



**GLA UNIVERSITY**

## **PRACTICAL FILE**

**SUBMITTED TO:** GURPREET KAUR

**SUBMITTED BY:** ANSH GARG

**SECTION :** G1

**UNIVERSITY ROLL NO.:** 2215000281

**CLASS ROLL NO. :** 09

**BRANCH :** BTECH CSE

**SUBJECT :** C PROGRAMMING LAB

**DEPARTMENT NAME :** COMPUTER SCIENCE

**& ENGINEERING**

## **LOOP QUESTIONS**

1. Write a C program to convert given decimal number to binary number.

```
#include<stdio.h>
int main(){
    int rem,rev,var,no,binary=0,i=1;
    printf("Enter the decimal number\n");
    scanf("%d",&no);
    var=no;
    while(no!=0)
    {
        rem=no%2;
        no=no/2;
        binary=binary+(rem*i);
        i=i*10;
    }
    printf("binary of %d is %d",var,binary);
    return 0;
}
```

### **OUTPUT**

```
Enter the decimal number :15
binary of 15 is 1111
```

2. Write a C program to find average of all odd number in the given range.

```
#include<stdio.h>
int main(){
    int i,start,end,count=0,sum=0;
    float avg;
    printf("Enter range from start to end\n");
    scanf("%d%d",&start,&end);
    for(i=start;i<=end;i++)
    {
        if(i%2!=0){
```

```

        sum=sum+i;
        count++;
    }
}
avg=(float)sum/count;
printf("sum is %d and avg is %f",sum,avg);
return 0;
}

```

### OUTPUT

```

Enter range from start to end : 2,25
sum is 1368 and avg is 38.000000

```

3. Write a c program to print n terms of Fibonacci series.
- 

```

#include<stdio.h>
int main(){
int i,n,c,a=0,b=1;
printf("Enter the terms of series\n");
scanf("%d",&n);
printf("%d terms of fabonacci series is : ",n);
for(i=1;i<=n;i++)
{
printf("%d ",a);
c=a+b;
a=b;
b=c;
}
return 0;
}

```

### OUTPUT

```

Enter the terms of series : 5
5 terms of fabonacci series is : 0 1 1 2 3

```

4. The program will read two integer numbers and find the multiplication of them using arithmetic plus (+) operator. Do not use multiplication operator to multiply the numbers.
- 

```
#include<stdio.h>
int main(){
    int mul=0,i,a,b;
    printf("Enter first and second number\n");
    scanf("%d%d",&a,&b);
    for(i=1;i<=b;i++){
        mul=mul+a;
    }
    printf("multiplication of %d and %d is %d",a,b,mul);
    return 0;
}
```

### OUTPUT

```
Enter first and second number : 5  6
multiplication of 5 and 6 is 30
```

5. Write a program to read an age of 15 person & find out how many of them fall under :
- a. Still a baby- age 0 to 5
  - b. Attending school - age 6 to 17
  - c. Adult life- age 18 & over
- 

```
#include<stdio.h>
int main(){
    int i,age,a=0,b=0,c=0;
    printf("Enter age of 15 persons\n");
    for(i=0;i<15;i++)
    {  scanf("%d",&age);
       if(age>=0&&age<=5)
         a++;
       else if(age>=6&&age<=17)
         b++;
       else
         c++;
    }
    printf("Number of babies = %d\n",a);
    printf("Number of school going children = %d\n",b);
    printf("Number of adults = %d\n",c);
}
```

```

else if(age>=6&&age<=17)
b++;
else
c++; }
printf("%d is baby age\n",a);
printf("%d is attending school\n",b);
printf("%d is adult life\n",c);
return 0; }
```

## OUTPUT

```

Enter age of 15 persons
Age of 1 is :2
Age of 2 is :3
Age of 3 is :36
Age of 4 is :6
Age of 5 is :12
Age of 6 is :10
Age of 7 is :15
Age of 8 is :20
Age of 9 is :16
Age of 10 is :14
Age of 11 is :18
Age of 12 is :17
Age of 13 is :7
Age of 14 is :9
Age of 15 is :19
2 is baby age
9 is attending school
4 is adult life
```

6. Write a program to find all leap year in the given range of years.

---

```
#include<stdio.h>
int main(){
    int year,start,end;
    printf("Enter range from start to end\n");
```

```

scanf("%d%d",&start,&end);
printf("leap year between %d and %d is:\n ",start,end);
for(year=start;year<=end;year++)
{ if(year%400==0||year%4==0){
    printf("%d\t",year); } }
return 0;
}

```

## OUTPUT

```

Enter range from start to end :2002 2022
leap year between 2002 and 2022 is:
2004          2008          2012          2016          2020

```

- 
7. Write a program to check whether a given number is prime or not.
- 

```

#include<stdio.h>
int main() {
    int i,n,c=0;
    printf("Enter the number that you want to check\n");
    scanf("%d",&n);
    for(i=1;i<n;i++){
        if(i%2==0){
            c++; } }
        if(i==2)
        printf("Number is prime");
        else
        printf("Number is not prime");
        return 0;
}

```

## OUTPUT

```

Enter the number that you want to check :23
Number is not prime

```

8. Write a program to find number is palindrome or not.

---

```
#include<stdio.h>
int main(){
    int n,i,rem,original,rev=0;
    printf("Enter the number that you want to check\n");
    scanf("%d",&n);
    original=n;
    while(n>0){
        rem=n%10;
        rev=rev*10+rem;
        n=n/10; }
        if(original==n){
            printf("palindrome");}
        else{
            printf("Not palindrome");}
    return 0;
}
```

### OUTPUT

```
Enter the number that you want to check
123
Not palindrome
```

9. Write a program to print all odd numbers from m to n.

---

```
#include<stdio.h>
int main(){
    int i,j,m,n;
    printf("Enter limit form m to n\n");
    scanf("%d%d",&m,&n);
```

```
for(i=m;i<=n;i++)
{ if(i%2==1){
    printf("%d ",i); }
return o; }
```

### OUTPUT

```
Enter limit form m to n
1
10
1 3 5 7 9
```

10. Consider a scenario where user enters numbers continuously and we are supposed to find the sum of all those numbers entered by the user and as the user enters a negative number we must stop him from entering numbers further and print the sum.

