Lab manual Programs   (week 1 - 11)

Archana Uppala B

B181867

Week 1

Tutorial 1: Problem solving using computers:

Lab1: Familiarization with programming environment

Week 2

Tutorial 2: Variable types and type conversions:

Lab 2: Simple computational problems using arithmetic expressions

1))

1)Problem statement:

```
1. Write a C program to find the area of a circle u
sing the formula:
//Area = PI * r2
```

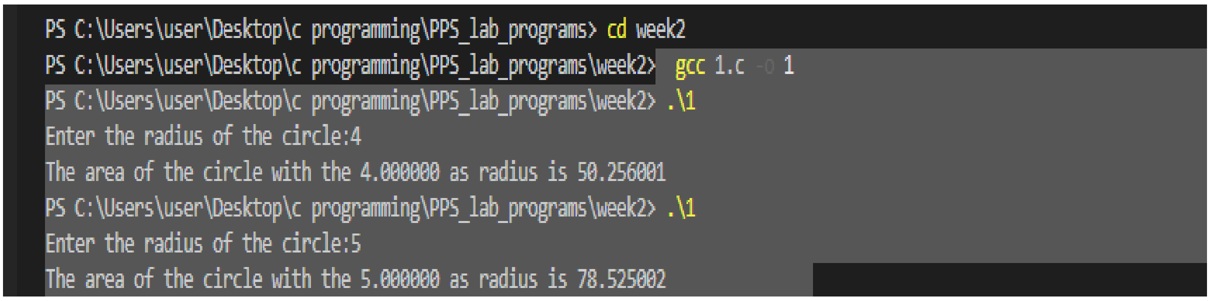
2)Pseudo code:

```
Start
Define NUMBER r, area.
INPUT r.
area=3.14*r*r.
OUTPUT area
End
```

3)Executable C program :

```
# include <stdio.h>
# define PI 3.141
int main()
{
    float r,Area;

    printf("Enter the radius of the circle:");
    scanf("%f",&r);
    Area=PI*r*r;
    printf("The area of the circle with the %f as radius is
%f",r,Area);
    return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs> cd week2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> gcc 1.c -o 1
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\1
Enter the radius of the circle:4
The area of the circle with the 4.000000 as radius is 50.256001
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\1
Enter the radius of the circle:5
The area of the circle with the 5.000000 as radius is 78.525002
```

5)Observations:

Define the constant PI

2))

1)Problem statement:

```
Write a C program to find the area and volume of sphere.
Formulas are: Area =4*PI*R*R Volume = 4/3*PI*R*R*R
```

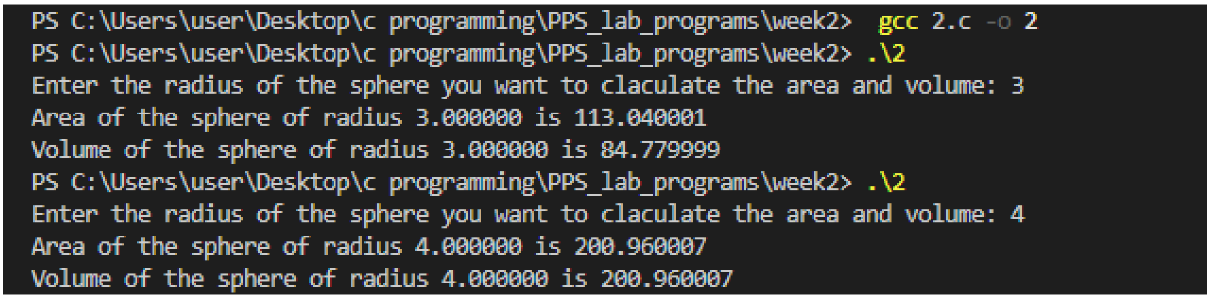
2)Psuedo code:

```
Start

Define PI
Define R
Input R
Area=4*PI*R*R
Volume=(4/3)*PI*R*R*R
Output area and volume
End
```

3)Executable C program :

```c
# include <stdio.h>
# define PI 3.14
int main()
{
    float R;
    printf("Enter the radius of the sphere you want to claculate the area and volume: ");
    scanf("%f",&R);
     float Area=4*PI*R*R;
     float Volume=(4/3)*PI*R*R*R;
    printf("Area of the sphere of radius %f is %f\n",R,Area);
    printf("Volume of the sphere of radius %f is %f\n",R,Volume);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2>  gcc 2.c -o 2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\2
Enter the radius of the sphere you want to claculate the area and volume: 3
Area of the sphere of radius 3.000000 is 113.040001
Volume of the sphere of radius 3.000000 is 84.779999
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\2
Enter the radius of the sphere you want to claculate the area and volume: 4
Area of the sphere of radius 4.000000 is 200.960007
Volume of the sphere of radius 4.000000 is 200.960007
```

3))

1)Problem statement:

```
Write a C program to convert centigrade into Fahrenheit.
Formula: C= (F-32)/1.8.
```

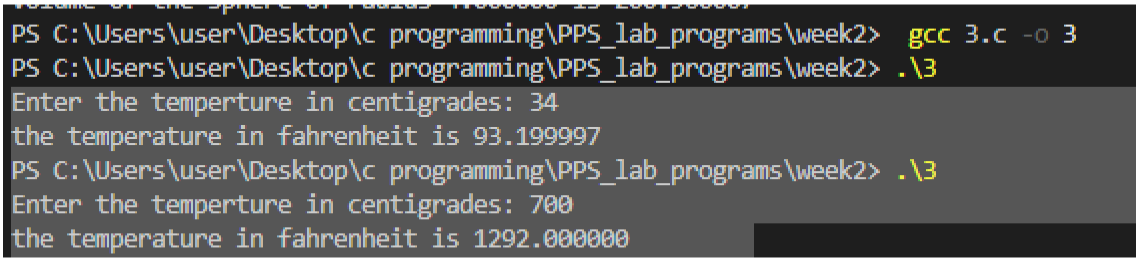
2)Psuedo code:

```
Start
Define c and f
Input the temperature in centigrade(c)
f=c*1.8+32
output the temperature in farenheit
end
```

3)Executable C program :

```c
# include <stdio.h>
int main()
{
    float c,f;
    printf("Enter the temperture in centigrades: ");
    scanf("%f",&c);
    f=c*1.8+32;
    printf("the temperature in fahrenheit is %f ",f);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2>  gcc 3.c -o 3
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\3
Enter the temperture in centigrades: 34
the temperature in fahrenheit is 93.199997
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\3
Enter the temperture in centigrades: 700
the temperature in fahrenheit is 1292.000000
```

4))

1)Problem statement:

```
Write a C program to read in two integers and display one as
percentage of the other. Typically your output should look
like
20 is 50.00% of 40 assuming that the input numbers where 20
and
40. Display the percentage correct to 2 decimal places.
```

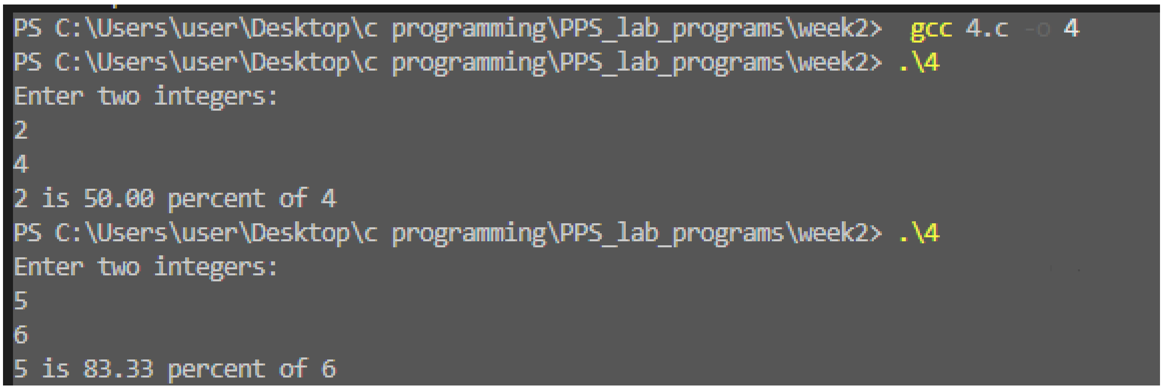
2)Psuedo code:

```
Start
Define a,b,Percent
Input a,b
Percent=a/b*100
Output a is percent of b
End
```

3)Executable C program :

```
# include <stdio.h>
int main()
{
    float a,b;
    float percent;
    printf("Enter two integers:\n");
    scanf("%f %f",&a,&b);
    percent=a/b*100;
    printf("%.0f is %.2f percent of %.0f",a,percent,b);
  return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2>  gcc 4.c -o 4
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\4
Enter two integers:
2
4
2 is 50.00 percent of 4
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week2> .\4
Enter two integers:
5
6
5 is 83.33 percent of 6
```

5)Observations:

Week 3

Tutorial 3: Branching and logical expressions:

Lab 3: Problems involving if-then-else structures

1))

1)Problem statement:

```
 Write a C program to find the maximum from given three nos.
```

2)Psuedo code:

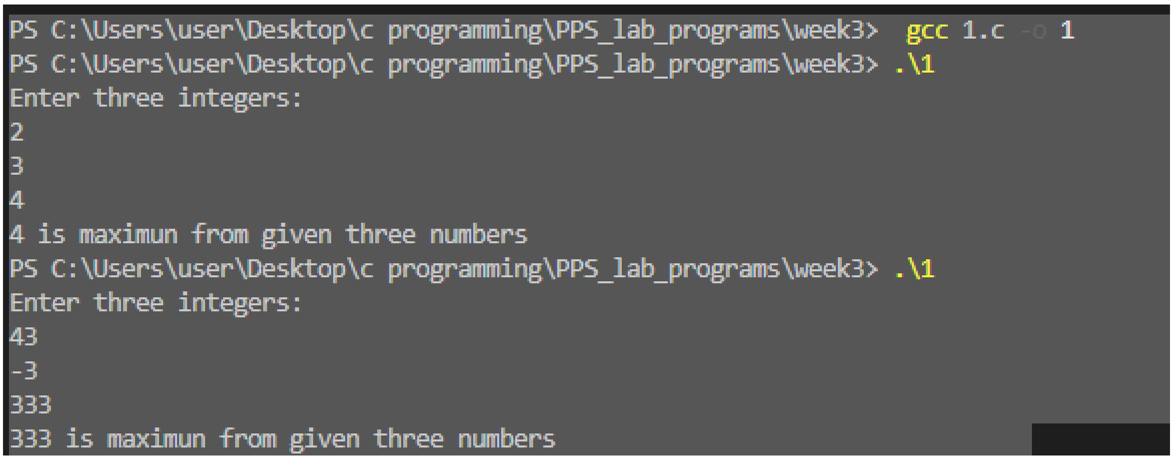
```
Start
Input a,b,c
If a>b && a>c
 Output a is maximum
```

```
Else if b>a && b>c
 Output b is maximum
Else
  Output c is maximum
End
```

3)Executable C program :

```c
# include <stdio.h>
int main()
{
    int a,b,c;
    printf("Enter three integers: \n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b && a>c)
       printf("%d is maximun from given three numbers",a);
    else if(b>a && b>c)
       printf("%d is maximun from given three numbers",b);
    else
       printf("%d is maximun from given three numbers",c);

   return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3>  gcc 1.c -o 1
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\1
Enter three integers:
2
3
4
4 is maximun from given three numbers
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\1
Enter three integers:
43
-3
333
333 is maximun from given three numbers
```

2))

1)Problem statement:

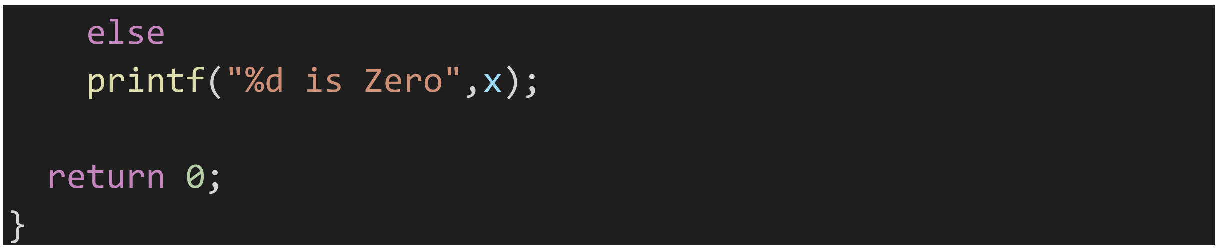
Write a C program to find that the accepted no is Negative, Positive or Zero.

