

# LAB MANUAL

## SEMESTER- I

### M.sc DEGREE PROGRAMME Lab-I - Programming in C++ (502105)



Prepared by

**Dr.RM.Vidhyavathi**

**Assistant Professor**  
**DEPARTMENT OF BIOINFORMATICS**  
**(DST-FIST & PURSE Sponsored Department)**  
**ALAGAPPA UNIVERSITY**  
**(A State University Reaccredited with 'A' Grade by NAAC)**  
**Karaikudi-630 004, Tamil Nadu, India.**

<b>EX.NO</b>	<b>LIST OF PROGRAMS</b>
1	Arithmetic operators
2	Logical operator
3	Size of variables
4	Simple interest and compound interest
5	Fahrenheit to Celsius
6	Gross salary of the employee
7	Student mark list
8	Eligibility for voting
9	Sorting in ascending order
10	Complementary DNA
11	Standard Genetic Code Table
12	Convert The DNA to RNA
13	Amino Acids
14	Finding Mismatch

## **EX.NO:1      ARITHMETIC OPERATOR**

**AIM:**

Write a c program for addition, subtraction, multiplication, division, modulus using arithmetic operators.

**ALGORITHM:**

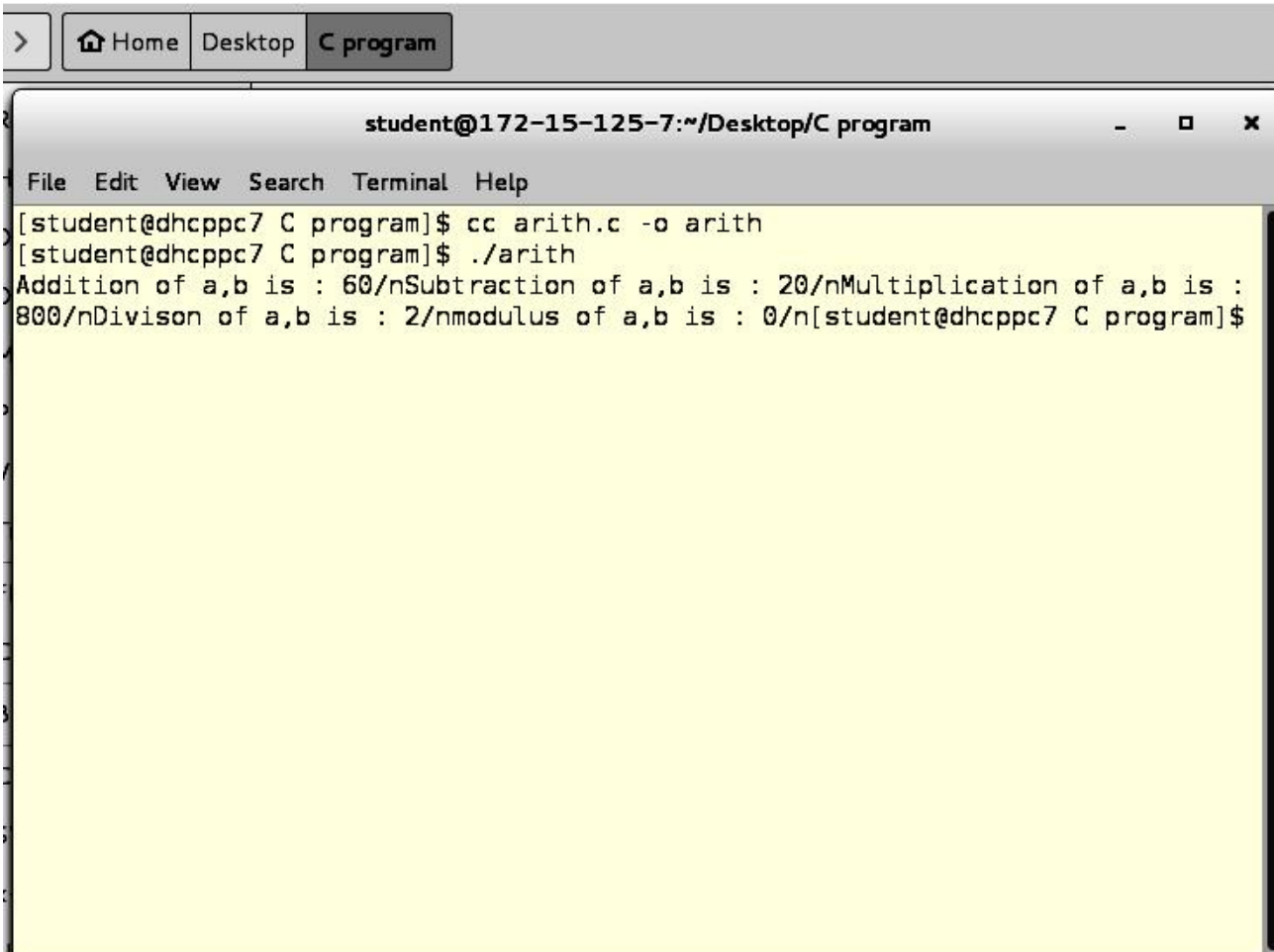
1. Start the program.