Write a C program to accomplish the task  
mentioned above.

---

```
#include<stdio.h>
int main(){
int sum=0,num;
printf("Enter positive numbers for sum\n");
while(1)
{ scanf("%d",&num);
if(num<0)
{ break; }
sum=sum+num;}
```

```
printf("sum is %d",sum);
return 0; }
```

## OUTPUT

```
Enter positive numbers for sum
4
5
8
9
14
2
-2
sum is 42
```

## **1-D ARRAY**

1. WAP to input an array of N number of elements and display it.
- 

```
#include<stdio.h>
int main() {
int i,n;
printf("Enter the number of elements \n");
scanf("%d",&n);
int a[n];
for(i=0;i<n;i++) {
printf("enter element %d:",i+1);
scanf("%d",&a[i]); }
printf("Entered elements are : ");
for(i=0;i<n;i++) {
printf("%d ",a[i]);}
return 0;
}
```

### **OUTPUT**

```
Enter the number of elements
4
enter element 1:12
enter element 2:2
enter element 3:34
enter element 4:24
Entered elements are : 12 2 34 24
```

2. WAP to input an array of N number of elements and display it in reverse order.
- 

```
#include<stdio.h>
int main() {
int i,j,k,c,n;
printf("Enter the value of n\n");
scanf("%d",&n);
```

```

int a[n];
for(i=0;i<n;i++) {
printf("enter the value of element %d:",i+1);
scanf("%d",&a[i]); }
printf("array before reverse\n");
for(i=0;i<n;i++) {
printf("%d\t",a[i]); }
k=n;
while(k>=0){
for(j=0;j<k-1;j++) {
c=a[j];
a[j]=a[j+1];
a[j+1]=c; }
k--; }
printf("\narray afre reverse\n");
for(i=0;i<n;i++) {
printf("%d\t",a[i]); }
return 0; }

```

## OUTPUT

```

Enter the value of n
4
enter the value of element 1:1
enter the value of element 2:2
enter the value of element 3:3
enter the value of element 4:4
array before reverse
1      2      3      4
array afre reverse
4      3      2      1

```

3. WAP to input an array of N number of elements and find the sum and average of all the elements of that array.
- 

```

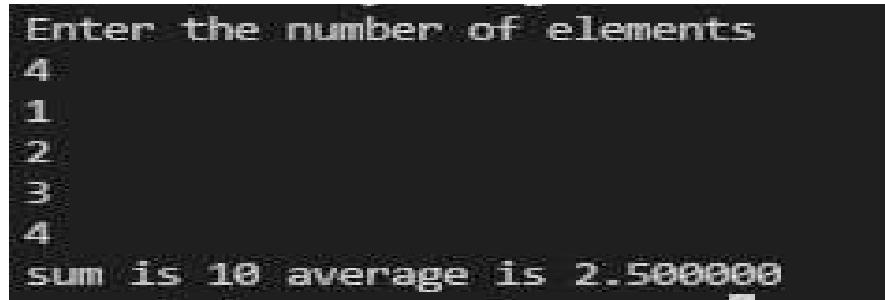
#include<stdio.h>
int main(){
    int i,n,sum=0;
    float avg;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    int a[n];
    for(i=0;i<n;i++){

```

```

        scanf("%d",&a[i]); }
for(i=0;i<n;i++){
    sum+=a[i];
    avg=(float)sum/n; }
printf("sum is %d ",sum);
printf("average is %f ",avg);
return 0; }
```

## OUTPUT



```

Enter the number of elements
4
1
2
3
4
sum is 10 average is 2.500000
```

- 4.WAP to input an array of N number of elements and count total number of positives, negatives and zero elements in that array and display those counts.
- 

```

#include<stdio.h>
int main(){
    int a[100];
    int i,l=0,p=0,n,z=0;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    for(i=0;i<n;i++) {
        printf("Enter elements %d: ",i+1);
        scanf("%d",&a[i]); }
    for(i=0;i<n;i++) {
        if(a[i]<0)
        { l++; }
        else if (a[i]>0)
        { p++; }
        else if (a[i]==0)
        { z++; } }
    printf("total number of postive elemenst is %d\n",p);
    printf("total number of negative elemenst is %d\n",l);
    printf("total number of zero elemenst is %d\n",z);
```

---

```
    return o; }
```

## OUTPUT

```
Enter the number of elements
4
Enter elements 1: 1
Enter elements 2: 0
Enter elements 3: 0
Enter elements 4: 2
total number of positive elemenst is 2
total number of negative elemenst is 0
total number of zero elemenst is 2
```