2)Psuedo code:

```
Start
Input x
If x>0
  Output positive
Else if x>0
  Output negative
Else
  Output zero
End
```

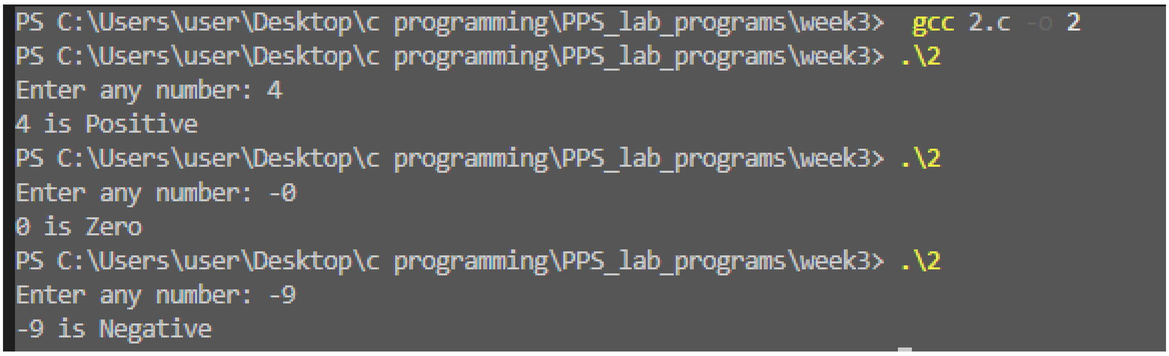
3)Executable C program :

```
# include <stdio.h>
int main()
{
    int x;
    printf("Enter any number: ");
    scanf("%d",&x);
    if(x>0)
     printf("%d is Positive",x);
    else if(x<0)
     printf("%d is Negative",x);
```



```
    else
    printf("%d is Zero",x);

  return 0;
}
```

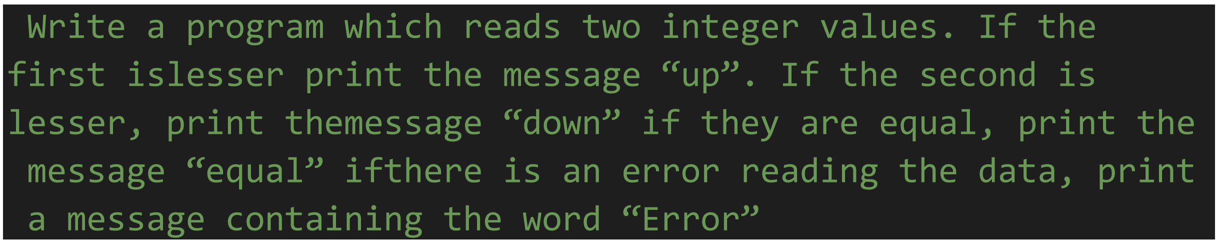
4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3>  gcc 2.c -o 2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\2
Enter any number: 4
4 is Positive
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\2
Enter any number: -0
0 is Zero
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\2
Enter any number: -9
-9 is Negative
```

3))

1)Problem statement:



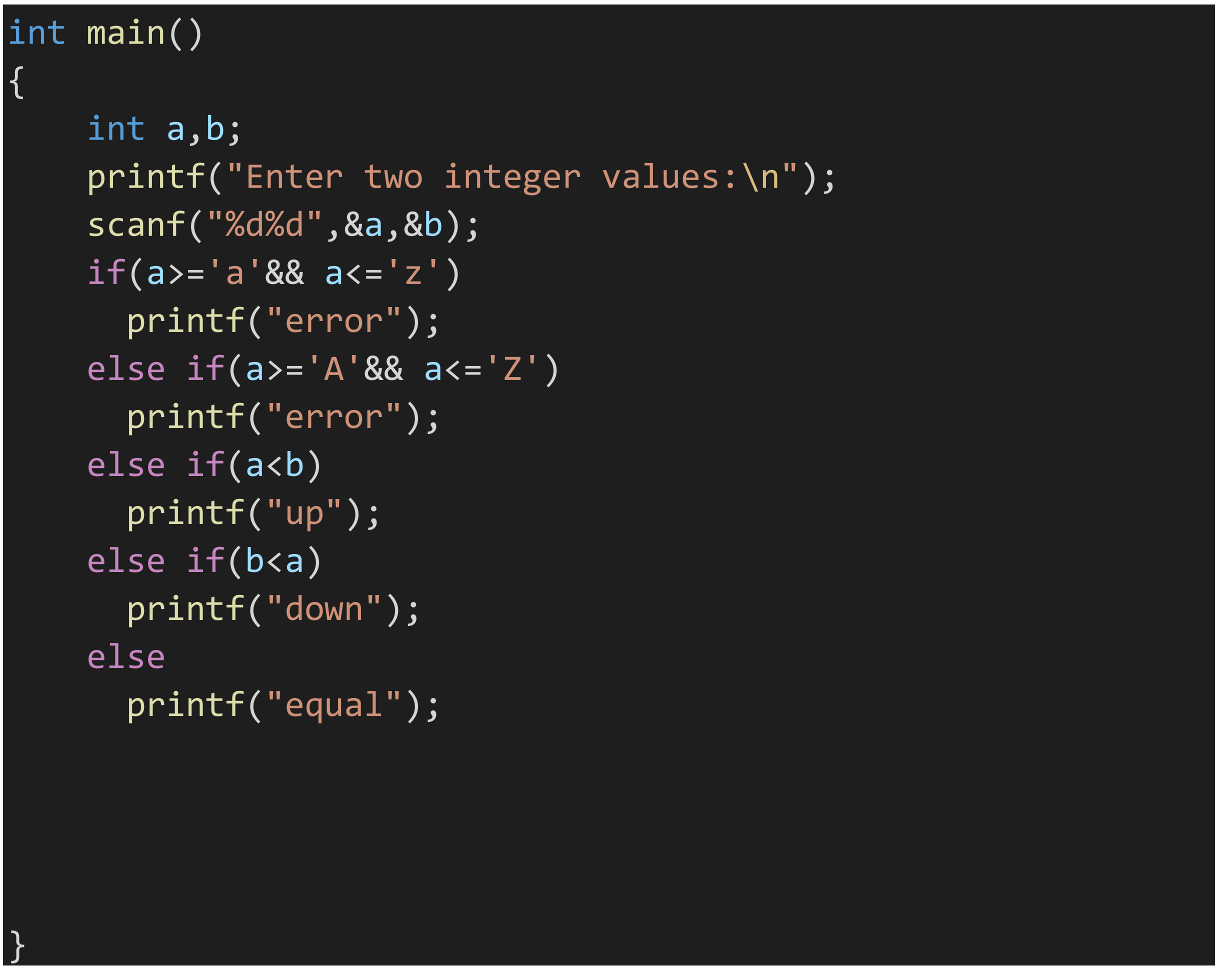
Write a program which reads two integer values. If the first islesser print the message “up”. If the second is lesser, print themessage “down” if they are equal, print the message “equal” ifthere is an error reading the data, print a message containing the word “Error”

2)Psuedo code:

```
Start
Input a,b
If a>=’a’ &&  a<=’z’
  Output error
Else if a>=’A’ && a<=’Z’
 Output error
Else if a<b
 Output up
```

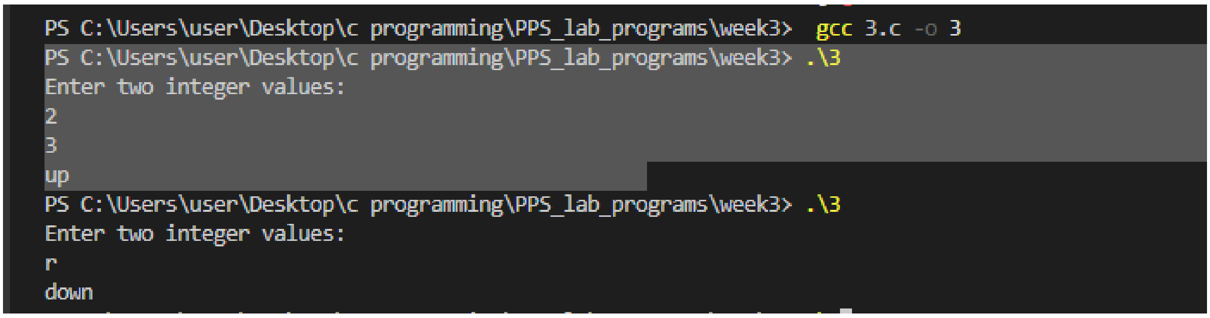
```
Else if b<a
  Output down
Else
Equal
End
```

3)Executable C program :



```c
int main()
{
    int a,b;
    printf("Enter two integer values:\n");
    scanf("%d%d",&a,&b);
    if(a>='a'&& a<='z')
      printf("error");
    else if(a>='A'&& a<='Z')
      printf("error");
    else if(a<b)
      printf("up");
    else if(b<a)
      printf("down");
    else
      printf("equal");






}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3>  gcc 3.c -o 3
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\3
Enter two integer values:
2
3
up
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\3
Enter two integer values:
r
down
```

4))

1)Problem statement:

Write a C program that prints the given three integers in Ascending order using if – else.

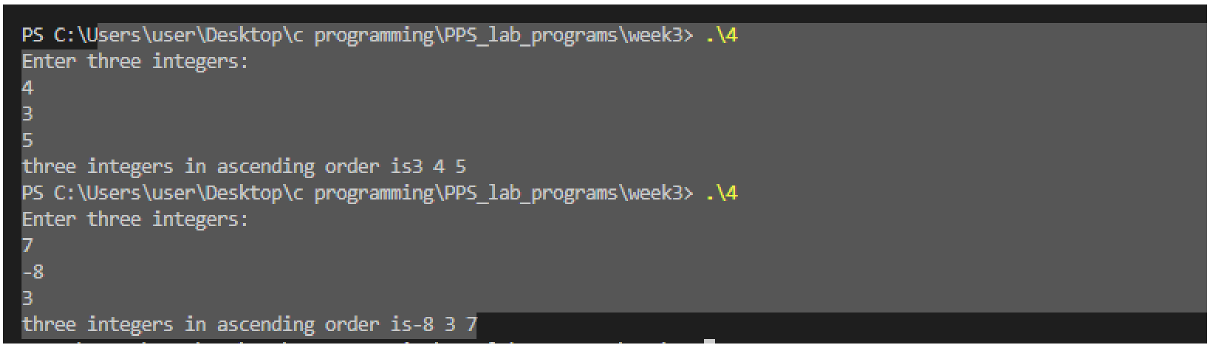
2)Psuedo code:

```
Start
Input a,b,c
If a>b && a>c
  If b>c
    Output  three integers in ascending order c,b,a
  Else
    Output three integers in ascending order  b,c,a
Else if b>a && b>c
  If a>c
    Output three integers in ascending order c,a,b
  Else
    Output three integers in ascending order a,c,b
Else if  c>a && c>b
    If a>b
       Output three integers in ascending order b,a,c
    Else
       Output three integers in ascending order a,b,c
End
```

3)Executable C program :

```c
# include <stdio.h>
int main()
{
    int a,b,c;
    printf("Enter three integers:\n");
    scanf("%d%d%d",&a,&b,&c);
    if(a>b && a>c)
    {   if(b>c)
        printf("three integers in ascending order is %d %d %
d",c,b,a);
        else
        printf("three integers in ascending order is %d %d %
d",b,c,a);

    }
    else if(b>a && b>c)
    {  if(a>c)
       printf("three integers in ascending order is %d %d %d
",c,a,b);
       else
       printf("three integers in ascending order is %d %d %d
",a,c,b);
    }
    else if(c>a && c>b)
    {   if(a>b)
       printf("three integers in ascending order is %d %d %d
",b,a,c);
       else
       printf("three integers in ascending order is %d %d %d
",a,b,c);
    }

}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\4
Enter three integers:
4
3
5
three integers in ascending order is3 4 5
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\4
Enter three integers:
7
-8
3
three integers in ascending order is-8 3 7
```

5))

1)Problem statement:

```
Given as input three integers representing a date as day,
month,year, print the number day, month and year for the next day's date.
Typical input: “28 2 1992” Typical output: “Date following
28:02:1992 is 29:02:1992
```

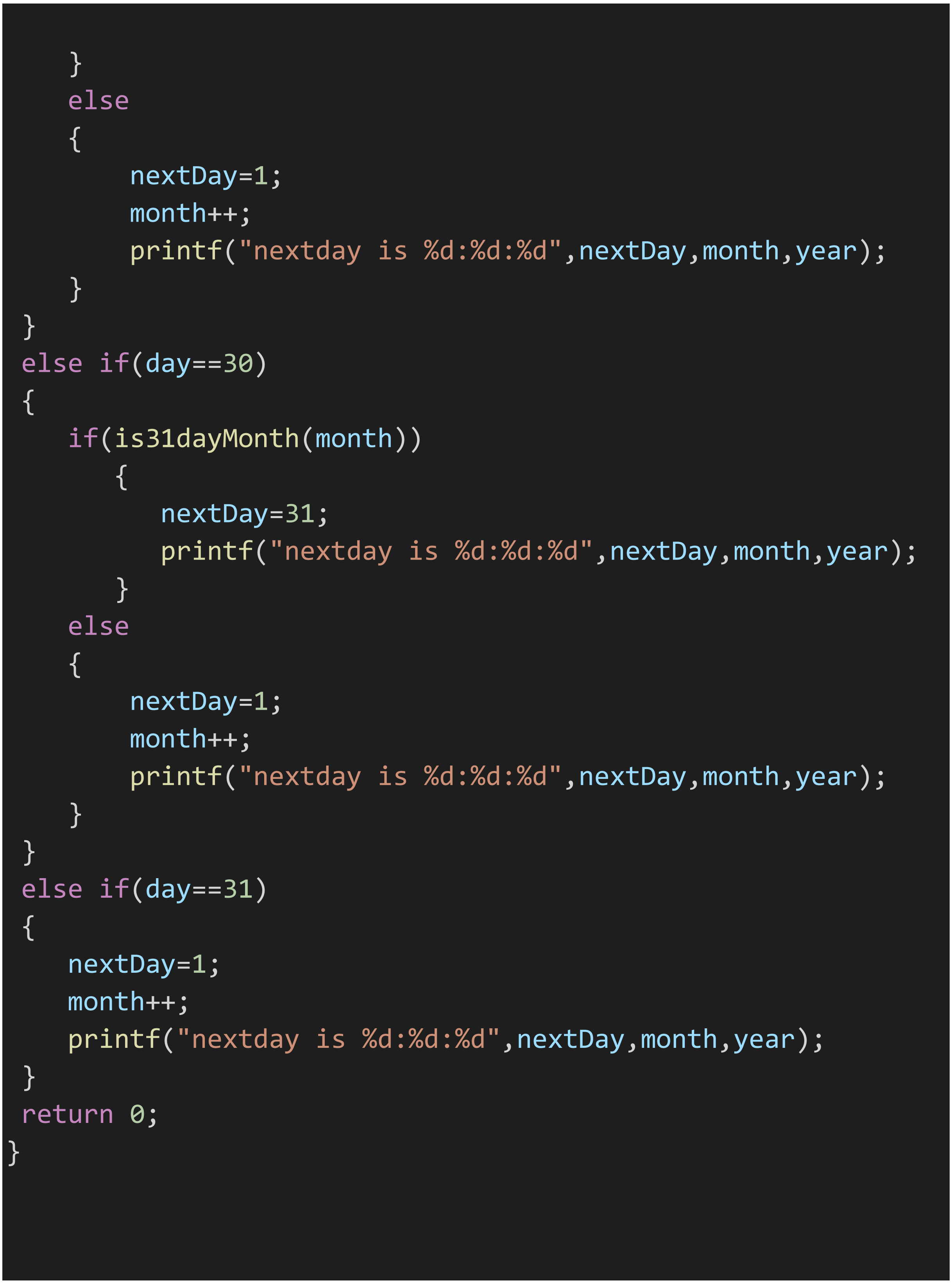
2)Psuedo code:

```
Start
Declare enum jan=1,feb,mar,apr,may,Jun,jul,aug,sep,oct,nov,dec
Declare _Bool is 31dayMonth()
                    If    m==jan or m==mar or m==may or m==jul or m==aug or m==oct
                          or m==dec
                         return 1
                    else
                         return 0
Declare day,month,year,nextDay
Input day,month,year
If month==13
    Month=1
    Year++
     Nextday=1
     Output nextday is nextday:month:year
Else if  month==12
     Month=1
      Nextday=1
      Year++
       Output nextday is nextday:month:year
Else if day==28 and month=feb
     If year%4==0
         Nextday=29
         Output nextday is nextday:month:year
    Else
       Nextday=1
       Month++
       Output nextday is nextday:month:year
Else if day==30
    If (is31dayMonth(month))
         nextDay=31
         Output nextday is nextday:month:year
     Else
         Nextday=1
         Month++
         Output nextday is nextday:month:year
Else if day==31
  Nextday=1
  Month++
  Output nextday is nextday:month:year

End
```

3)Executable C program :

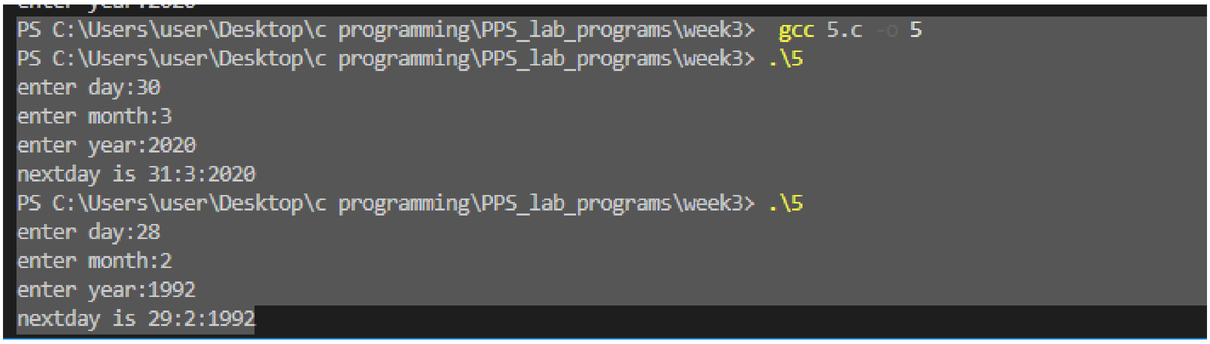
```
# include <stdio.h>

enum { Jan=1, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct,
Nov, Dec};

_Bool is31dayMonth (int m)
{
    if ((m==Jan) || (m==Mar)|| (m==May)|| (m==Jul)||
(m==Aug)|| (m==Oct)|| (m==Dec))
    {
      return 1;
    }

    else
      return 0;
}

int main()
```

```
{
   //enum month{ Jan=1, Feb=2, Mar=3, Apr=4, May=5, Jun=6,
Jul=7, Aug=8, Sep=9, Oct=10, Nov=11, Dec=12};

    int day;
    int month;
    int year ;

   printf("enter day:");
    scanf("%d",&day);

    printf("enter month:");
    scanf("%d",&month);

    printf("enter year:");
    scanf("%d",&year);
   int nextDay;

 if(month==13)
 {
    month=1;
    year++;
    nextDay=1;
    printf("nextday is %d:%d:%d",nextDay,month,year);
 }
 else if (month==12)
 {
    month=1;
    nextDay=1;
    year++;
    printf("nextday is %d:%d:%d",nextDay,month,year);
 }
 else if(day==28 && month==Feb)
 {
    if(year%4==0)
    {
        nextDay=29;
        printf("nextday is %d:%d:%d",nextDay,month,year);
```



```
    }
    else
    {
        nextDay=1;
        month++;
        printf("nextday is %d:%d:%d",nextDay,month,year);
    }
 }
 else if(day==30)
 {
    if(is31dayMonth(month))
       {
          nextDay=31;
          printf("nextday is %d:%d:%d",nextDay,month,year);
       }
    else
    {
        nextDay=1;
        month++;
        printf("nextday is %d:%d:%d",nextDay,month,year);
    }
 }
 else if(day==31)
 {
    nextDay=1;
    month++;
    printf("nextday is %d:%d:%d",nextDay,month,year);
 }
 return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3>  gcc 5.c -o 5
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\5
enter day:30
enter month:3
enter year:2020
nextday is 31:3:2020
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week3> .\5
enter day:28
enter month:2
enter year:1992
nextday is 29:2:1992
```

Week 4 & 5

Tutorial 4: Loops, while and for loops:

Lab 4: Iterative problems e.g., sum of series

1))

1)Problem statement:

```
1. Write a C program to find the sum of first 100 odd nos.
and even nos.
```

2)Psuedo code:

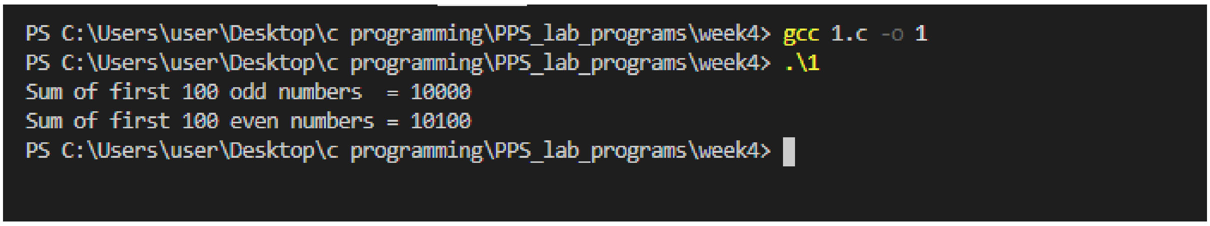
```
Start
Declare i,odd_sum,even_sum,count
Start for loop
 For i=1;i<200;i++
    If  i%2==0
       even_sum=even__sum+i
       count++
    else
       odd_sum=odd_sum+i
        count++
output
  sum of first 100 odd numbers=odd_sum
```

  sum of first 100 even numbers=even_sum
End

3)Executable C program :

```
# include <stdio.h>
 void main()
{
    int i, odd_sum = 0, even_sum = 0,count=0;
        for(i=1;i<=200;i++)
         if(i%2==0)
         {
             even_sum=even_sum+i;
             count++;
             //printf("count is %d",count);
         }
         else
         { odd_sum=odd_sum+i;
             count++;
            //printf("count is %d",count);
         }
    printf("Sum of first 100 odd numbers  = %d\n", odd_sum);
    printf("Sum of first 100 even numbers = %d\n", even_sum)
;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 1.c -o 1
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\1
Sum of first 100 odd numbers  = 10000
Sum of first 100 even numbers = 10100
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> 
```

2))

1)Problem statement:

```
2. Write a C program to display first 100 prime nos
```

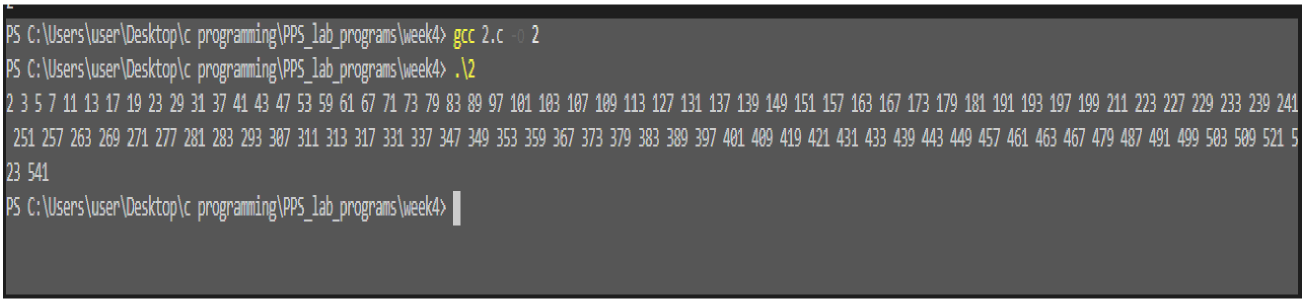
2)Psuedo code:

```
Start
Declare ct,n,i,j
 While n<100
    j=1
   ct=0
   while j<=i
      if  i%j==0
      ct++
      j++
  if ct==2
  output i
  n++
i++
End
```

3)Executable C program :

