;
}
```

Multidimensional Array

Multidimensional Array

- Multidimensional array are those who have two or more than two dimensions.
- Multidimensional array are defined in the same manner as one dimensional arrays, except that separate pair of square brackets is required for each subscript.
- Thus, two dimensional array will require two pair of square brackets, a three dimensional array will require three pairs of square brackets and so on.

General form:

```
        datatype array_name[s1][s2][s3]….[sn]
```

where, si is the size of i^{th} dimension.

Two Dimensional Array (2D array)

A two dimensional array is a collection of similar data items, structured in two dimensions (referred to as rows and columns).

- It is also called array of arrays.
- It is required to manipulate the data in table format or in matrix format which contains rows and columns.

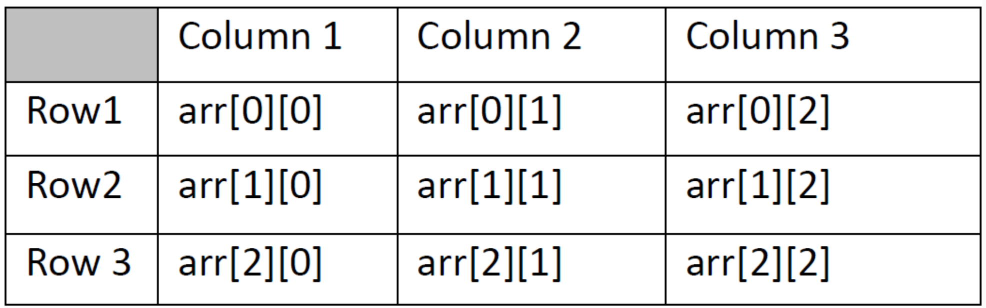
Declaration:

```
        data_type array_name[row_size][column_size];
```

eg. `int arr[3][3];`

Multidimensional Array: Two Dimensional Array (2D array)

It creates two dimensional array to store 9 elements of integer type. There are 3 rows and 3 columns in array matrix.



	Column 1	Column 2	Column 3
Row1	arr[0][0]	arr[0][1]	arr[0][2]
Row2	arr[1][0]	arr[1][1]	arr[1][2]
Row 3	arr[2][0]	arr[2][1]	arr[2][2]

Two Dimensional Array (2D array) : Initialization

Compile time initialization: Two dimensional array may be initialized by following their declaration with a list of values enclosed in braces.

Example: `int arr[3][3]={ {2,4,6},{8,9,12},{15,16,18}};`

These are equivalent to following assignments

```
arr[0][0]= 2            arr[0][1]= 4            arr[0][2]= 6

arr[1][0]= 8             arr[1][1]= 9           arr[1][2]= 12

arr[2][0]=15            arr[2][1]= 16            arr[2][2]=18
```

Types of Array : Initialization of 2D array

Compile time initialization

Example

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,j;
int arr[3][3]={{12,14,16},{5,7,15},{15,25,45}};
printf("The matrix is\n");
for(i=0;i<3;i++)
{
        for(j=0;j<3;j++)
        {
        printf("%d\t",arr[i][j]);
        }
        printf("\n");
}
getch();
return 0;
}
```

Types of Array : Initialization of 2D array

Run time initialization

Program to input 3*3 matrix and display it.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,j;
int arr[3][3];
printf("Enter the elements of matrix\n");
for(i=0;i<3;i++)
        {
        for(j=0;j<3;j++)
        {
        scanf("%d",&arr[i][j]);
        }
}
```

Types of Array :
Initialization of 2D array

Run time initialization

```
printf("Elements of matrix are\n");
for(i=0;i<3;i++)
{
            for(j=0;j<3;j++)
            {
            printf("%d\t",arr[i][j]);
            }
            printf("\n");
}
getch();
return 0;
}
```

Some Questions

WAP to input 10 number in an array and display it.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,num[10];
printf("Enter 10 array elements\n");
for(i=0;i<10;i++)
    {
           scanf("%d",&num[i]);
     }
printf("The array elements are\n");
for(i=0;i<10;i++)
     {
     printf("%d\n",num[i]);
     }
getch();
return 0;
}
```