1. Declare the variable.
2. Declare the arithmetic operations.
3. Save the program with (.c).
4. Close the program.
5. Run the terminal and obtain the result.
6. Stop the program.

### ARITHMETIC OPERATIONS

```
#include <stdio.h>
int main()
{
int a=40,b=20, add,sub,mul,div,mod;
add = a+b;
sub = a-b;
mul = a*b;
div = a/b;
mod = a%b;
printf("Addition of a, b is : %d\n", add);
printf("Subtraction of a, b is : %d\n", sub);
printf("Multiplication of a, b is : %d\n", mul);
printf("Division of a, b is : %d\n", div);
printf("Modulus of a, b is : %d\n", mod);
}
```

**OUTPUT:**



The image shows a terminal window with a yellow background. The window title is "student@172-15-125-7:~/Desktop/C program". The terminal output is as follows:

```
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc arith.c -o arith
[student@dhcppc7 C program]$ ./arith
Addition of a,b is : 60/nSubtraction of a,b is : 20/nMultiplication of a,b is :
800/nDivison of a,b is : 2/nmodulus of a,b is : 0/n[student@dhcppc7 C program]$
```

## EX.NO:2 LOGICAL OPERATORS

**AIM:**

Write a c program for comparison of three numbers using logical operator

## ALGORITHM:

1. Start the program.
2. Declare the variable.
3. Declare the 3 numbers for the comparison of using logical operator.
4. Save the program with (.c).
5. Close the program.
6. Run the terminal and obtain the result.
7. Stop the program.

## COMPARITION OF 3 NUMBERS USING LOGICAL OPERATORS (WITHOUT VALUE)

```
#include <stdio.h>
int main()
{
int m=40,n=20;
int o=20,p=30;
if (m>n && m !=0)
{
printf("&& Operator : Both conditions are true\n");
}
if (o>p || p!=20)
{
printf("|| Operator : Only one condition is true\n");
}
if (!(m>n && m !=0))
{
```

```
printf("! Operator : Both conditions are true\n");
}
else
{
printf("! Operator : Both conditions are true. " \
"But, status is inverted as false\n");
}
}
```

## OUTPUT:



```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc logic.c -o logic
[student@dhcppc7 C program]$ ./logic
&& Operator:Both conditions are true
|| OPerator:Only one condition is true
!Operation:Both conditions are true.But,Status is inverted as false
[student@dhcppc7 C program]$
```

## **EX.NO:3          SIZES OF VARIABLES**

### **AIM:**

Write a c program for size of variables.

### **ALGORITHM:**

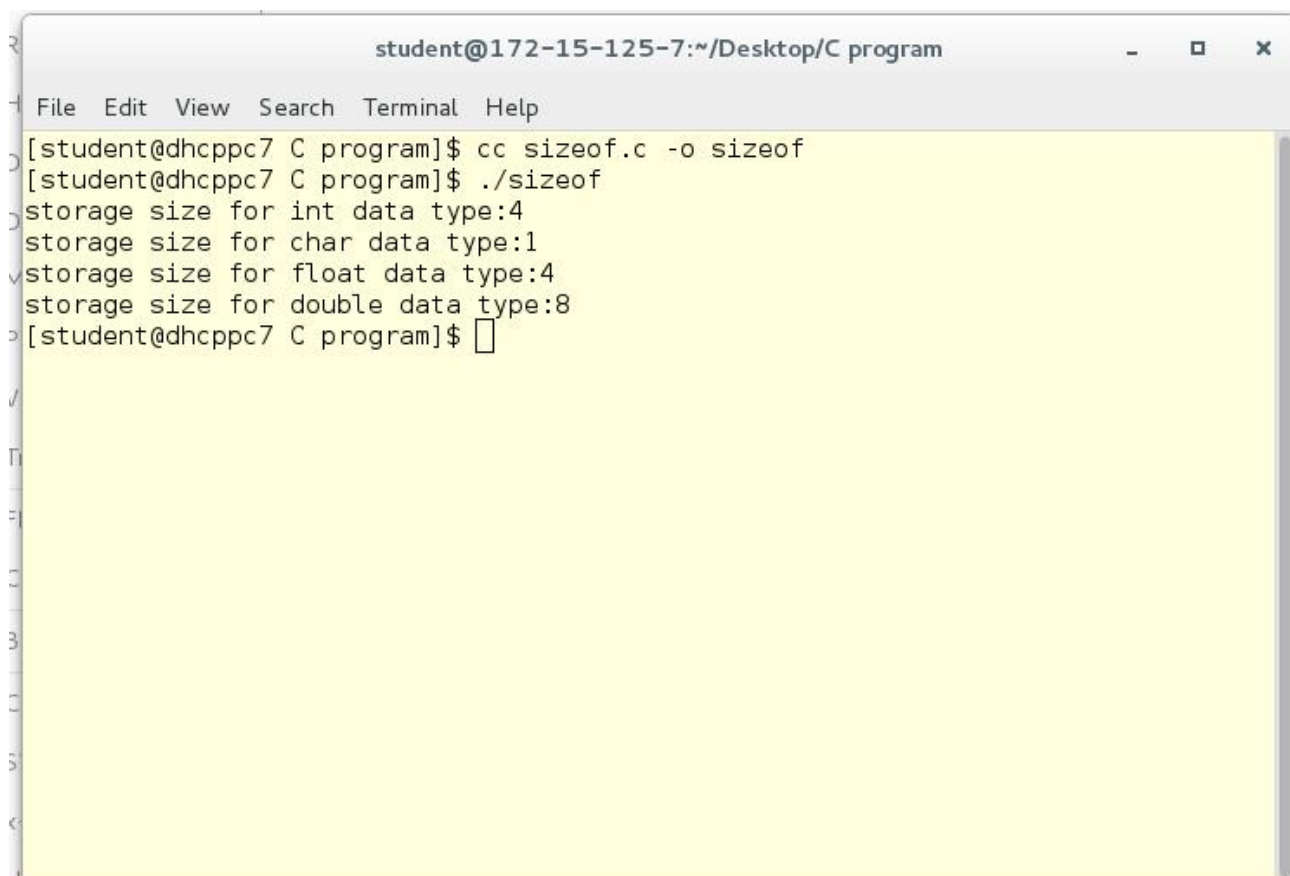
1. Start the program.
2. Declare the variable.
3. Save the program with (.c).
4. Close the program.
5. Run the terminal and obtain the result.
6. Stop the program.