5. WAP to input an array of N number of elements and store all even numbers in 1 array and all odd numbers in another array. Print both the even and odd array separately.
- 

```
#include<stdio.h>

int main(){
    int a[100],b[100],c[100];
    int i,j=0,k=0,n;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    { printf("Enter elements %d: ",i+1);
        scanf("%d",&a[i]);
    }
    for(i=0;i<n;i++) {
        if(a[i]%2==0){
            b[j]=a[i];
            j++;
        }
        else{ c[k]=a[i];
            k++;
        }
    }
    printf("\nEven array: ");
```

```

for(i=0;i<j;i++)
{ printf("%d\t",b[i]); }
printf("\nodd array: ");
for(i=0;i<k;i++)
{ printf("%d\t",c[i]); }
return o; }

```

## OUTPUT

```

Enter the number of elements
6
Enter elements 1: 1
Enter elements 2: 2
Enter elements 3: 3
Enter elements 4: 4
Enter elements 5: 5
Enter elements 6: 6

Even array: 2      4      6
odd array: 1      3      5

```

- 
6. WAP to input an array of N number of elements and find the largest element in that array.
- 

```

#include<stdio.h>
int main(){
    int a[100];
    int i,j,n,lar,smal;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    { printf("Enter the value of elements %d: ",i+1);
        scanf("%d",&a[i]); }
    lar=a[0];
    smal=a[0];
    for(i=0;i<n;i++)
    { if(a[i]>lar) {
        lar=a[i];
    } }
}

```

```
printf("largest element is %d",lar);
return 0;}
```

## OUTPUT

```
Enter the number of elements
5
Enter the value of elements 1: 21
Enter the value of elements 2: 25
Enter the value of elements 3: 12
Enter the value of elements 4: 15
Enter the value of elements 5: 45
largest element is 45
```

7.WAP to input an array of N number of elements and find  
the smallest element in that array.

---

```
#include<stdio.h>

int main(){
    int a[100];
    int i,j,n,lar,smal;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    { printf("Enter the value of elements %d: ",i+1);
        scanf("%d",&a[i]); }
    lar=a[0];
    smal=a[0];
    for(i=0;i<n;i++)
    { if(a[i]<smal)
        { smal=a[i]; } }
    printf("smallest element is %d",smal);
    return 0;}
```

## OUTPUT

```
Enter the number of elements
5
Enter the value of elements 1: 1
Enter the value of elements 2: 2
Enter the value of elements 3: 12
Enter the value of elements 4: 54
Enter the value of elements 5: 5
smallest element is 1
```

8. WAP to input an array of N number of distinct elements.  
Input an element you want to search and find it. If found  
then print the position of that element otherwise print not  
found.
- 

```
#include<stdio.h>
int main(){
    int a[100],n,count=0,element,i;
    printf("Enter the number of elements\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the value of element %d : ",i+1);
        scanf("%d",&a[i]);
    }
    printf("Enter the element that you want to find : ");
    scanf("%d",&element);
    for(i=0;i<=n;i++)
    {
        if(a[i]==element)
            printf("Element found at position %d ",i+1);
        else
            count++;
    }
    if(count==1)
```

```
{ printf("element not found");}  
return o;}
```

## OUTPUT

```
Enter the number of elements  
4  
Enter the value of element 1 :1  
Enter the value of element 2 :2  
Enter the value of element 3 :8  
Enter the value of element 4 :7  
Enter the element that you want to find :7  
Element found at position 4
```

9. WAP to input an array of N number of elements and find the frequency of an inputted element in that array.
- 

```
#include<stdio.h>  
int main(){  
int i,j,n,e,c=0;  
int a[100];  
printf("Enter the length of the array\n");  
scanf("%d",&n);  
for(i=0;i<n;i++){  
printf("enter element :");  
scanf("%d",&a[i]); }  
printf("Enter the element that you want to find\n");  
scanf("%d",&e);  
for(i=0;i<n;i++)  
{ if(a[i]==e){  
c++;}}  
printf("frequency of %d is %d",e,c);  
return 0;}
```

## OUTPUT

```
Enter the length of the array
4
enter element :1
enter element :1
enter element :2
enter element :3
Enter the element that you want to find
1
frequency of 1 is 2
```

10. WAP to input an array of N number of elements. Input an element you want to insert in that array along with the position and insert it. Print the final array after insertion.

---

```
#include<stdio.h>

int main(){
    int i,j;element,index,position,num;
    printf("Enter the length of the array \n");
    scanf("%d",&num);
    int a[num],b[num];
    for(i=0;i<num;i++) {
        printf("Enter elements %d:",i+1);
        scanf("%d",&a[i]); }
    printf("array before insertion\n");
    for(i=0;i<num;i++) {
        printf(" %d ",a[i]); }
    printf("\nEnter the element and postion of num\n");
    scanf("%d%d",&element,&position);
    index=position-1;
    num++;
    for(i=num-1;i>index;i--) {
```

```
a[i]=a[i-1]; }  
a[i]=element;  
printf("array after insertion\n");  
for(i=0;i<num;i++) {  
printf("%d ",a[i]); }  
return 0; }
```

## OUTPUT

```
Enter the length of the array  
5  
Enter elements 1:2  
Enter elements 2:3  
Enter elements 3:4  
Enter elements 4:5  
Enter elements 5:8  
array before insertion  
2 3 4 5 8  
Enter the element and position of num  
0  
2  
array after insertion  
2 0 3 4 5 8
```