Some Questions

WAP to input 10 number in an array and display it.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,num[10];
printf("Enter 10 array elements\n");
for(i=0;i<10;i++)
    {
           scanf("%d",&num[i]);
     }
printf("The array elements are\n");
for(i=0;i<10;i++)
     {
     printf("%d\n",num[i]);
     }
getch();
return 0;
}
```

Some Questions

- **WAP to read n elements in array and display them in reverse order**

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,num[100],n;
printf("Enter number of array elements \n");
scanf("%d",&n);
printf("Enter %d array elements\n",n);
for(i=0;i<n;i++)
{
            scanf("%d",&num[i]);
}
printf("Array elements in reverse order are\n");
for(i=n-1;i>=0;i--)
{
            printf("%d\n",num[i]);
}
getch();
return 0;
}
```

Some Questions

- WAP to read n numbers using array and find their average.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int n,i;
float num[100],sum=0,avg;
printf("Enter number of array elements \n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
              scanf("%f",&num[i]);
}
for(i=0;i<n;i++)
{
              sum=sum+num[i];
}
avg=sum/n;
printf("Average=%f\n",avg);
getch();
return 0;
}
```

Some Questions

- WAP to input n numbers in an array and find the sum of all even numbers and count them.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int num[100],i,n,esum=0,ecount=0;
printf("Enter number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
           scanf("%d",&num[i]);
}
for(i=0;i<n;i++)
{
   if(num[i]%2==0)
   {
   esum=esum+num[i];
   ecount++;
   }
}
printf("Sum of Even numbers=%d\n",esum);
printf("Number of Even numbers=%d",ecount);
getch();
return 0;
}
```

String

String

- Strings are sequence of characters stored in consecutive memory location.
- Each character in string occupies one byte of memory.
- Strings always terminated with null character ‘\0’.

Declaration:

```
char string_name[size];
```

Here, size determines the number of characters in string_name.

Eg. `char name[5];`

Note: *When the compiler assigns a character string to character array, it automatically supplies a null character (‘\0’) at the end of string. Therefore, size should be equal to maximum number of characters in string plus one*

String: Initialization

Compile time initialization

General form:

`char string_name[size]=”list of character”;`

eg. `char name[5]=”ram”;`

OR

`char string_name[size]={list of character};`

eg. `char name[5]={‘r’,’a’,’m’,’\0’};`

Note: C also permits to initialize a character array without specifying the number of elements. In such cases size of array will determined automatically, based on number of elements initialized.

Eg. `char name[ ]={‘r’,’a’,’m’,’\0’};`

defines, the array string as a four element array

String

Compile time initialization

Example

```
#include<stdio.h>
#include<conio.h>
int main()
{
char greeting[6]={'H','e','l','l','o','\0'};
char name[4]="ram";
printf("Greeting message:%s",greeting);
printf("\nName: %s",name);
getch();
return 0;
}
```

Output:

```
Greeting message:Hello
Name:ram
```

String: Initialization

Runtime initialization

The familiar input function scanf() can be used with %s format specification to read string.

Example:

```
char name[20];

scanf(“%s”,name);
```

Limitation: Here, string variable takes only single word, It is because when whitespace is encountered the scanf() function terminates

String: Initialization

To overcome this problem the gets() function is used to read a string of text containing whitespaces, until newline character is encountered.

```
#include<stdio.h>
#include<conio.h>
int main()
{
char name[20];
printf("Enter your name:");
gets(name);
printf("Your name is:");
puts(name);
getch();
return 0;
}
```

Output:

```
Enter your name: Hari Bahadur
Your name is: Hari Bahadur
```

String as Array of Characters

String as Array of Characters

- Array of strings means two dimensional array of characters.

Eg. `char name[5][10];`

Here, first dimension tells, how many strings can be stored in array. The second dimension tells the maximum length of each string.