### **SIZE OF ( ) FUNCTION**

```
#include <stdio.h>
#include <limits.h>
int main()
{
int a;
char b;
float c;
double d;
printf("storage size for int data type:%d\n",sizeof (a));
```

```
printf("storage size for char data type:%d\n",sizeof (b));  
printf("storage size for float data type:%d\n",sizeof (c));  
printf("storage size for double data type:%d\n",sizeof (d));  
return 0;  
}
```

## OUTPUT



```
student@172-15-125-7:~/Desktop/C program  
File Edit View Search Terminal Help  
[student@dhcppc7 C program]$ cc sizeof.c -o sizeof  
[student@dhcppc7 C program]$ ./sizeof  
storage size for int data type:4  
storage size for char data type:1  
storage size for float data type:4  
storage size for double data type:8  
[student@dhcppc7 C program]$
```

## EX.NO:4            SIMPLE INTEREST AND COMPOUND INTEREST

### AIM:

Write the c program for simple and compound interest.

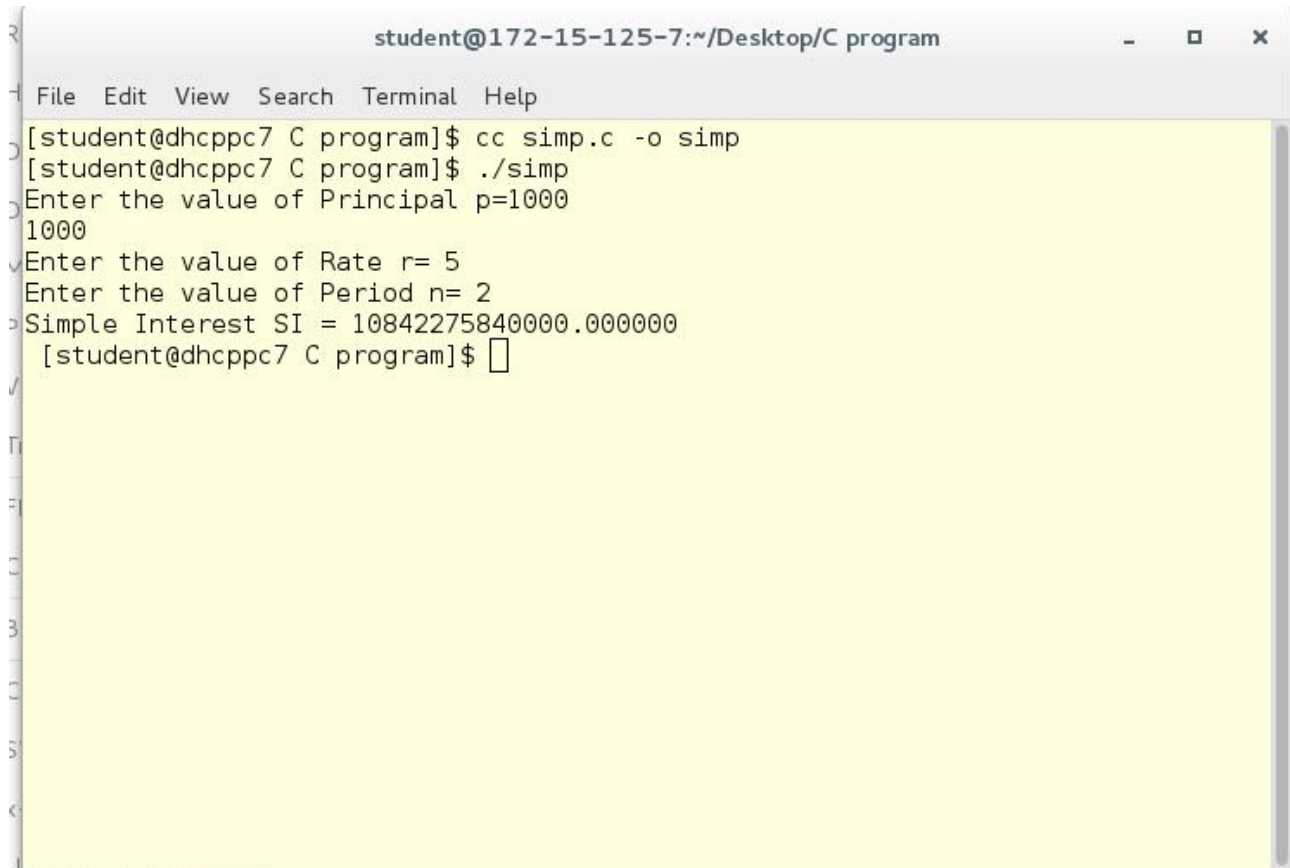
### SIMPLE INTEREST

```
#include<stdio.h>
#include<math.h>
int main()
{
float p,r,SI;
int n;
printf("Enter the value of Principal p =");
scanf("%f",&p);
printf("Enter the value of Rate r = ");
scanf("%f",&r);
printf("Enter the value of Period n = ");
scanf("%d \n",&n);
SI = (p*r*n)/100;
printf("Simple Interest SI=%f \n",SI);
}
```