## **2-D ARRAY**

- 1.** WAP to input a 2D array of size M\*N and display the transpose of it.
- 

```
#include<stdio.h>

int main() {
    int i,j,row,column,count=0;
    int a[10][10],b[10][10];
    printf("Enter the number of rows and columns\n");
    scanf("%d%d",&row,&column);
    for(i=0;i<row;i++) {
        for(j=0;j<column;j++) {
            printf("Enter element [%d] [%d]",i,j);
            scanf("%d",&a[i][j]); } }
    printf(" matrix before transpose is :\n ");
    for(i=0;i<row;i++) {
        for(j=0;j<column;j++) {
            printf("%d\t",a[i][j]); }
        printf("\n"); }
    for(i=0;i<column;i++) {
        for(j=0;j<row;j++) {
            b[i][j]=a[j][i]; } }
    printf("Transpose of a matrix is : \n");
    for(i=0;i<column;i++) {
        for(j=0;j<row;j++) {
```

```

printf("%d\t",a[j][i]); }

printf("\n");
return 0;
}

```

## OUTPUT

```

Enter the number of rows and columns
2
2
Enter element [0] [0]5
Enter element [0] [1]10
Enter element [1] [0]15
Enter element [1] [1]20
matrix before transpose is :
5      10
15     20
Transpose of a matrix is :
5      15
10     20

```

- 2.** WAP to input a 2D array of size M\*N and display it in tabular form.
- 

```

#include<stdio.h>

int main(){

    int i,j,n,rows,columns;

    int a[20][20];

    printf("Enter the value of rows and columns\n");

    scanf("%d%d",&rows,&columns);

    for(i=0;i<rows;i++){

        for(j=0;j<columns;j++){

            printf("Enter element [%d][%d]",i,j);

            scanf("%d",&a[i][j]);}

    }

    printf("Matrix in tabular form :\n");

    for(i=0;i<rows;i++){


```

```

for(j=0;j<columns;j++){
    printf("%d\t",a[i][j]);
}
printf("\n");
return 0;
}

```

## OUTPUT

```

Enter the value of rows and columns
3
3
Enter element [0][0]1
Enter element [0][1]2
Enter element [0][2]3
Enter element [1][0]4
Enter element [1][1]5
Enter element [1][2]6
Enter element [2][0]7
Enter element [2][1]8
Enter element [2][2]9
Matrix in tabular form :
1      2      3
4      5      6
7      8      9

```

- 
3. WAP to input a 2D array of size M\*N and display boundary elements in matrix form.
- 

```

#include<stdio.h>

int main(){
    int i,j,row,column;
    printf("Enter the number of row and column\n");
    scanf("%d%d",&row,&column);
    int a[row][column];
    for(i=0;i<row;i++) {
        for(j=0;j<column;j++){
            printf("Enter element [%d][%d]",i,j);
            scanf("%d",&a[i][j]); }
        printf("boundary elements of matrix :\n");
}

```

---

```

for(i=0;i<row;i++) {
    for(j=0;j<column;j++){
        if(i==0||j==0||i==row-1||j==row-1){
            printf("%d",a[i][j]);
        }
        else {
            printf(" ");
        }
        printf("\n");
    }
    return 0;
}

```

## OUTPUT

```

Enter the number of row and column
4
4
Enter element [0][0]1
Enter element [0][1]2
Enter element [0][2]3
Enter element [0][3]4
Enter element [1][0]1
Enter element [1][1]2
Enter element [1][2]3
Enter element [1][3]4
Enter element [2][0]1
Enter element [2][1]2
Enter element [2][2]3
Enter element [2][3]4
Enter element [3][0]1
Enter element [3][1]2
Enter element [3][2]3
Enter element [3][3]4
boundary elements of matrix :
1234
1 4
1 4

```

- 4.** WAP to input a matrix and check if its identity matrix or not.

---

```

#include<stdio.h>

int main(){
    int i,j,row,column,c=1;
    printf("Enter the number of row and columns\n");

```

```

scanf("%d%d",&row,&column);

int a[5][5],b[5][5];

printf("Enter the elements of matrix\n");

for(i=0;i<row;i++) {

    for(j=0;j<column;j++) {

        printf("Enter elements [%d][%d]",i,j);

        scanf("%d",&a[i][j]); } }

for(i=0;i<row;i++) {

    for(j=0;j<column;j++) {

        if(i==j&&a[i][j]!=1){

            c++;

            break; }

        else if (i!=j && a[i][j]!=0){

            c++;

            break; } } }

if(c==1) {

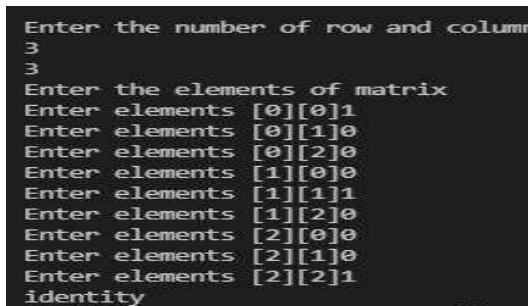
    printf("identity\n"); }

else{ printf("not identity\n"); }

return o;

```

## **OUTPUT**



```

Enter the number of row and column
3
3
Enter the elements of matrix
Enter elements [0][0]1
Enter elements [0][1]0
Enter elements [0][2]0
Enter elements [1][0]0
Enter elements [1][1]1
Enter elements [1][2]0
Enter elements [2][0]0
Enter elements [2][1]0
Enter elements [2][2]1
identity