In above declaration, we can store 5 strings, each can store maximum 9 characters. Last 10th space is for null terminator in each string.

Arrays of String

WAP to input name of 5 students and display it.

Example

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i;
char name[5][20];
printf("Enter the name of 5 students\n");
for(i=0;i<5;i++)
{
           gets(name[i]);
}
printf("The names are\n");
for(i=0;i<5;i++)
{
           puts(name[i]);
}
getch();
return 0;
}
```

String Handling Functions

String Handling Functions

- C provides several built-in functions for string manipulation, which are declared in the <string.h> header file. These functions can be used for tasks such as copying, concatenation, comparison, and finding the length of strings.

Some of them are as follows:

1. **strlen()** : Finds the length of a string excluding null character
2. **strcpy():** Copies one string to another including null character
3. strrev(): Reverses all characters in string except null character
4. strcat(): Concatenates two strings. i.e. it appends one string at the end of other
5. strlwr() converts the uppercase string into lowercase
6. strupr() converts the lowercase string into uppercase

String Handling Functions

7. **strcmp():** Compares two strings to find out whether they are same or different. This functions accepts two strings as parameters and returns an integer whose value is

i. Less than 0 if the first string is less than second

ii. Equal to 0 if both are same

iii. Greater than 0 if first string is greater than second

String handling function

Example

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main ()
{
int l ;
char str1[20] = “rajarshi";
char str2[20] = "university";
char str3[20];
l=strlen(str1);
printf("Length of the str1 is %d\n",l);
strcpy(str3,str1);
printf("string copied from str1 to str3 is %s\n",str3 );
strcat(str1,str2);
printf("string after concatenation(s1+s2)is %s\n",str1);
strrev(str3);
printf("Reverse of string str3 is %s\n",str3);
strupr(str3);
printf("Uppercase of string str3 is %s\n",str3);
getch();
return 0;
}
```

String handing functions : strlen()

- This function returns the an integer which denotes the length of a string passed. Length of a string is number of characters present in it, excluding the terminating null character.

Its syntax is :

```
        integer_variable =strlen(string);
```

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str[20];
int len;
printf("Enter the string:");
gets(str);
len=strlen(str);
printf("The length of a string is
%d",len);
getch();
return 0;
}
```

Output:

```
Enter the string : ramesh
The length of string is 6
```

String handing functions : strcpy()

- This function copies one string to another.
- The function accepts two strings as parameters and copies the second string character by character into first one upto including the null character of the second string.

The syntax is :

```
strcpy(destination_string,source_string);
```

i.e. strcpy(str2,str1) means the content of str1 is copied to str2

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

char str1[20]="ramesh",str2[20];

strcpy(str2,str1);

printf("The copied string is

%s",str2);

getch();

return 0;

}
```

Output:
The copied string is ramesh

String handing functions : strcat()

- This function concatenates two strings i.e it appends one string at the end of another.
- This function accepts two strings as parameters and stores the contents of second string at the end of first. Its syntax is :

```
strcat (string1,string2);
```

ie.string1=string1+string2

```
#include<stdio.h>

#include<conio.h>

#include<string.h>

int main()

{

char

str1[20]=“rajarshi",str2[20]="university";

strcat(str1,str2);

printf("The concatenated string is %s",str1);

getch();

return 0;

}
```

Output:
The concatenated string is rajarshiuniversity

String handing functions : strupr()

- This function converts the lowercase string into uppercase.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str[20]="hello";
strupr(str);
printf("The uppercase of given string is %s",str);
getch();
return 0;
}
```

Output:

```
The uppercase of given string is HELLO
```

String handing functions : strlwr()

- This function converts the uppercase string into lowercase.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str[20]="HELLO";
strlwr(str);
printf("The lowercase of given string is %s",str);
getch();
return 0;
}
```

Output:

```
The lowercase of given string is hello
```

String handing functions : strrev()

This function is used to reverse all characters in a string except null character at the end of string. The reverse of string “abc” is “cba”.
It’s syntax is strrev(string);