### COMPOUND INTEREST

```
#include <stdio.h>
int main()
{
float amount, rate, time, CI;
printf("Enter principle (amount): ");
scanf("%f", &amount);
printf("Enter time: ");
scanf("%f", &time);
printf("Enter rate: ");
scanf("%f", &rate);
CI = amount * (((1 + rate / 100), time)-1);
printf("Compound Interest = %f", CI); return 0; }
```

## OUTPUT:

A terminal window titled "student@172-15-125-7:~/Desktop/C program" with standard window controls. The terminal shows the compilation and execution of a C program named "simp.c". The user enters the principal value 1000, the rate 5, and the period 2. The program outputs "Simple Interest SI = 10842275840000.000000".

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc simp.c -o simp
[student@dhcppc7 C program]$ ./simp
Enter the value of Principal p=1000
1000
Enter the value of Rate r= 5
Enter the value of Period n= 2
Simple Interest SI = 10842275840000.000000
[student@dhcppc7 C program]$
```

**EX.NO:5 FAHRENHEIT TO CELSIUS**

**AIM:**

To write a c program to find a Fahrenheit to Celsius.

### **Fahrenheit and celcius**

```
#include<stdio.h>

int main()
{
float cent,faren;
printf("FAHRENHEIT TO CELSIUS & CELSIUS TO FAHRENHEIT\n");
printf("-----\n");
printf("Enter fahrenheit value:\n");
scanf("%f",&faren);
cent = 5.0/9.0*(faren - 32);
faren = ((9.0/5.0)*cent+32);
printf("\ncentigrade value is : %f",cent);
printf("\nfaren value is : %f",faren);
}
```

### **OUTPUT**

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc fahren.c -o fahren
[student@dhcppc7 C program]$ ./fahren
FAHRENHEIT TO CELSIUS & CELSIUS TO FAHRENHEIT
.....
Enter fahrenheit value:
7
/ncentigrade value is : -13.888889/nfaren value is : 6.999999[student@dhcppc7 C
program]$
```

## **EX.NO:6 GROSSPAY AND NETPAY**

### **AIM:**

Write the c program for gross pay of the employee.

### **GROSSPAY AND NETPAY**

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

```
int main()
{
char name[20];
float basic,da,hra,pf,ns,wc,gpay,npay,ded;
printf("GROSSPAY AND NETPAY");
printf("Enter the employee name:");
scanf("%s",&name);
printf("\nEnter the basic pay:");
scanf("%f",&basic);
if(basic>10)
{
hra=0.15*basic;
da=0.9*basic;
gpay=basic+da+hra;
pf=0.1*basic;
wc=200;
ded=pf+wc;
npay=gpay-ded;
printf("\nEMPLOYEE NAME :%s",name);
printf("\nBASIC SALARY :%f",basic);
printf("\nDEARNESS ALLOWANCE :%f",da);
printf("\nHOUSE RENT ALLOWANCE:%f",hra);
printf("\nPROVIDENT FUND :%f",pf);
printf("\nDED :%f",ded);
printf("\nGROSSPAY :%f",gpay);
printf("\nNETPAY :%f",npay);
}
}
```

## OUTPUT

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc gross.c -o gross
[student@dhcppc7 C program]$ ./gross
GROSSPAY AND NETPAYEnter the employee name:sidu