```

## 5. WAP to input a 2D array of size M\*N and find the sum of individual rows and individual columns.

---

```
#include<stdio.h>

int main(){
    int i,j,row,column,rowsum,columnsum,temp=0;
    printf("Enter the value of row and column\n");
    scanf("%d%d",&row,&column);
    int a[row][column];
    for(i=0;i<row;i++) {
        for(j=0;j<column;j++) {
            printf("Enter the value of [%d][%d] :",i,j);
            scanf("%d",&a[i][j]); } }
    printf("Matrix is .....\\n");
    for(i=0;i<row;i++) {
        for(j=0;j<column;j++) {
            printf("%d\\t",a[i][j]); }
        printf("\\n"); }
    for(i=0;i<row;i++)
    {   rowsum=0;
        for(j=0;j<column;j++) {
            rowsum+=a[i][j]; }
        printf("\\n Sum of all the elements in row %d is %d ",i,rowsum); }
    for(i=0;i<row;i++)
    {   columnsum=0;
        for(j=0;j<column;j++) {
```

```

columnsum+=a[j][i]; }

printf("\n Sum of all the elements in column %d is %d ",i,columnsum);
}

return 0;
}

```

## OUTPUT

```

Enter the value of row and column
3
3
Enter the value of [0][0] :1
Enter the value of [0][1] :1
Enter the value of [0][2] :1
Enter the value of [1][0] :1
Enter the value of [1][1] :1
Enter the value of [1][2] :1
Enter the value of [2][0] :1
Enter the value of [2][1] :1
Enter the value of [2][2] :1
Matrix is .....
1      1      1
1      1      1
1      1      1

Sum of all the elements in row 0 is 3
Sum of all the elements in row 1 is 3
Sum of all the elements in row 2 is 3
Sum of all the elements in column 0 is 3
Sum of all the elements in column 1 is 3
Sum of all the elements in column 2 is 3

```

6. WAP to input 2 matrices from the user and add them.

---

```

#include<stdio.h>
int main(){
    int i,j,row,columns;
    int a[20][20],c[20][20],b[20][20];
    printf("Enter the number of row and columns\n");
    scanf("%d%d",&row,&columns);
    if(row==columns){
        printf("Enter element of 1st matrix :\n");
        for(i=0;i<row;i++){
            for(j=0;j<columns;j++){
                printf("Enter element [%d][%d]",i,j);
                scanf("%d",&a[i][j]); } }
        printf("Enter element of 2nd matrix :\n");
        for(i=0;i<row;i++){
            for(j=0;j<columns;j++){

```

```

        printf("Enter element [%d][%d]",i,j);
        scanf("%d",&b[i][j]); } }
for(i=0;i<row;i++){
    for(j=0;j<columns;j++){
        c[i][j]=a[i][j]+b[i][j]; } }
for(i=0;i<row;i++){
    for(j=0;j<columns;j++){
        printf("%d\t",c[i][j]); }
        printf("\n"); } }
else
printf("addition not possible");
return 0;

```

## OUTPUT

```

Enter the number of row and columns
2
2
Enter element of 1st matrix :
Enter element [0][0]1
Enter element [0][1]2
Enter element [1][0]3
Enter element [1][1]1
Enter element of 2nd matrix :
Enter element [0][0]2
Enter element [0][1]2
Enter element [1][0]2
Enter element [1][1]2
3      4
5      3

```

7. WAP to input a matrix of order M\*N and check if  
it's sparse or dense matrix.

---

```

#include<stdio.h>
int main(){
    int i,j,row,column,count=0;
    int a[10][10];
    printf("Enter the number of rows and columns\n");
    scanf("%d%d",&row,&column);
    for(i=0;i<row;i++)
    { for(j=0;j<column;j++)
        { printf("Enter element [%d] [%d]",i,j);
        scanf("%d",&a[i][j]);
        if(a[i][j]==0){

```

```

        count++; } } }
if(count>(row*column)/2) {
printf("The given matrix is sparsh matrix\n");
}
else
{ printf("The matrix is dense"); }
return o;
}

```

## OUTPUT

```

Enter the number of rows and columns
3
3
Enter element [0] [0]0
Enter element [0] [1]0
Enter element [0] [2]0
Enter element [1] [0]0
Enter element [1] [1]0
Enter element [1] [2]0
Enter element [2] [0]1
Enter element [2] [1]0
Enter element [2] [2]0
The given matrix is sparsh matrix

```

### 8. WAP to input a matrix and print its Lower triangular matrix.

---

```

#include<stdio.h>
int main(){
    int i,j,row,column;
    printf("Enter the number of rows and columns\n");
    scanf("%d%d",&row,&column);
    int a[10][10];
    printf("Enter the elements of the matrix\n");
    for(i=0;i<row;i++)
    { for(j=0;j<column;j++)
        { scanf("%d",&a[i][j]); } }
    printf("\nlower triangular is\n");
    for(i=0;i<row;i++)
    { printf("\n");
        for(j=0;j<column;j++)
        { if(i>j){
            printf("%2d",a[i][j]); }
            else{ printf(" "); } } }
    return 0;
}

```

## OUTPUT

```
Enter the number of rows and columns
3
3
Enter the elements of the matrix
1
2
3
4
5
6
7
8
9

lower triangular is

4
7 8
```