For example:
strrev(s) means it reverses the characters in string s and stores reversed string in s.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str[20]="hello";
strrev(str);
printf("Reversed string is %s\n",str);
getch();
return 0;
}
```

Output:

```
Reversed string is olleh
```

String handing functions : strcmp()

This function accepts two strings as parameters and returns an integer whose value is

i)less than 0 if the first string is less than second

ii)equal to 0 if both are same

iii)greater than 0 if first string is greater than second

The two strings are compared character by character until there is a mismatch or end of one string is reached.

Whenever two characters in two string differ, the string which has the character with higher ASCII value is greater.

For example, consider two strings “ram” and “rajesh”.The first two characters are same but the third character in string ram and that is in rajesh are different. Since ASCII value of character m in string is ram is greater than that of j in string rajesh, the string ram is greater than rajesh.

String handing functions : strcmp()

```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str1[20],str2[20];
int diff;
printf("Enter the first string:");
gets(str1);
printf("Enter the second string:");
gets(str2);
diff=strcmp(str1,str2);
if(diff>0)
{
    printf("%s is greater than %s",str1,str2);
}
else if(diff<0)
{
    printf("%s is greater than %s ",str2,str1);
}
else
{
     printf("Both strings are same");
}
getch();
return 0;
}
```

Output:

```
Enter first string: ram
Enter second string: rajesh
ram is greater than rajesh
```

Some Questions to Work On

- WAP to read n numbers in an array find the largest number among them.
- WAP to input n number in an array and sort them in Ascending order
- Write a program to add two 3X3 matrix. Display the sum stored in third matrix
- WAP to check whether the given number is present in a array or not.
- Write a program to enter values in 3*3 order matrix and compute the sum of odd elements.

Some Questions to Work On

WAP to store Fibonacci series of 10 terms in array and display them.

```c
#include<stdio.h>
#include<conio.h>

int main()
{
int fib[10],i;
fib[0]=0;
fib[1]=1;
printf("The fibonacci series is \n");
printf("%d\t%d\t",fib[0],fib[1]);
for(i=2;i<=9;i++)
{
  fib[i]=fib[i-1]+fib[i-2];
}

for(i=2;i<=9;i++)
{
  printf("%d\t",fib[i]);
}

getch();
return 0;
}
```

Some Questions to Work On

WAP to read an array of length n. Display the total count for positive numbers, negative numbers and zeros separately.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int i,num[100],n,pcount=0,ncount=0,zcount=0;
printf("Enter the number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
     scanf("%d",&num[i]);
}
for(i=0;i<n;i++)
{
     if(num[i]>0)
     {
        pcount++;
     }
     else if(num[i]<0)
     {
        ncount++;
     }
     else
     {
        zcount++;
     }
}
printf("Number of postive numbers=%d",pcount);
printf("\nNumber of Negative numbers=%d",ncount);
printf("\nNumber of Zeros=%d",zcount);
getch();
return 0;
}
```

Some Questions to Work On

WAP to store n numbers in an array and display prime numbers stored in array and calculate the sum of those prime numbers.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int a[100],flag,n,i,j,sum=0;
printf("Enter the no. of elements:");
scanf("%d",&n);
printf("Enter %d array elements\n",n);
for(i=0;i<n;i++)
{
     scanf("%d",&a[i]);
}
printf("\nprime numbers are:\n");
for(i=0;i<n;i++)
{
   flag=0;
   for(j=2;j<a[i];j++)
   {
      if(a[i]%j==0)
      {
      flag=1;
      break;
      }
   }

if(flag==0)
{
   printf("%d\t",a[i]);
   sum=sum+a[i]
}
}
printf("\nThe sum of prime no. is %d",sum);
getch();
}
```

Some Questions to Work On

WAP to check whether the given number is present in a array or not.