Enter the basic pay:5000

EMPLOYEE NAME :sidu
BASIC SALARY :5000.000000
DEARNNESS ALLOWANCE :4500.000000
HOUSE RENT ALLOWANCE:750.000000
PROVIDENT FUND :500.000000
DED :700.000000
GROSSPAY :10250.000000
NETPAY :9550.000000[student@dhcppc7 C program]$
```

## EX.NO.7 STUDENT MARK LIST

### AIM:

write a c program to find a student mark list.

### ALGORITHM:

1. Start the program.

2. Declare the variable.
3. Save the program with (.c).
4. Close the program.
5. Run the terminal and obtain the result.
6. Stop the program.

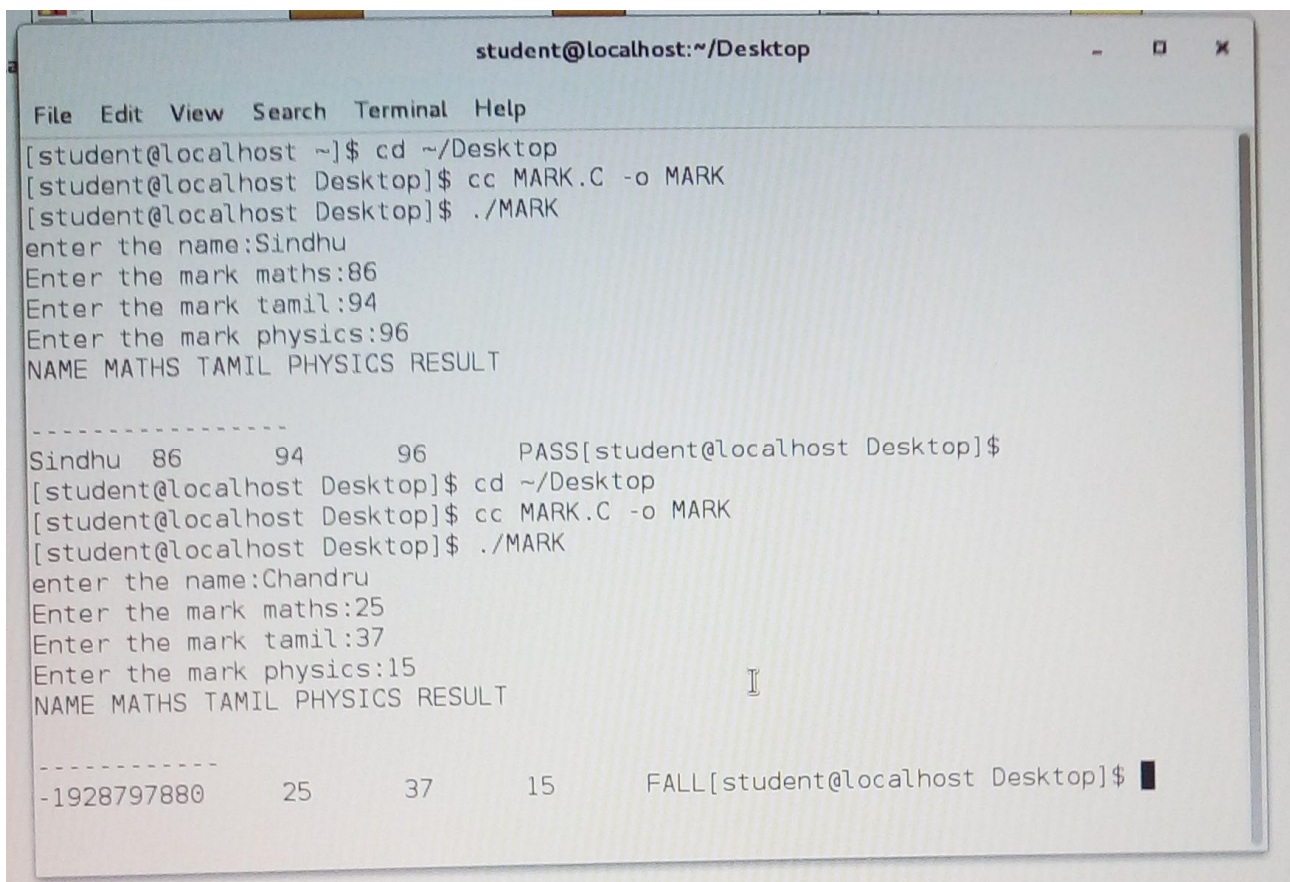
### **Mark statement**

```
#include<stdio.h>

int main()
{
int i,m1[10],m2[10],m3[10];
char name[20][30];
for(i=1;i<=3;i++)
{
printf("enter the name:");
scanf("%s",& name[i]);
printf("Enter the mark maths:");
scanf("%d",&m1[i]);
printf("Enter the mark tamil:");
scanf("%d",&m2[i]);
printf("Enter the mark physics:");
scanf("%d",&m3[i]);
}
printf("NAME MATHS TAMIL PHYSICS RESULT\n");
for(i=1;i<=3;i++)
{
if(m1[i]>40 && m2[i]>40 && m3[i]>40)
{
printf("\n-----\n");
printf("%s\t",name[i]);
printf("%d\t",m1[i]);
printf("%d\t",m2[i]);
printf("%d\t",m3[i]);
printf("PASS");
}
else
{
printf("\n-----\n");
printf("%d\t",name[i]);
printf("%d\t",m1[i]);
printf("%d\t",m2[i]);
printf("%d\t",m3[i]);
printf("FALL");
}
```