- 
9. WAP to input a 2D array and display diagonal elements in matrix form.
- 

```
#include<stdio.h>

int main(){

int i,j,row,size,column,c=0, a[6][6];

printf("Enter the number of rows and columns\n");

scanf("%d%d",&row,&column);

printf("Enter the elements of matrix1\n");

for(i=0;i<row;i++)

{ for(j=0;j<column;j++)

{ scanf("%d",&a[i][j]);} }

printf("diagonal elements of matrix is :\n");

for(i=0;i<row;i++)

{ for(j=0;j<column;j++)

{ if(i==j)

{ printf("%d\t",a[i][j]); } }}
```

```

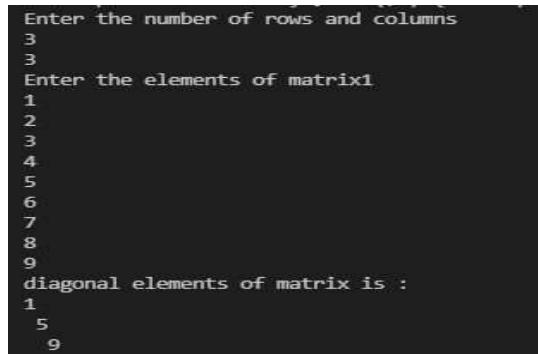
else { printf(" "); } }

printf("\n"); }

return o; }

```

## OUTPUT



```

Enter the number of rows and columns
3
3
Enter the elements of matrix1
1
2
3
4
5
6
7
8
9
diagonal elements of matrix is :
1
5
9

```

10. WAP to input a 2D array and find the sum of its diagonal elements.
- 

```

#include<stdio.h>
int main(){
int i,j,row,size,column,c=0;
printf("Enter the number of rows and columns\n");
scanf("%d%d",&row,&column);
int a[6][6],sum=0;
printf("Enter the elements of matrix1\n");
for(i=0;i<row;i++)
{ for(j=0;j<column;j++)
{ scanf("%d",&a[i][j]); } }
printf("diagonal elements of matrix is :\n");
for(i=0;i<row;i++)
{ for(j=0;j<column;j++) {
if(i==j)
{ printf("%d\t",a[i][j]);
sum=sum+a[i][j]; }
else {
printf(" ");
}
printf("\n");}
printf("sum of diagonal elements is %d",sum);
return o; }

```

## OUTPUT

```
Enter the number of rows and columns
3
3
Enter the elements of matrix1
1
1
1
1
1
1
1
1
diagonal elements of matrix is :
1
1
1
sum of diagonal elements is 3
```

## **STRING**

1. Write a C program to find length of a string with and without function.
- 

```
#include <stdio.h>
#include <string.h>
int main() {
    char a[100];
    int length;
    char str[100];
    int i;
    printf("Enter the String\n");
    gets(str);
    for (i = 0; str[i] != '\0'; ++i);
    printf("Length of String = %d\n", i);
    printf("enter a string again to calculate its length using
strlen function\n");
    scanf("%s", a);
    length = strlen(a);
    printf("Length of the string = %d\n", length);
    return 0;
}
```

## **OUTPUT**

```
Enter the String
Kanishk
Length of String = 7
enter a string again to calculate its length using strlen function
kanishk
Length of the string = 7
```

## 2. Write a C program to count frequency of each character in a string.

---

```
#include<stdio.h>

int main(){
    int i,b,count=0;
    char c;
    char a[100];
    printf("Enter the sentence\n");
    gets(a);
    for(b='a';b<='z';b++)
    { count=0;
        for(i=0;a[i]!='\0';i++)
        { if(a[i]==b)
            { count++; }
        }
        printf("letter %c is repeated %d times\n",b,count);
    } return 0; }
```

### OUTPUT

```
Enter the sentence
hindi hindu hindustan
letter a is repeated 1 times
letter b is repeated 0 times
letter c is repeated 0 times
letter d is repeated 3 times
letter e is repeated 0 times
letter f is repeated 0 times
letter g is repeated 0 times
letter h is repeated 3 times
letter i is repeated 4 times
letter j is repeated 0 times
letter k is repeated 0 times
letter l is repeated 0 times
letter m is repeated 0 times
letter n is repeated 4 times
letter o is repeated 0 times
letter p is repeated 0 times
letter q is repeated 0 times
letter r is repeated 0 times
letter s is repeated 1 times
letter t is repeated 1 times
letter u is repeated 2 times
letter v is repeated 0 times
letter w is repeated 0 times
letter x is repeated 0 times
letter y is repeated 0 times
letter z is repeated 0 times
```

3. Write a C program to count total number of words in a string.
- 

```
#include<stdio.h>
int main(){
    int i,count=1;
    char a[30];
    printf("Enter the sentence\n");
    gets(a);
    for(i=0;a[i]!='\0';i++)
    { if(a[i]==' ')
        { count++; }
    }
    printf("words in this sentence is %d",count);
    return 0;
}
```

### OUTPUT

```
Enter the sentence
GOVIND SHARMA
words in this sentence is 2
```

4. Write a C program to count total number of vowels and consonants in a string.
- 

```
#include<stdio.h>
int main(){
    int i,count=0,cs=0;
    char a[30];
    printf("Enter the string\n");
    gets(a);
    for(i=0;a[i]!='\0';i++)
    {
```

```

if(a[i]=='A'||a[i]=='a'||a[i]=='e'||a[i]=='E'||a[i]=='I'||a[i]=='i'||a[i]=='o'
||a[i]=='O'||a[i]=='u'||a[i]=='U')
{
    count++;
}
else { cs++; }

printf(" total vowels is %d and consonents is %d",count,cs++);

return o;
}