```c
#include <stdio.h>
int main()
{
    int ct=0,n=0,i=1,j=1;
    while(n<100)
    {
        j=1;
        ct=0;
        while(j<=i)
        {
            if(i%j==0)
            ct++;
            j++;
        }
        if(ct==2)
        {
            printf("%d ",i);
            n++;
        }
        i++;
}
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 2.c -o 2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\2
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167 173 179 181 191 193 197 199 211 223 227 229 233 239 241
 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 439 443 449 457 461 463 467 479 487 491 499 503 509 521 5
23 541
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4>
```

---

3))

1)Problem statement:

```
3. Write a C program to read in a three digit number produce
following output
(Assuming that the input is 347) 3 hundreds, 4 tens, 7 units
```

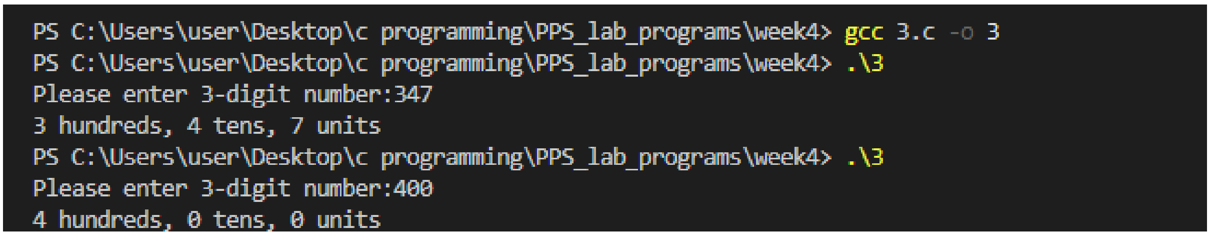
2)Psuedo code:

```
Start
Declare number ,hun,ten,unit
Input number
hun=number/100
number=number%100
ten=number/10
unit=number%10
output
  hun,ten,unit
End
```

3)Executable C program :

```c
# include <stdio.h>
int main()
{
    int number,hun,ten,unit;
    printf("Please enter 3-digit number:");
    scanf("%d",&number);
    hun=number/100;
    number=number%100;
    ten=number/10;
    unit=number%10;
    printf("%d hundreds, %d tens, %d units",hun,ten,unit);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 3.c -o 3
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\3
Please enter 3-digit number:347
3 hundreds, 4 tens, 7 units
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\3
Please enter 3-digit number:400
4 hundreds, 0 tens, 0 units
```

4))

1)Problem statement:

```
Write a C program to display Fibonacci series
```

2)Psuedo code:

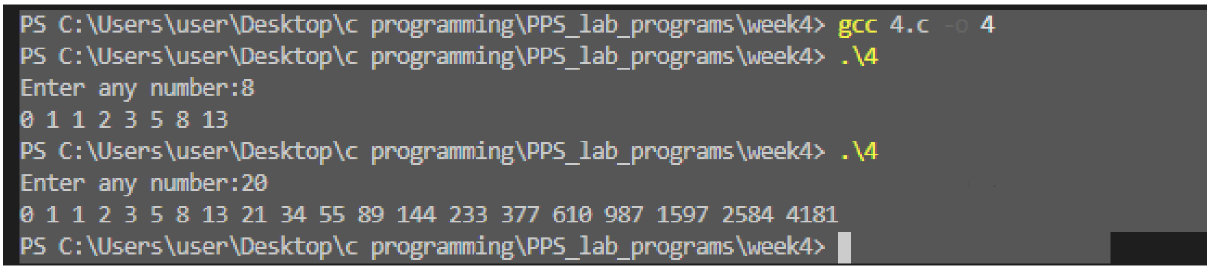
```
Start
Declare n,a=0,b=1,c,i
Input n
Start for loop
For i=1;i<=n;i++
 Output a
 c=a+b
 a=b
b=c
```

End

3)Executable C program :

```
#include<stdio.h>
int main()
{
int n,a=0,b=1,c,i;
printf("Enter any number:");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
printf("%d ",a);
c=a+b;
a=b;
b=c;
}
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 4.c -o 4
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\4
Enter any number:8
0 1 1 2 3 5 8 13
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\4
Enter any number:20
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4>
```

5i))

1)Problem statement:

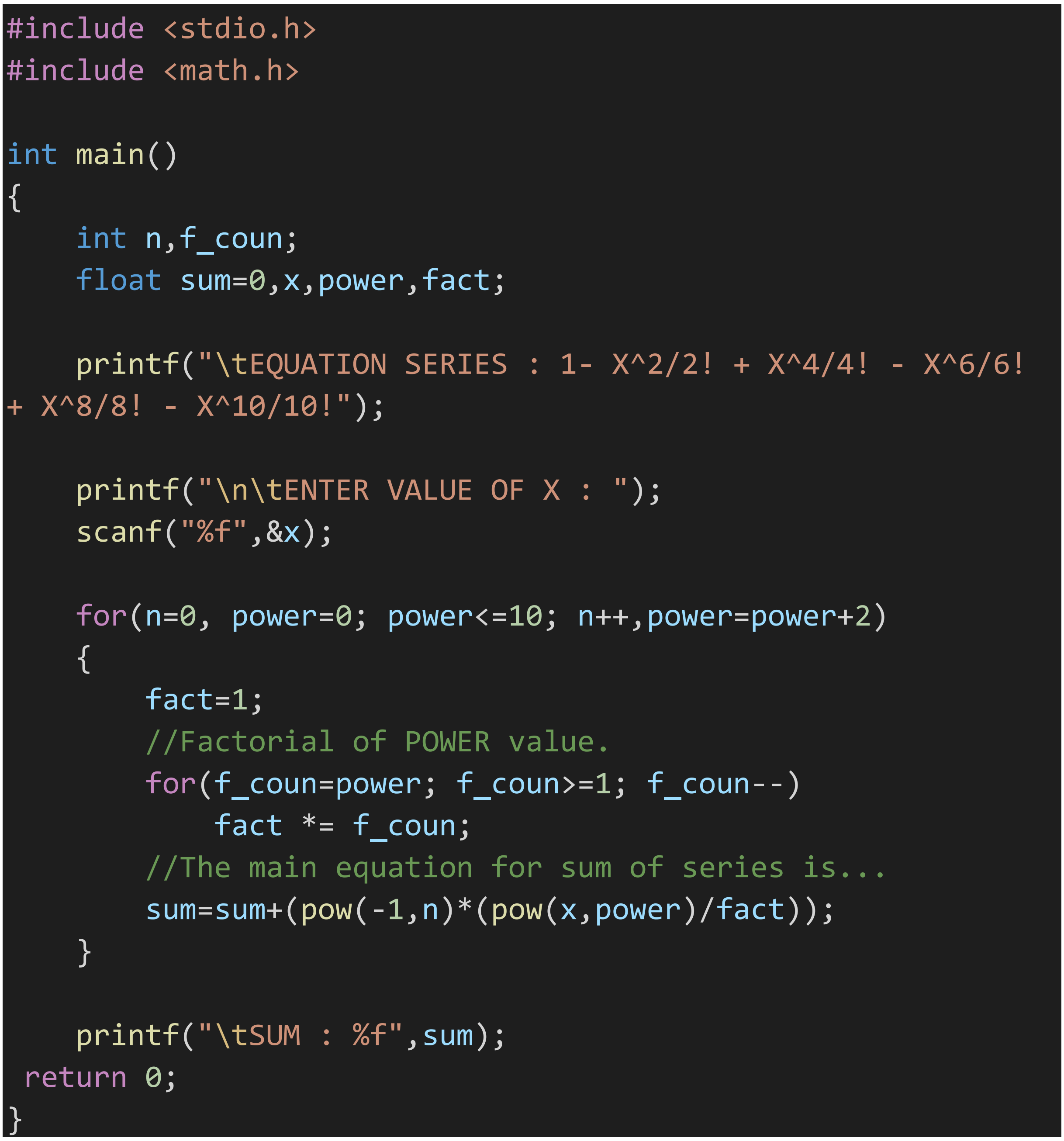
```
5. Write a C program to calculate the following
i.sum=1-x2/2! +x4/4!-x6/6!+x8/8!-x10/10!+………,
```

2)Psuedo code:

```
Start
Declare n,f_coun
Declare sum,power,fact
For n=0,power=0; power<=10;n++,power=power+2
```

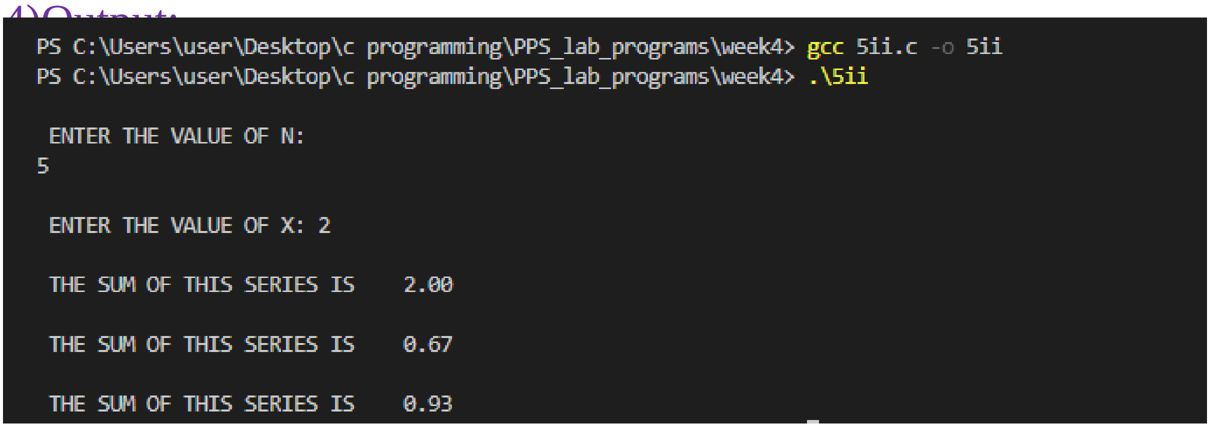
```
  Fact=1
  For f_coun=power;f_coun>=1;f_coun- -
    Fact*=f_coun
 Sum=sum+(pow(-1,n)*(pow(x,power)/fact))

Output
SUM=sum
End
```

3)Executable C program :



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

int main()
{
    int n,f_coun;
    float sum=0,x,power,fact;

    printf("\tEQUATION SERIES : 1- X^2/2! + X^4/4! - X^6/6!
+ X^8/8! - X^10/10!");

    printf("\n\tENTER VALUE OF X : ");
    scanf("%f",&x);

    for(n=0, power=0; power<=10; n++,power=power+2)
    {
        fact=1;
        //Factorial of POWER value.
        for(f_coun=power; f_coun>=1; f_coun--)
            fact *= f_coun;
        //The main equation for sum of series is...
        sum=sum+(pow(-1,n)*(pow(x,power)/fact));
    }

    printf("\tSUM : %f",sum);
 return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 5ii.c -o 5ii
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\5ii

 ENTER THE VALUE OF N:
5

 ENTER THE VALUE OF X: 2

 THE SUM OF THIS SERIES IS    2.00

 THE SUM OF THIS SERIES IS    0.67

 THE SUM OF THIS SERIES IS    0.93
```

5ii))

1)Problem statement:

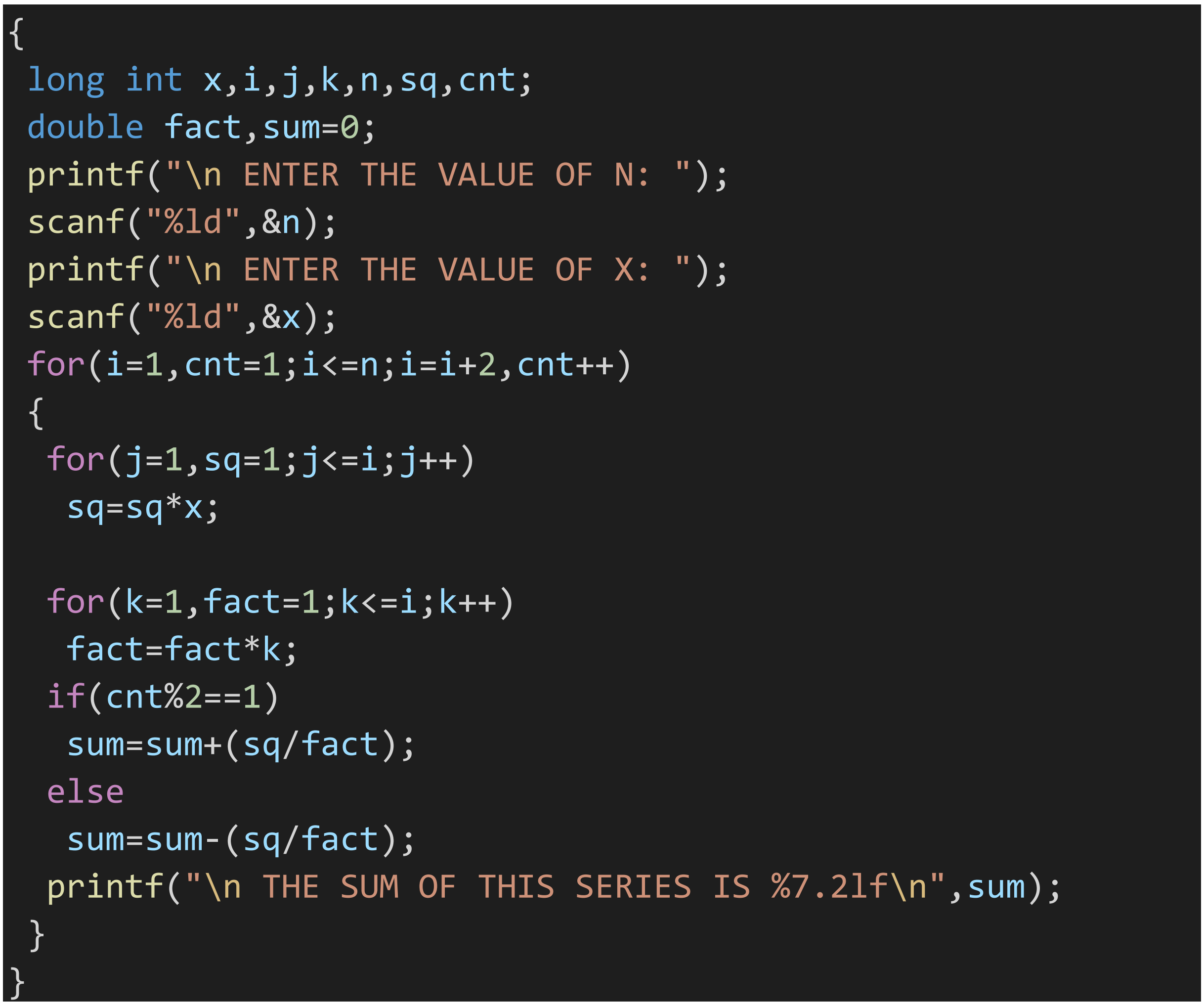
```
ii.sum=x-x3/3!+x5/5!.........................,
```

2)Psuedo code:

```
Start
Declare x,I,j,k,n,sq,cnt
Declare fact,sum=0
Input n,x
For i=1,cnt=1;i<=n;i=i+2,cnt++
   For j=1,sq=1;j<=i;j++
       sq=sq*x
   For k=1,fact=1;k<=I;k++
       fact=fact*k
   if cnt%2==1
        sum=sum+(sq/fact)
   else
        sum=sum-(sq/fact)
output
"sum of the series is" ,sum

End
```

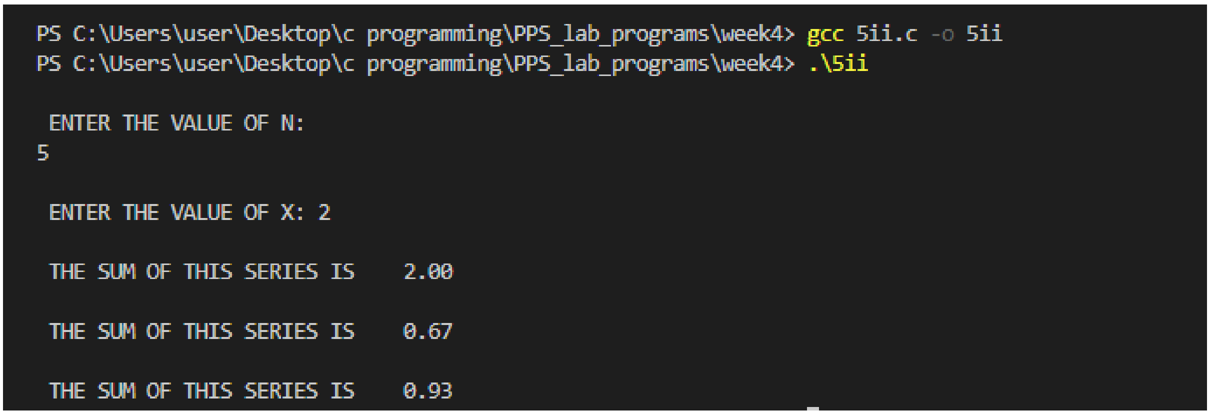
3)Executable C program :

```
#include<stdio.h>

int main()
```



```c
{
 long int x,i,j,k,n,sq,cnt;
 double fact,sum=0;
 printf("\n ENTER THE VALUE OF N: ");
 scanf("%ld",&n);
 printf("\n ENTER THE VALUE OF X: ");
 scanf("%ld",&x);
 for(i=1,cnt=1;i<=n;i=i+2,cnt++)
 {
  for(j=1,sq=1;j<=i;j++)
   sq=sq*x;

  for(k=1,fact=1;k<=i;k++)
   fact=fact*k;
  if(cnt%2==1)
   sum=sum+(sq/fact);
  else
   sum=sum-(sq/fact);
  printf("\n THE SUM OF THIS SERIES IS %7.2lf\n",sum);
 }
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 5ii.c -o 5ii
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\5ii

 ENTER THE VALUE OF N:
5

 ENTER THE VALUE OF X: 2

 THE SUM OF THIS SERIES IS    2.00

 THE SUM OF THIS SERIES IS    0.67

 THE SUM OF THIS SERIES IS    0.93
```

5iii))

1)Problem statement:

```
iii.sum=1+x/1!+x^2/2!+x^3/3!..............,
```

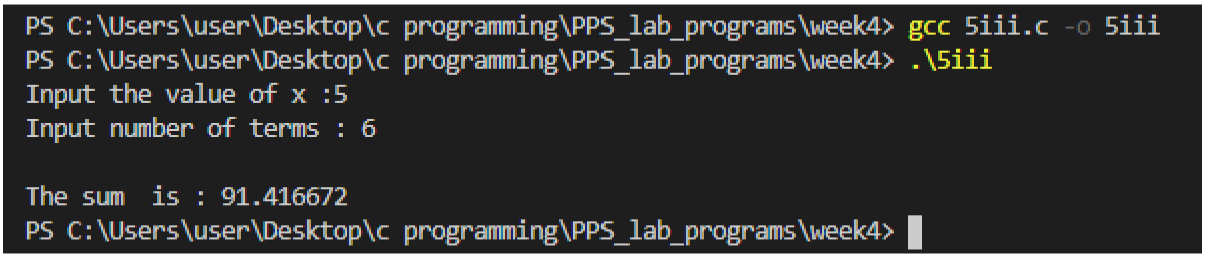
2)Psuedo code:

```
Start
Declare x,sum,no_row,i,n
Input x,n
Initalise sum=1;no_row=1
For i=1;i<n;i++
  No_row=no_row*x/(float)i
  Sum=sum+no_row
Output
 “the sum is”,sum
End
```

3)Executable C program :

```c
#include <stdio.h>

void main()
{
    float x,sum,no_row;
    int i,n;
    printf("Input the value of x :");
    scanf("%f",&x);
    printf("Input number of terms : ");
    scanf("%d",&n);
    sum =1; no_row = 1;
    for (i=1;i<n;i++)
    {
      no_row = no_row*x/(float)i;
      sum =sum+ no_row;
    }
    printf("\nThe sum  is : %f\n",sum);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 5iii.c -o 5iii
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\5iii
Input the value of x :5
Input number of terms : 6

The sum  is : 91.416672
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> 
```

6))

1)Problem statement:

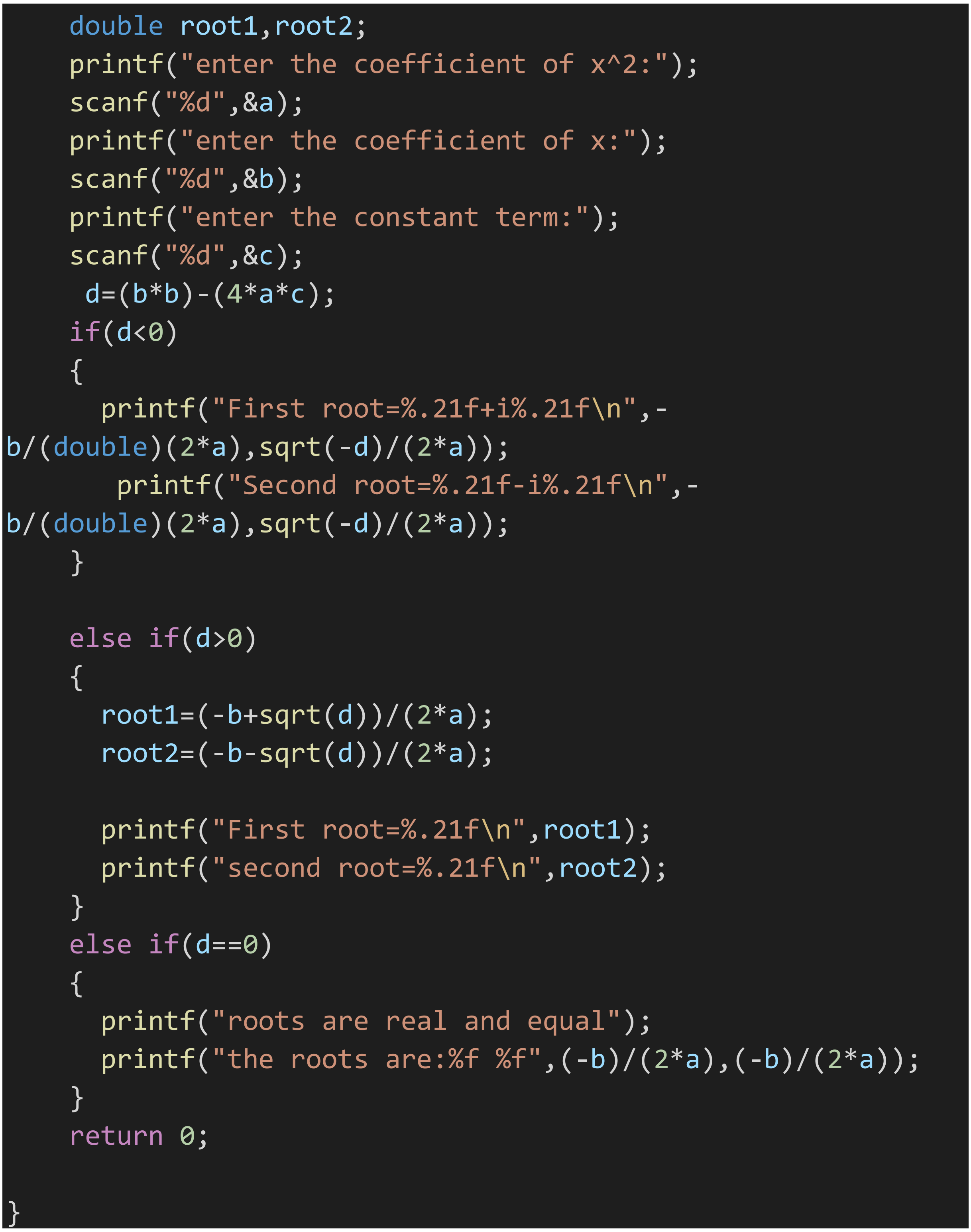
```
6. Write a C program to find the roots of a Quadratic
equation.
```

2)Psuedo code:

```
Start
Declare a,b,c,d,root1,root2
Input a,b,c
d=(b*b)-(4*a*c)
if(d<0)
  output root1
  output root2
else if d>0
  output root1=(-b+sqrt(d))/(2*a)
  output root2=(-b-sqrt(d))/(2*a)
else if d==0
  output roots are real and equal
  output the roots are (-b)/(2*a),(-b)/(2*a)
End
```