```
}  
}  
}
```

## OUTPUT



```
student@localhost:~/Desktop  
File Edit View Search Terminal Help  
[student@localhost ~]$ cd ~/Desktop  
[student@localhost Desktop]$ cc MARK.C -o MARK  
[student@localhost Desktop]$ ./MARK  
enter the name:Sindhu  
Enter the mark maths:86  
Enter the mark tamil:94  
Enter the mark physics:96  
NAME MATHS TAMIL PHYSICS RESULT  
-----  
Sindhu 86 94 96 PASS[student@localhost Desktop]$  
[student@localhost Desktop]$ cd ~/Desktop  
[student@localhost Desktop]$ cc MARK.C -o MARK  
[student@localhost Desktop]$ ./MARK  
enter the name:Chandru  
Enter the mark maths:25  
Enter the mark tamil:37  
Enter the mark physics:15  
NAME MATHS TAMIL PHYSICS RESULT  
-----  
-1928797880 25 37 15 FALL[student@localhost Desktop]$
```

### EX.NO.8 ELIGIBILITY FOR VOTING

AIM:

To write a c program to find the eligibility for voting

## **Voting**

```
#include<stdio.h>
int main()
{
int age,d;
char name[20];
printf("ELIGIBILITY FOR VOTING:\n");
printf("-----\n");
printf("Enter the Name:");
scanf("%s",&name);
printf("Enter the Age:");
scanf("%d",&age);
if(age>18)
{
printf("NAME:%s\n",name);
printf("AGE:%d\n",age);
printf("HE IS ELIGIBLE TO VOTE");
}
else
{
printf("NAME:%s\n",name);
printf("AGE:%d\n",age);
d=18-age;
printf("\n%d years has to be wait for voting",d);}
}
```

## **OUTPUT**

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcpc7 C program]$ cc eligible.c -o eligible
[student@dhcpc7 C program]$ ./eligible
ELIGIBILITY FOR VOTING:
.....
Enter the Name :anu
Enter the Age :20
NAME:anu
AGE:20
HE IS ELIGIBLE TO VOTE[student@dhcpc7 C program]$
```

## **EX.NO:9      SORTING IN ASCENDING ORDER**

### **AIM:**

Write the c program for sorting in ascending order.

### **ASCENDING ORDER**

```
#include<stdio.h>
int main()
{
int i,j,n,t=0;
```

```
int a[10];
printf("-----\n");
printf("ASCENDING ORDER IN THE VALUES\n");
printf("-----\n");
printf("enter the number of elements:\n");
scanf("%d",&n);
printf("Enter the values one by one:\n");
for(i=0;i<n;i++)
{
scanf("%d",&a[i]);
}
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(a[i]>=a[j])
{
t=a[i];
a[i]=a[j];
a[j]=t;
}
}
}
printf("sorted:\n");
for(i=0;i<n;i++)
{
printf("%d",a[i]);
printf("\n");
}
}
```

## OUTPUT

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc sort.c -o sort
[student@dhcppc7 C program]$ ./sort
.....
ASCENDING ORDER IN THE GIVEN VALUES
.....
Enter the no of elements :
5
Enter the values one by one:
1
2
3
4
4
sorted :
1
2
3
4
4
[student@dhcppc7 C program]$
```

## **EX.NO:10            COMPLEMENTARY DNA**

### **AIM:**

Write a c program for complementary DNA sequence

## COMPLEMENTARY DNA

```
#include<stdio.h>
main ()
{
int n;
printf("\n\nEnter length of sequence:");
scanf("%d",&n);
char seq[n], com[n];
int i;
printf("\n\nENTER THE SEQUENCE:");
scanf("%s",&seq);
for(i;i<n;i++)
{
if(seq[i]=='A')
com[i]='T';
else if(seq[i]=='T')
com[i]='A';
else if(seq[i]=='G')
com[i]='C';
else if(seq[i]=='C')
com[i]='G';
else if(seq[i]==' ')
com[i]='_';
else if(seq[i]!='T' && seq[i]!='A' && seq[i]!='G' && seq[i]!='C')
com[i]='*';
}
printf("\n\n%s\n\n",com);
printf("\n\n* is non-DNA bases\n\n");
}
```

## OUTPUT

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc comp.c -o comp
[student@dhcppc7 C program]$ ./ comp
bash: ./: Is a directory
[student@dhcppc7 C program]$ ./comp

Enter length of sequence:6

ENTER THE SEQUENCE:ATGC

TACG**

* is non-DNA bases

[student@dhcppc7 C program]$ □
```

## **EX.NO:11          STANDARD GENETIC CODE TABLE**

### **AIM:**

Write a c program for Standard genetic code table

### **Standard genetic code table**

```

#include<stdio.h>
int main()
{
char *text[20]={

    "UUU","UCU","UAU","UGU","U","UUC","UCC",
    "UAC","UGC","C","UUA","UCA","UAA","UGA","A",
    "UUG","UCG","UAG","UGG","G"};

char *text1[20]={

    "CUU","CCU","CAU","CGU","U","CUC","CCC",
    "CAC","CGC","C","CUA","CCA","CAA","CGG","A",
    "CUG","CCG","CAG","CGG","G"};

char *text2[20]={

    "AUU","ACU","AAU","AGU","U","AUC","ACC",
    "AAC","AGC","C","AUA","ACA","AAA","AGA","A",
    "AUG","ACG","AAG","AGG","G"};

char *text3[20]={

    "GUU","GCU","GAU","GGU","U","GUC","GCC",
    "GAC","GGC","C","GUA","GCA","GAA","GGA","A",
    "GUG","GCG","GAG","GGG","G"};

int i,j=0;
/*now,display them*/
printf("\t STANDARD GENETIC CODE TABLE\n");
printf("\t ***** \n");
printf("\tU\tC\tA\tG\n");
printf("\nU");
while(j<19)
{
for(i=0;i<5;i++)
    {
    printf("\t%s",text[j]);
    j++;
    }
printf("\n");
}
}