```

## OUTPUT

```

Enter the string
kanishk
total vowels is 2 and consonents is 5
PS U:\C Programs\strings> █

```

- 
5. Write a C program to toggle case of each character of a string.
- 

```

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

int main(){

    char a[100];
    int i;
    printf("Enter the string\n");
    gets(a);
    for(i=0;a[i]!='\0';i++){
        if(a[i]>='a'&&a[i]<='z'){
            a[i]-=32;
        }
        else if(a[i]>='A'&&a[i]<='Z'){
            a[i]+=32;
        }
    }
    printf("string after toggling case is :");
    puts(a);
    return o;
}

```

---

## OUTPUT

```
Enter the string  
KANIshk  
string after toggling case is :kaniSHK
```

- 
6. Write a C program to convert lower case string to uppercase.
- 

```
#include<stdio.h>  
#include<string.h>  
int main(){  
char a[30];  
printf("Enter the string in lowercase\n");  
gets(a);  
strupr(a);  
printf("%s",a);  
return 0; }
```

## OUTPUT

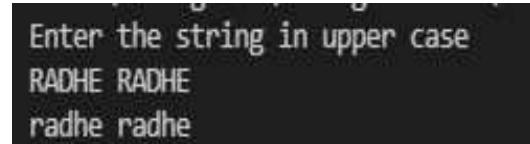
```
Enter the string in lowercase  
jay shree ram  
JAY SHREE RAM
```

- 
7. Write a C program to convert upper case string to lower case .
- 

```
#include<stdio.h>  
#include<string.h>  
int main(){  
char a[30];  
printf("Enter the string in upper case \n");  
gets(a);  
strlwr(a);
```

```
printf("%s",a);
return o; }
```

## OUTPUT



```
Enter the string in upper case
RADHE RADHE
radhe radhe
```

8. Write a C program to find total number of alphabets ,digits or special character in a string.

---

```
#include<stdio.h>

int main(){
char a[100];
int dgt=0,alp=0,spc=0;
printf("Enter the string\n");
gets(a);
for(int i=0;a[i]!='\0';i++){
    if(a[i]>=65 && a[i]<=90|| a[i]>=97 && a[i]<=122)
        alp++;
    else if (a[i]>=48 && a[i]<=57)
        dgt++;
    else
        spc++; }
printf(" total no alphabet is %d ",alp);
printf(" \ntotal no digits  is %d ",dgt);
printf(" \ntotal no special character is %d ",spc);
return o; }
```

## OUTPUT

```
Enter the string  
Kanishk@!$123  
total no alphabet is 7  
total no digits is 3  
total no special character is 3
```

- 
9. Write a C program to check whether a string is palindrome or not.
- 

```
#include<stdio.h>  
  
#include <string.h>  
  
int main()  
{ char a[100];  
    int i,n,c=0;  
    printf("Enter the string : ");  
    gets(a);  
    n=strlen(a);  
    for(i=0;i<n/2;i++)  
    { if(a[i]==a[n-i-1])  
        c++;}  
    if(c==i)  
        printf("string is palindrome");  
    else  
        printf("string is not palindrome");  
    return 0; }
```

## OUTPUT

```
Enter the string : mom  
string is palindrome
```

**10. Write a C program to compare two strings with and without function.**

---

```
#include <stdio.h>
#include <string.h>
int main()
{ char Str1[100],Str2[100];
char stri1[100],stri2[100];
int result, i;
printf("\n Please Enter the First String : ");
gets(Str1);
printf("\n Please Enter the Second String : ");
gets(Str2);
for(i = 0; Str1[i] == Str2[i] && Str1[i] == '\0'; i++);
if(Str1[i] < Str2[i])
{ printf("\n str1 is Less than str2");}
else if(Str1[i] > Str2[i])
{ printf("\n str2 is Less than str1");}
else
{ printf("\n str1 is Equal to str2");}
printf("Enter string again to check without using pre defined function");
printf("\nEnter string 1 :");
gets(stri1);
printf("\nEnter string 2 :");
gets(stri2);
if(strcmp(stri1,stri2)==0)
{ printf("The two strings are EQUAL!!!\n");}
else
{printf("The two string are NOT EQUAL!!!\n");}
```

---

```
return o;}
```

## OUTPUT

```
Please Enter the First String : Kanishk
Please Enter the Second String : Kanishk
str1 is Equal to str2Enter string again to check without using pre defined function
Enter string 1 :Kanishk
Enter string 2 :Kanishk
The two strings are EQUAL!!!
```

## **FUNCTION**

- 1- Write a c program user defined function addition().

```
#include <stdio.h>
int addition(int num1, int num2)
{
    int sum;
    sum = num1+num2;
    return sum;
}

int main()
{
    int var1, var2;
    printf("Enter number 1: ");
    scanf("%d",&var1);
    printf("Enter number 2: ");
    scanf("%d",&var2);
    int res = addition(var1, var2);
    printf ("Output: %d", res);
    return 0;
}
```

## **OUTPUT**

```
Enter number 1: 150
Enter number 2: 260
Output: 410
```

2- Write a program to swap two numbers using call by value.