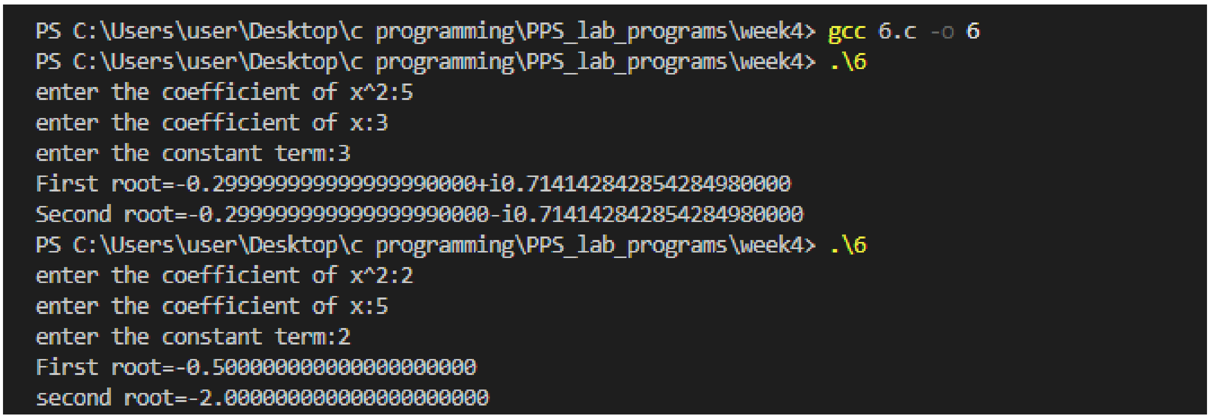
3)Executable C program :

```c
# include <stdio.h>
# include <math.h>
int main()
{
    int a,b,c,d;
```



```
    double root1,root2;
    printf("enter the coefficient of x^2:");
    scanf("%d",&a);
    printf("enter the coefficient of x:");
    scanf("%d",&b);
    printf("enter the constant term:");
    scanf("%d",&c);
     d=(b*b)-(4*a*c);
    if(d<0)
    {
      printf("First root=%.21f+i%.21f\n",-
b/(double)(2*a),sqrt(-d)/(2*a));
       printf("Second root=%.21f-i%.21f\n",-
b/(double)(2*a),sqrt(-d)/(2*a));
    }

    else if(d>0)
    {
      root1=(-b+sqrt(d))/(2*a);
      root2=(-b-sqrt(d))/(2*a);

      printf("First root=%.21f\n",root1);
      printf("second root=%.21f\n",root2);
    }
    else if(d==0)
    {
      printf("roots are real and equal");
      printf("the roots are:%f %f",(-b)/(2*a),(-b)/(2*a));
    }
    return 0;

}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> gcc 6.c -o 6
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\6
enter the coefficient of x^2:5
enter the coefficient of x:3
enter the constant term:3
First root=-0.29999999999999990000+i0.71414284285428498000
Second root=-0.29999999999999990000-i0.71414284285428498000
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week4> .\6
enter the coefficient of x^2:2
enter the coefficient of x:5
enter the constant term:2
First root=-0.50000000000000000000
second root=-2.00000000000000000000
```

Week 6

Tutorial 5: 1D Arrays: searching, sorting:

Lab 5: 1D Array manipulation

1))

1)Problem statement:

```
C program that reads N integer numbers and arrange them in
ascending order using Bubble Sort
```

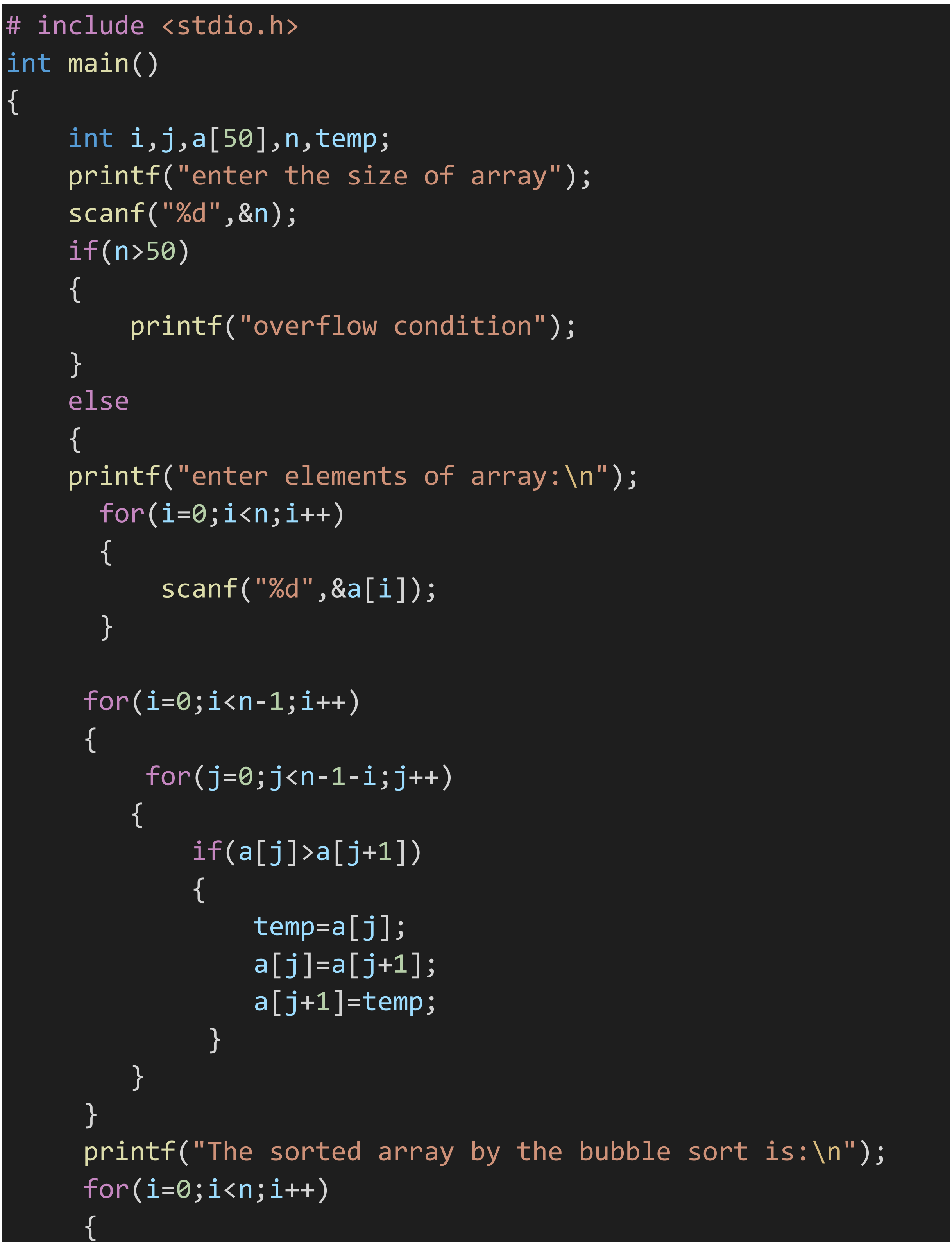
2)Psuedo code:

```
Start
Begin Bubble sort(list)
For all elements of list
If list[i]>list[i+1]
Swap(list[i],list[i+1])
End if
End for
Return list
```

End Bubble sort

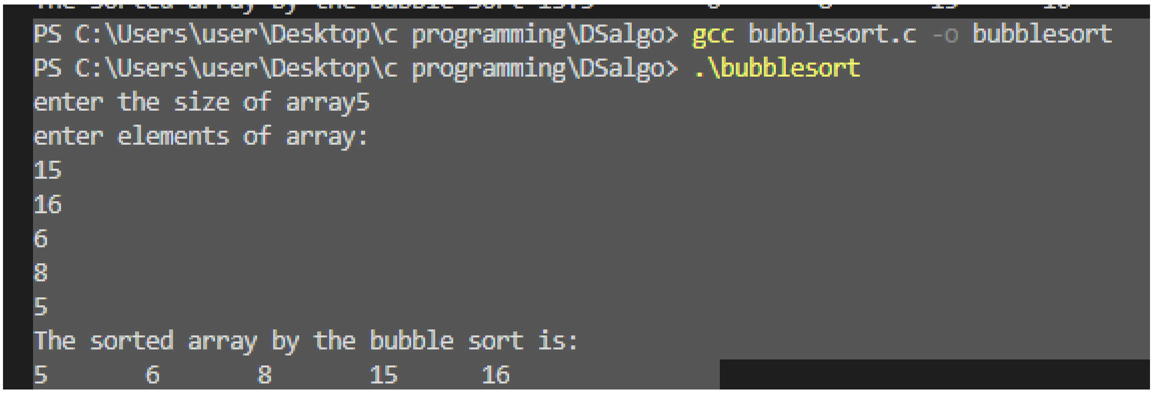
End

3)Executable C program :



```
# include <stdio.h>
int main()
{
    int i,j,a[50],n,temp;
    printf("enter the size of array");
    scanf("%d",&n);
    if(n>50)
    {
        printf("overflow condition");
    }
    else
    {
    printf("enter elements of array:\n");
      for(i=0;i<n;i++)
      {
          scanf("%d",&a[i]);
      }

     for(i=0;i<n-1;i++)
     {
         for(j=0;j<n-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
             }
        }
     }
     printf("The sorted array by the bubble sort is:\n");
     for(i=0;i<n;i++)
     {
         printf("%d\t",a[i]);
     }
    }
  return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\DSalgo> gcc bubblesort.c -o bubblesort
PS C:\Users\user\Desktop\c programming\DSalgo> .\bubblesort
enter the size of array5
enter elements of array:
15
16
6
8
5
The sorted array by the bubble sort is:
5       6       8       15      16
```

5)Observations:

- **The main condition is if(a[j]>a[j+1]) to check in the bubble sort**

2))

1)Problem statement:

```
 C program that reads N integer numbers and arrange them in
ascending order using Merge Sort
```

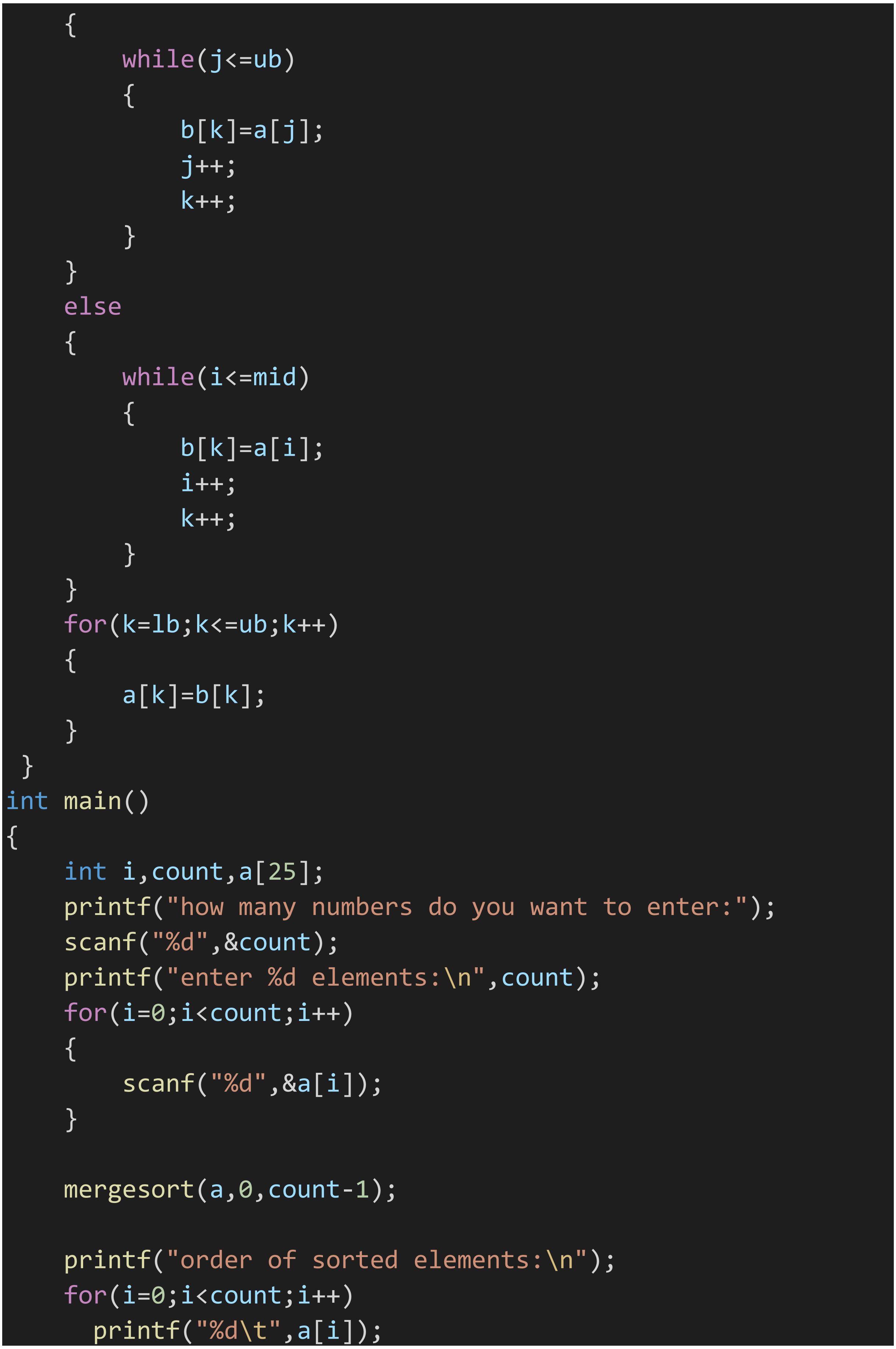
2)Psuedo code:

```
Start
MergeSort(arr[],left,right)
If left>right
```

```
return mid=(left+right)/2
MergeSort(arr,left,mid)
MergeSort(arr,mid,right)
End
```