```

**OUTPUT:**

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc genetic.c -o genetic
[student@dhcppc7 C program]$ ./genetic
    STANDARD GENETIC CODE TABLE
    *****
          U      C      A      G
U      UUU     UCU     UAU     UGU     U
      UUC     UCC     UAC     UGC     C
      UUA     UCA     UAA     UGA     A
      UUG     UCG     UAG     UGG     G
C
A
G[student@dhcppc7 C program]$
```

**EX.NO:12      CONVERT THE DNA TO RNA**

**AIM:**

Write A C Program for **Convert the DNA to RNA**

## CONVERT THE DNA TO RNA

```
#include<stdio.h>
int main()
{
char s[50];
int i=0,n;
printf("Convert the DNA to RNA\n");
printf("\nEnter the DNA:\n");
scanf("%s",&s);
for(i=0;i<50;i++)
{
if(s[i]=='T')
{
s[i]='U';
}
}
printf("THE RNA IS:%s\n",s);
}
```

## OUTPUT

```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc DNA.c -o DNA
[student@dhcppc7 C program]$ ./DNA
Convert the DNA to RNA

Enter the DNA:
T
THE RNA IS:U
[student@dhcppc7 C program]$
```

**EX.NO:13**      **AMINO ACIDS**

**AIM:**

Write A C Program for Amino Acids

## AMINO ACIDS

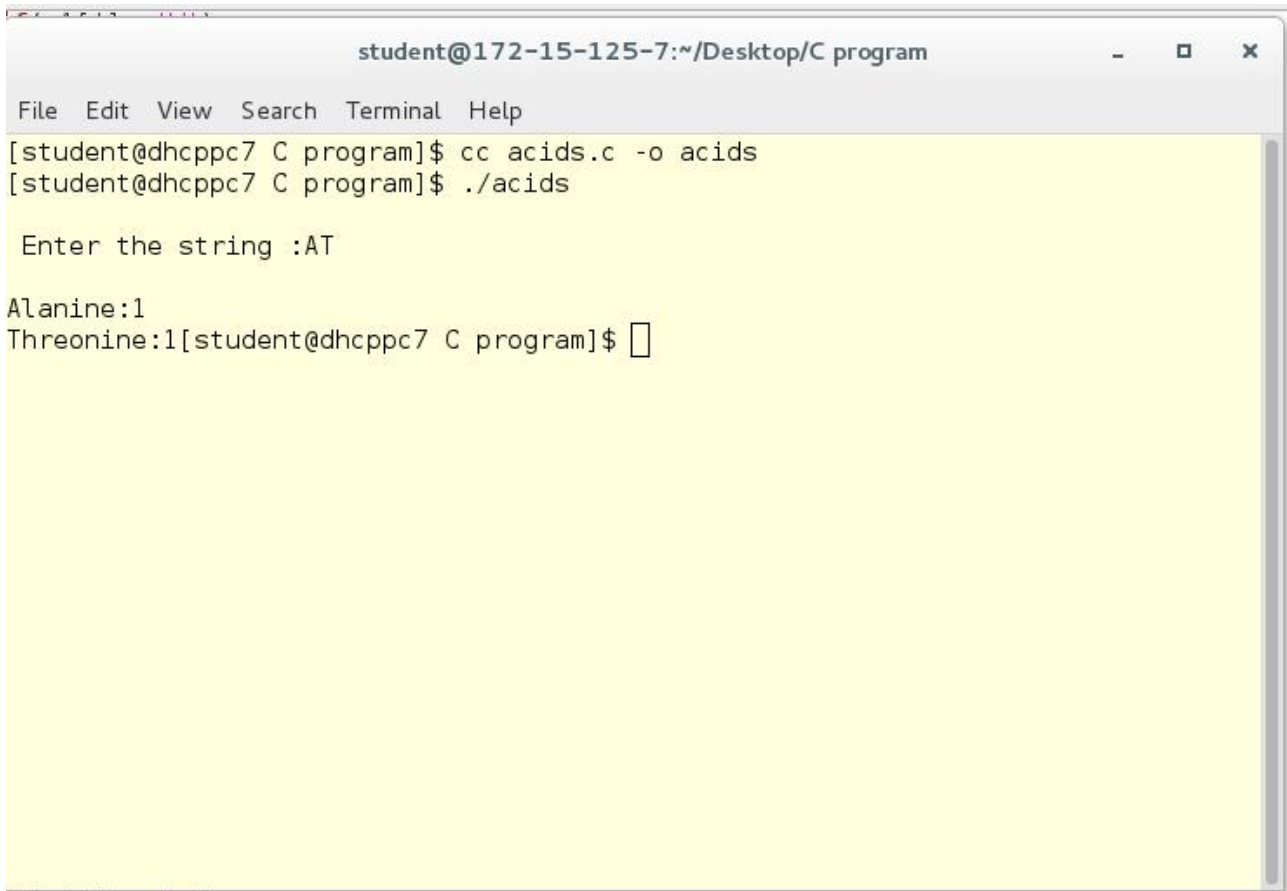
```
#include<stdio.h>
int main()
{
char s1[50];
int j=0,a=0,v=0,i=0,l=0,m=0,f=0,y=0,w=0,k=0,r=0,h=0,
d=0,s=0,t=0,e=0,n=0,q=0,c=0,g=0,p=0;
printf("\n Enter the string :");
scanf("%s",&s1);
while(s1[j]!='\0')
{
if(s1[j]=='A')
{
a=a+1;
}
else if(s1[j]=='V')
{
v=v+1;
}
else if(s1[j]=='I')
{
i=i+1;
}
else if(s1[j]=='L')
{
l=l+1;
}
else if(s1[j]=='M')
{
m=m+1;
}
else if(s1[j]=='F')
{
f=f+1;
}
else if(s1[j]=='Y')
{
y=y+1;
}
else if(s1[j]=='W')
{
w=w+1;
}
else if(s1[j]=='K')
```