---

```
#include<stdio.h>
void swap(int a,int b){
    int t=a;
    a=b;
    b=t;
    printf("\nvalues of a = %d and b=%d after swap ",a,b); }
int main(){
    int a,b;
    printf("Enter the value of a and b\n");
    scanf("%d%d",&a,&b);
    printf("values of a = %d and b=%d before swap ",a,b);
    swap(a,b);
    printf("\nnow value of a=%d and b=%d",a,b);
    return 0; }
```

### OUTPUT

```
Enter the value of a and b
2
3
values of a = 2 and b=3 before swap
values of a = 3 and b=2 after swap
now value of a=2 and b=3
```

3- Write a program to calculate binomial coefficient using function.

---

```
#include<stdio.h>
int c(int n, int k);
int main()
{ int n,k;
```

```

printf("Enter n and k : ");
scanf("%d%d",&n,&k);
printf("\nBinomial coefficient\n",c(n,k));
printf("%d\n",c(n,k));
return 0;
}

int c(int n, int k) {
    if(k==0 || k==n)
        return 1;
    return c(n-1,k-1) + c(n-1,k); }
```

## OUTPUT

```

Enter n and k : 5
2
Binomial coefficient
10
```

- 4-** .What do you mean by call by value ?Give one example.

**call by value :** In this parameter passing method, values of actual parameters are copied to function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of caller.

### **Ex-swapping of two numbers**

- 5-** Write a program to calculate  $x^n$  without using library function pow() but using user defined function.

```
#include<stdio.h>

void power(int,int);
```

```

void main() {
    int b,e;
    printf("Enter the base\n");
    scanf("%d",&b);
    printf("Enter the exponent\n");
    scanf("%d",&e);
    power(b,e);}
void power(int b,int e){
    int power=1;
    while(e>0)
    { power=power*b;
        e--; }
    printf("The power of the given number is %d",power); }
```

## OUTPUT

```

Enter the base
2
Enter the exponent
3
```

6- Define function. What are the types of function in c? Categorize user defined functions?

---

**Function-**A function is a group of statements that together perform a task. Every C program has at least one function, which is main(), and all the most trivial programs can define additional functions.

**Types of function-There are two types of function in C programming:**

1. Standard library functions.

## 2. User-defined functions.

### Types of user defined functions in C

Category I: Functions with no arguments and no return values.

Category 2: Functions with no arguments and with return values.

Category 3: Functions with arguments and no return values.

Category 4: Functions with arguments and with return values.

- 7- Discuss the following terms–function declaration ,function definition ,actual and formal arguments , calling function and called function with suitable example.

---

**Function declaration** -A function declaration tells the compiler about a function's name, return type, and parameters.

**Function definition**-A function definition provides the actual body of the function.

**Actual and formal arguments**- The variables declared in the function prototype or definition are known as Formal arguments.

The values that are passed to the called function from the main function are known as Actual arguments.

The actual arguments and formal arguments must match in number, type, and order.

**Calling function** - A function call is an important part of the C programming language. It is called inside a program whenever it is required to call a function. It is only called by its name in the main() function of a program. We can pass the parameters to a function calling in the main() function.

**Called function**- A called function performs a defined task and when its return statement is executed or when its function-ending closing brace is reached, it returns the program control back to the main program

```
Ex- #include <stdio.h>
void callee(void) {
    puts("I am the called function.");
```

```

}

void caller(void) {
    puts("I am the calling function.");
    callee(); // Call the function 'callee()'.

}

int main() {
    caller();
}

```

## **8- Write any three advantages of using function.**

---

### **Ans. Advantages of using function-**

1. Use of functions enhances the readability of a program. A big code is always difficult to read. Breaking the code in smaller Functions keeps the program organized, easy to understand and makes it reusable.
2. The C compiler follows *top-to-down* execution, so the control flow can be easily managed in case of functions. The control will always come back to the main() function.

## **9- Write a program in C to print all perfect numbers in given range using the function**

---

```

#include <stdio.h>

int checkPerfect(int n1);

void PerfectNumbers(int stLimit, int enLimit);

int main()
{
    int stLimit, enLimit;
    printf("\n\n Function : perfect numbers in a given range :\n");
    printf(" Input lowest search limit of perfect numbers : ");

```

```

scanf("%d", &stLimit);
printf(" Input highest search limit of perfect numbers : ");
scanf("%d", &enLimit);

printf("\n The perfect numbers between %d to %d are : \n", stLimit,
enLimit);

PerfectNumbers(stLimit, enLimit);
printf("\n\n");
return 0;
}

int checkPerfect(int n1)
{
    int i, sum;

    sum = 0;
    for(i=1; i<n1; i++)
    {
        if(n1 % i == 0)
        {
            sum += i; } }
    if(sum == n1)
        return 1;
    else
        return 0; } void PerfectNumbers(int stLimit, int enLimit)
{ while(stLimit <= enLimit)
{
    if(checkPerfect(stLimit))
    { printf(" %d ", stLimit);

```

```
    }stLimit++;
}
```

## OUTPUT

```
Function : perfect numbers in a given range :
Input lowest search limit of perfect numbers : 1
Input highest search limit of perfect numbers : 100

The perfect numbers between 1 to 100 are :
6 28
```

---

### **10- write a c program using function to check weather the number is prime or not.**

---

```
#include <stdio.h>

void prime();
void prime(int n)
{ int i, count = 0;
  for (i = 1; i < n; i++) {
    if (n % i == 0)
      { count++; }
  }
  if (count == 1)
    printf("prime\n");
  else
    { printf("not prime\n"); }
}

void main()
{
  int n;
  printf("Enter the value of n\n");
  scanf("%d", &n);
  prime(n); }
```

## OUTPUT

```
Enter the value of n  
3  
prime
```