3)Executable C program :

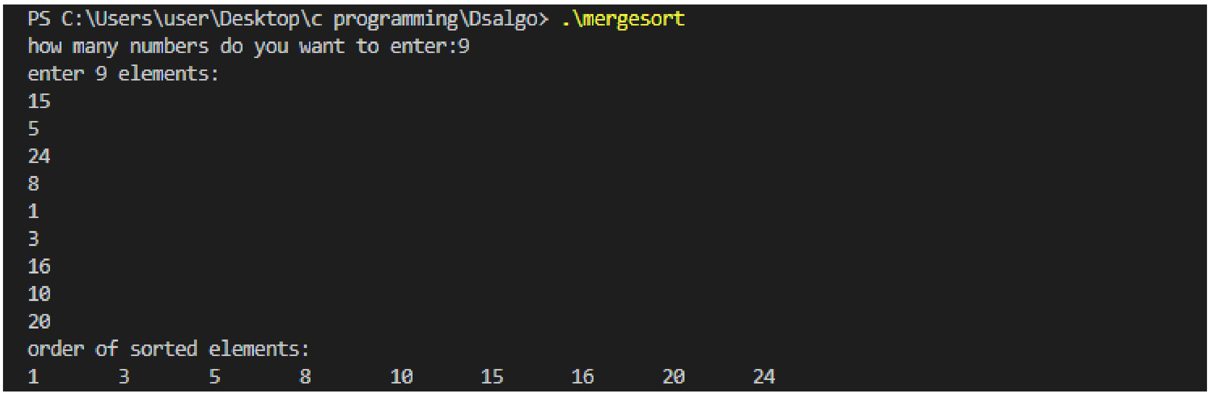
```
# include <stdio.h>
int merge(int[],int,int,int);
int mergesort(int[],int,int);
int mergesort(int a[25],int lb,int ub)
{   int mid;
    if(lb<ub)
    {
        mid=(lb+ub)/2;
        mergesort(a,lb,mid);
        mergesort(a,mid+1,ub);
        merge(a,lb,mid,ub);
    }
}
int merge(int a[25],int lb,int mid,int ub)
 {  int i,j,k,b[25];
    i=lb;
    j=mid+1;
    k=lb;
    while(i<=mid && j<=ub)
    {
      if(a[i]<=a[j])
      {
          b[k]=a[i];
          i++;
      }
      else
      {
          b[k]=a[j];
          j++;
      }
      k++;
    }
    if(i>mid)
```



```
    {
        while(j<=ub)
        {
            b[k]=a[j];
            j++;
            k++;
        }
    }
    else
    {
        while(i<=mid)
        {
            b[k]=a[i];
            i++;
            k++;
        }
    }
    for(k=lb;k<=ub;k++)
    {
        a[k]=b[k];
    }
 }
int main()
{
    int i,count,a[25];
    printf("how many numbers do you want to enter:");
    scanf("%d",&count);
    printf("enter %d elements:\n",count);
    for(i=0;i<count;i++)
    {
        scanf("%d",&a[i]);
    }

    mergesort(a,0,count-1);

    printf("order of sorted elements:\n");
    for(i=0;i<count;i++)
      printf("%d\t",a[i]);
```

```
    return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\Dsalgo> .\mergesort
how many numbers do you want to enter:9
enter 9 elements:
15
5
24
8
1
3
16
10
20
order of sorted elements:
1       3       5       8       10      15      16      20      24
```

5)Observations:

Complete array is divided into n sub arrays.

- **Each subarray is having one element.**

- **We keep on dividing the array into subarray, until we get the subarray containing only one element.**

- **After that, we keep on merging the subarrays to produce a new sorted array**.

3))

1)Problem statement:

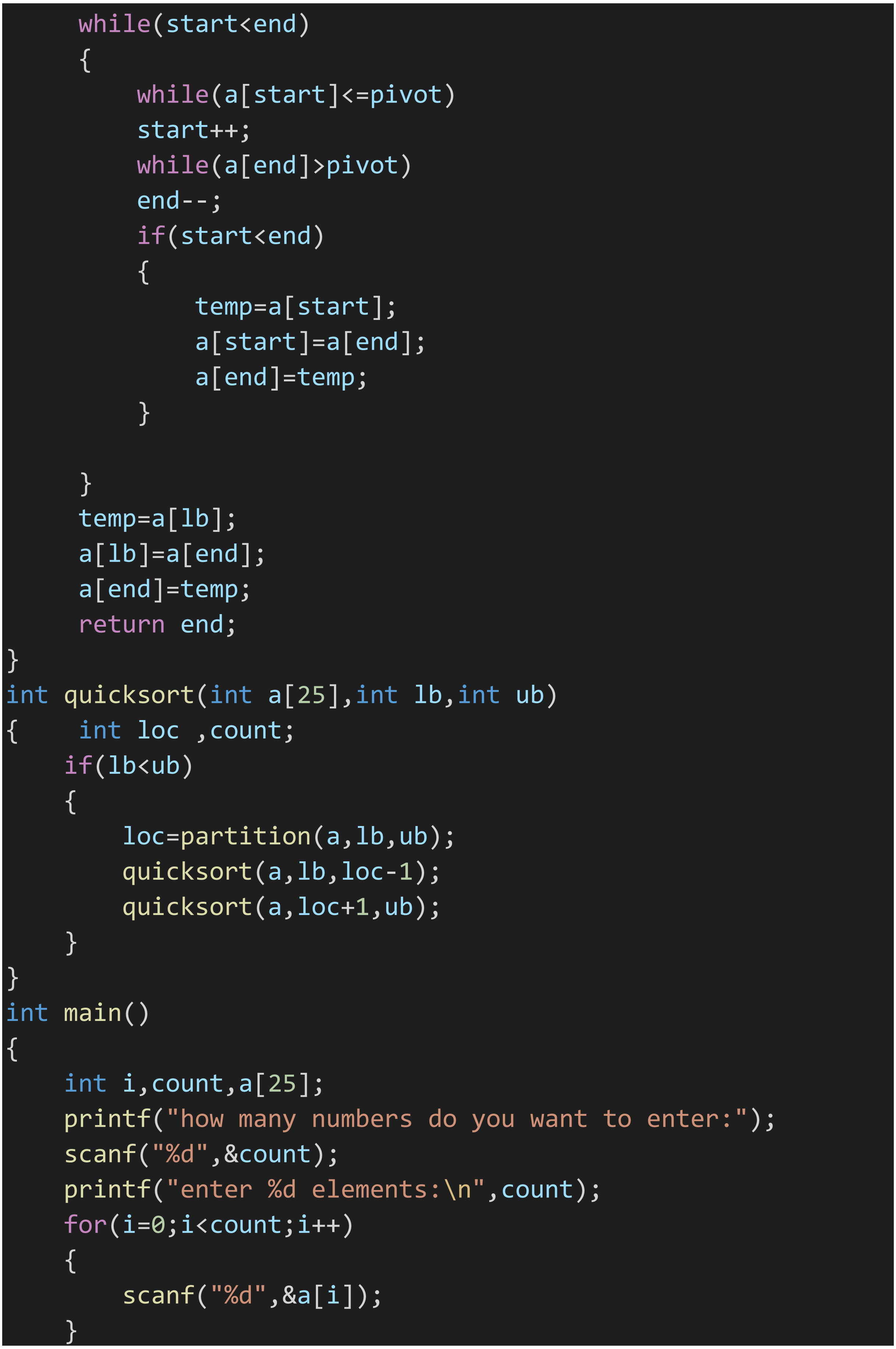
```
C program that reads N integer numbers and arrange them in
ascending order using Quick Sort
```

2)Psuedo code:

```
Quick sort( arr ,beg ,end)
If(beg<end)
Pivot index=position(arr,beg,end)
Quick sort(arr ,beg , pivotindex)
Quick sort(arr,pivotindex+1,end)
Partition(arr ,beg,end)
Set end as pivot
Pindex=beg-1
For i=beg to end-1
 If arr[i]<pivot
 Swap arr[i] and arr[pindex]
 Pindex++
 Swap pivot and arr[pindex+1]
Return pindex+1
```

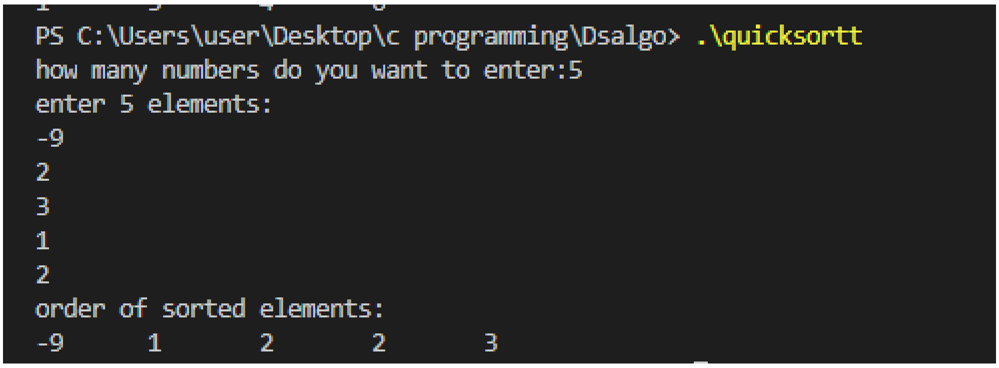
3)Executable C program :

```
# include <stdio.h>
int partition(int [],int ,int);
int quicksort(int[],int, int);
int  partition(int a[25],int lb,int ub)
{   int pivot,start,end,temp;
     pivot=a[lb];
     start=lb;
     end=ub;
```



```
     while(start<end)
     {
         while(a[start]<=pivot)
         start++;
         while(a[end]>pivot)
         end--;
         if(start<end)
         {
             temp=a[start];
             a[start]=a[end];
             a[end]=temp;
         }

     }
     temp=a[lb];
     a[lb]=a[end];
     a[end]=temp;
     return end;
}
int quicksort(int a[25],int lb,int ub)
{    int loc ,count;
    if(lb<ub)
    {
        loc=partition(a,lb,ub);
        quicksort(a,lb,loc-1);
        quicksort(a,loc+1,ub);
    }
}
int main()
{
    int i,count,a[25];
    printf("how many numbers do you want to enter:");
    scanf("%d",&count);
    printf("enter %d elements:\n",count);
    for(i=0;i<count;i++)
    {
        scanf("%d",&a[i]);
    }
    quicksort(a,0,count-1);

    printf("order of sorted elements:\n");
    for(i=0;i<count;i++)
      printf("%d\t",a[i]);
    return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\Dsalgo> .\quicksortt
how many numbers do you want to enter:5
enter 5 elements:
-9
2
3
1
2
order of sorted elements:
-9      1       2       2       3
```