```c
#include<stdio.h>
#include<conio.h>
int main()
{
int a[100],i,n,num,flag=0;
printf("Enter the number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
printf("\nEnter the element that you want to search \n");
scanf("%d",&num);
for(i=0;i<n;i++)
{
     if(num==a[i])
     {
        flag=1;
        break;
     }
}
if(flag==1)
{
     printf("your number is found");
}
else
{
     printf("your number is not found");
}
getch();
return 0;
}
```

Some Questions to Work On

Write a program to enter values in 3*3 order matrix and compute the sum of odd elements.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int arr[3][3],i,j,osum=0;
printf("Enter the elements of matrix\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
       scanf("%d",&arr[i][j]);
    }
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(arr[i][j]%2!=0)
        {
           osum=osum+arr[i][j];
        }
    }
}
printf("Sum of odd elements=%d",osum);
getch();
return 0;
}
```

Some Questions to Work On

WAP to input m*n matrix and find sum of each row.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int arr[20][20],i,j,m,n,sum;
printf("Enter the order of matrix\n");
scanf("%d%d",&m,&n);
printf("Enter %d elements of
matrix\n",m*n);
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
       scanf("%d",&arr[i][j]);
    }
}
for(i=0;i<m;i++)
{
    sum=0;
    for(j=0;j<n;j++)
    {
       sum=sum+arr[i][j];
    }
    printf("sum of %d row is %d\n",i+1,sum);
}
getch();
return 0;
}
```

Some Questions to Work On

WAP to read two 3*3 matrix and multiply them.

```
#include<stdio.h>
#include<conio.h>
int main()
{
int
a[3][3],b[3][3],mul[3][3],i,j,k;
printf("Enter 9 elements of first
matrix\n");
for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
      {
        scanf("%d",&a[i][j]);
      }
}

printf("Enter 9 elements of
second matrix\n");
for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
      {
          scanf("%d",&b[i][j]);
      }
}
```

```
for(i=0;i<3;i++)
{
      for(j=0;j<3;j++)
      {
          mul[i][j]=0;
          for(k=0;k<3;k++)
          {

mul[i][j]=mul[i][j]+a[i][k]*b[k][j];
          }
}
}

printf("Muliplication of matrix is
:\n");
for(i=0;i<3;i++)
{
        for(j=0;j<3;j++)
        {
           printf("%d\t",mul[i][j]);
         }
printf("\n");
}
getch();
return 0;
}
```

Some Questions to Work On

WAP to concatenate two strings inputted through keyboard. The result of concatenation should be displayed after copied on third variable.

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
char str1[20],str2[20],str3[40];
printf("Enter the first string\n");
gets(str1);
printf("Enter the second string\n");
gets(str2);
strcpy(str3,strcat(str1,str2));
puts(str3);
getch();
return 0;
}
```

Some Questions to Work On

Write a program to check whether the given string is palindrome or not. (Palindrome is a word which reads same from left to right and right to left.eg LIRIL,MADAM etc.)

```
#include<stdio.h>
#include<conio.h>
#include<string.h>

int main()
{
char str1[20],str2[20];
printf("Enter the string\n");
gets(str1);
strcpy(str2,str1);
strrev(str2);
if(strcmp(str1,str2)==0)
{
   printf("String is palindrome");
}
else
{
   printf("String is not palindrome");
}
getch();
return 0;
}
```

Some Questions to Work On

WAP to sort n students name in alphabetical order

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
int main()
{
int i,j,n;
char name[100][20],temp[20];
printf("Enter the number of students\n");
scanf("%d",&n);
printf("Enter the name of %d students:\n",n);
for(i=0;i<n;i++)
{
    gets(name[i]);
}
for(i=0;i<n-1;i++)
{
    for(j=0;j<n-1-i;j++)
    {
       if(strcmp(name[j],name[j+1])>0)
       {
         strcpy(temp,name[j]);
         strcpy(name[j],name[j+1]);
         strcpy(name[j+1],temp);
       }
     }
}
printf("Name of students in alphabetical order are\n");
for(i=0;i<n;i++)
{
     puts(name[i]);
}
getch();
return 0;
}
```

THANK YOU

Any Queries ?