```
{
k=k+1;
}
else if(s1[j]=='r')
{
r=r+1;
}
else if(s1[j]=='H')
{
h=h+1;
}
else if(s1[j]=='D')
{
d=d+1;
}
else if(s1['S'])
{
s=s+1;
}
else if(s1[j]=='T')
{
t=t+1;
}
else if(s1[j]=='E')
{
e=e+1;
}
else if(s1[j]=='N')
{
n=n+1;
}
else if(s1[j]=='Q')
{
q=q+1;
}
else if(s1[j]=='C')
{
c=c+1;
}
else if(s1[j]=='G')
{
g=g+1;
}
else if(s1[j]=='P')
{
```

```
p=p+1;
}
j=j+1;
}
if(a!=0)
{
printf("\nAlanine:%d",a);
}
if(v!=0)
{
printf("\nValine:%d",v);
}
if(i!=0)
{
printf("\nIsoleucine:%d",i);
}
if(l!=0)
{
printf("\nLeucine:%d",l);
}
if(m!=0)
{
printf("\nMethionine:%d",m);
}
if(f!=0)
{
printf("\nPhenylalanine:%d",f);
}
if(y!=0)
{
printf("\nTyrosine:%d",y);
}
if(w!=0)
{
printf("\nTryptophan:%d",w);
}
if(k!=0)
{
printf("\nLysine:%d",k);
}
if(r!=0)
{
printf("\nArginine:%d",r);
}
if(h!=0)
```

```
{
printf("\nHistidine:%d",h);
}
if(d!=0)
{
printf("\nAspartate:%d",d);
}
if(s!=0)
{
printf("\nSerine:%d",s);
}
if(t!=0)
{
printf("\nThreonine:%d",t);
}
if(e!=0)
{
printf("\nGlutamate:%d",e);
}
if(n!=0)
{
printf("\nAsparagine:%d",n);
}
if(q!=0)
{
printf("\nGlutamine:%d",q);
}
if(c!=0)
{
printf("\nCysteine:%d",c);
}
if(g!=0)
{
printf("\nGlycine:%d",g);
}
if(p!=0)
{
printf("\nProline:%d",p);
}
}
```

# OUTPUT



```
student@172-15-125-7:~/Desktop/C program
File Edit View Search Terminal Help
[student@dhcppc7 C program]$ cc acids.c -o acids
[student@dhcppc7 C program]$ ./acids

Enter the string :AT

Alanine:1
Threonine:1[student@dhcppc7 C program]$
```

## EX.NO:14 FINDING MISMATCH

### AIM:

Write a C Program to find the finding mismatch sequence

### FINDING MISMATCH

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

```

int main ()
{
    int flag;
    char s1[1000], s2[1000];

    printf("Input first string\n");
    scanf("%s",&s1);

    printf("Input second string\n");
    scanf("%s",&s2);

    /** Passing smaller length string first */

    if (strlen(s1) < strlen(s2))
        flag = check_subsequence(s1, s2);
    else
        flag = check_subsequence(s2, s1);

    if (flag)
        printf("YES\n");
    else
        printf("NO\n");

    return 0;
}

int check_subsequence (char a[], char b[]) {
    int c, d;

    c = d = 0;

    while (a[c] != '\0') {
        while ((a[c] != b[d]) && b[d] != '\0') {
            d++;
        }
    }
}

```

```
    }
    if (b[d] == '\0')
        break;
    d++;
    c++;
}
if (a[c] == '\0')
    return 1;
else
    return 0;
}
```

## **OUTPUT**

I

student@172-15-125-7:~/Desktop/C program

File Edit View Search Terminal Help

```
[student@dhcppc7 C program]$ cc mismatch.c -o mismatch
```

```
[student@dhcppc7 C program]$ ./mismatch
```

```
Input first string
```

```
ATGC
```

```
Input second string
```

```
CGTA
```

```
NO
```

```
[student@dhcppc7 C program]$
```