5)Observations:

- **We are going to choose one pivot element.**

- **Pivot element can be anything.**

- **Partition the array in such a way that all the elementspivot would be to the right side of the pivot.**

- **Passing the arguments into the function should be taken care of..**

Week 7

Tutorial 6: 2D arrays and Strings

Lab 6: Matrix problems, String operations

1i))

1)Problem statement:

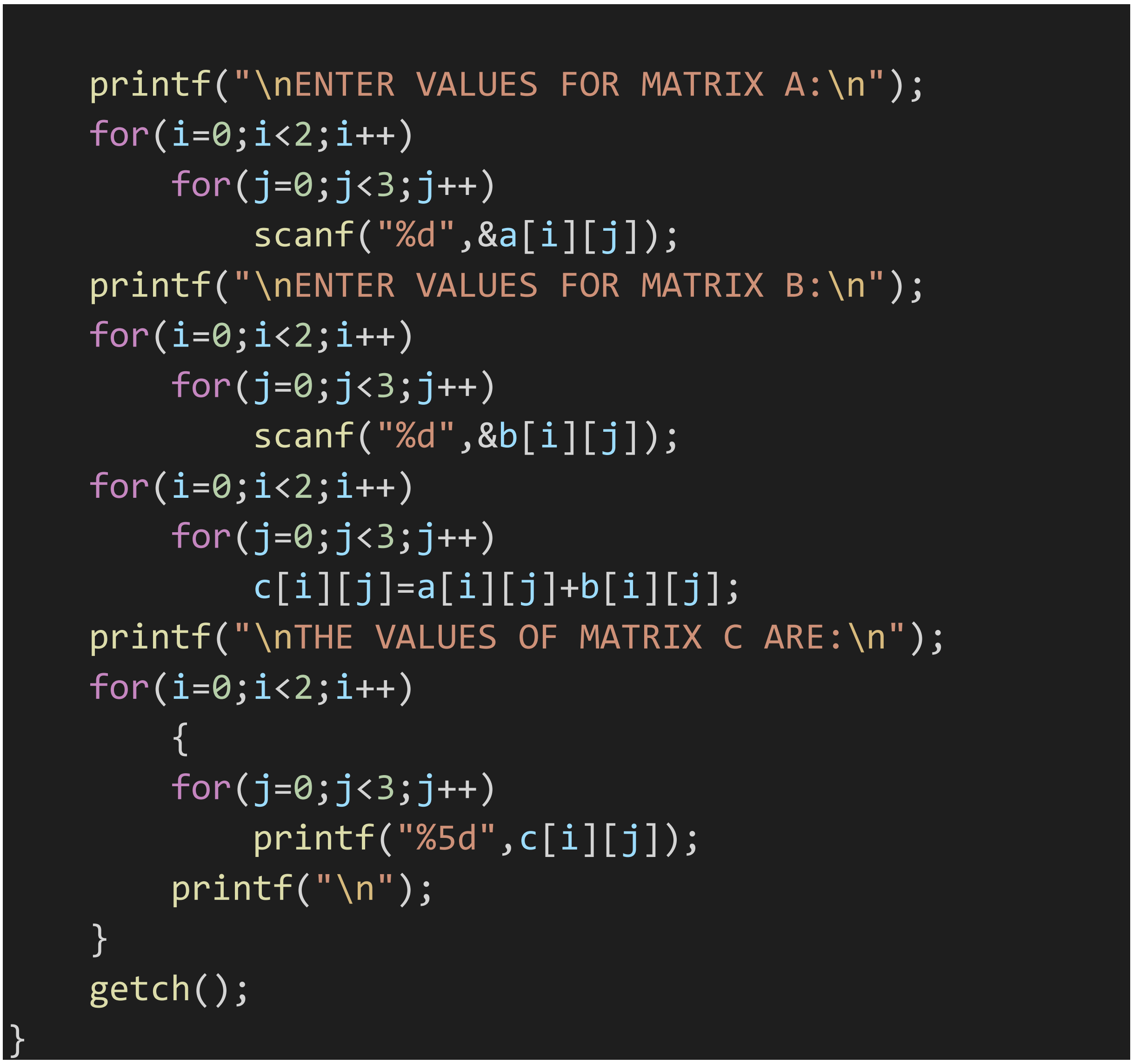
Write a C program to perform the basic Matrix operations
i) Addition

2)Pseudo code:

```
Start
Define a[2][3],b[2][3],c[2][3],i,j
Input the matrix a and b
Then
For(i=0;i<2;i++)
   For(i=0;i<3;j++)
        c[i][j]=a[i][j]+b[i][j]
then output the c matrix
For(i=0;i<2;i++)
   For(i=0;i<3;j++)
        Output (c[i][j])
```

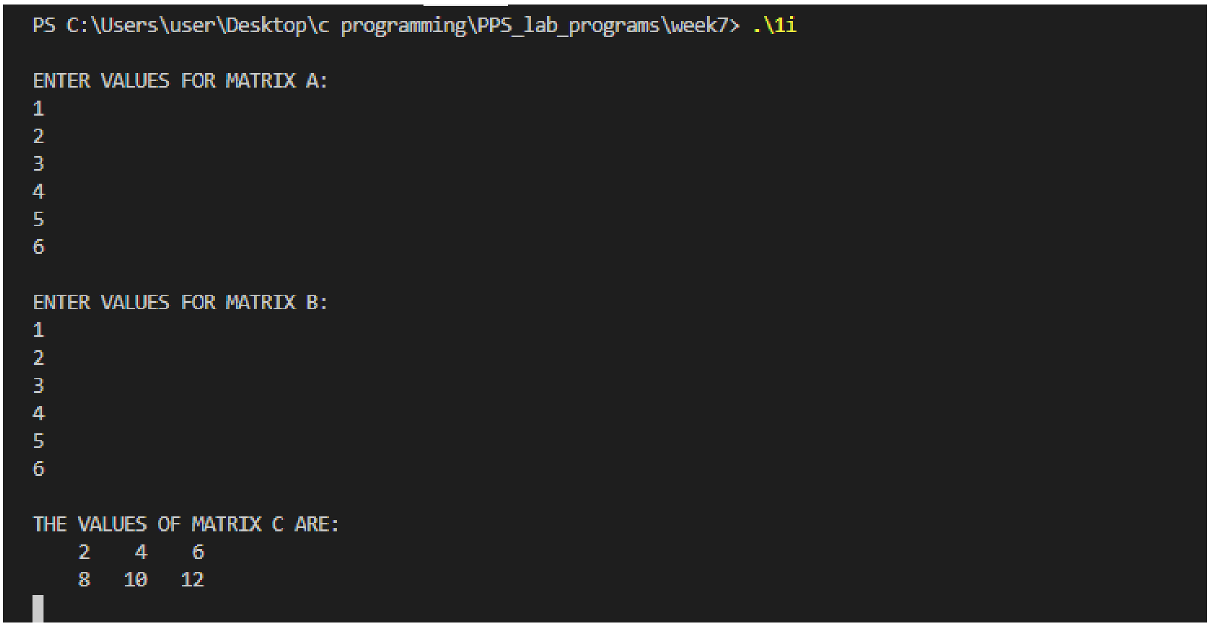
3)Executable C program :

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[2][3],b[2][3],c[2][3],i,j;
```



```
    printf("\nENTER VALUES FOR MATRIX A:\n");
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);
    printf("\nENTER VALUES FOR MATRIX B:\n");
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            scanf("%d",&b[i][j]);
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            c[i][j]=a[i][j]+b[i][j];
    printf("\nTHE VALUES OF MATRIX C ARE:\n");
    for(i=0;i<2;i++)
        {
        for(j=0;j<3;j++)
            printf("%5d",c[i][j]);
        printf("\n");
    }
    getch();
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\1i

ENTER VALUES FOR MATRIX A:
1
2
3
4
5
6

ENTER VALUES FOR MATRIX B:
1
2
3
4
5
6

THE VALUES OF MATRIX C ARE:
    2    4    6
    8   10   12
```

1ii))

1)Problem statement:

ii) Subtraction

2)Pseudo code:

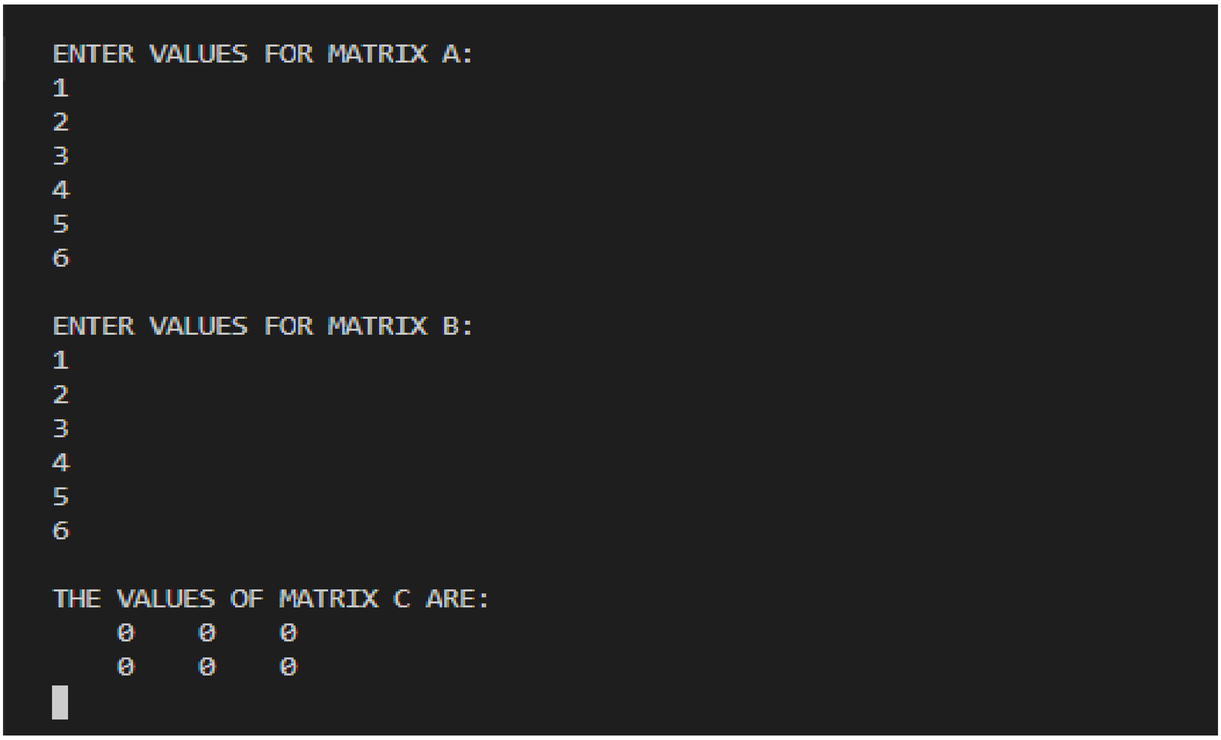
```
Start
Define a[2][3],b[2][3],c[2][3],i,j
Input the matrix a and b
Then
For(i=0;i<2;i++)
  For(i=0;i<3;j++)
      c[i][j]=a[i][j]-b[i][j]
then output the c matrix
```

```
For(i=0;i<2;i++)
  For(i=0;i<3;j++)
      Output (c[i][j])
```

3)Executable C program :

```c
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[2][3],b[2][3],c[2][3],i,j;

    printf("\nENTER VALUES FOR MATRIX A:\n");
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            scanf("%d",&a[i][j]);
    printf("\nENTER VALUES FOR MATRIX B:\n");
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            scanf("%d",&b[i][j]);
    for(i=0;i<2;i++)
        for(j=0;j<3;j++)
            c[i][j]=a[i][j]-b[i][j];
    printf("\nTHE VALUES OF MATRIX C ARE:\n");
    for(i=0;i<2;i++)
        {
        for(j=0;j<3;j++)
            printf("%5d",c[i][j]);
        printf("\n");
    }
    getch();
}
```

4)Output:



```
ENTER VALUES FOR MATRIX A:
1
2
3
4
5
6

ENTER VALUES FOR MATRIX B:
1
2
3
4
5
6

THE VALUES OF MATRIX C ARE:
    0    0    0
    0    0    0
```

1iii))

1)Problem statement:

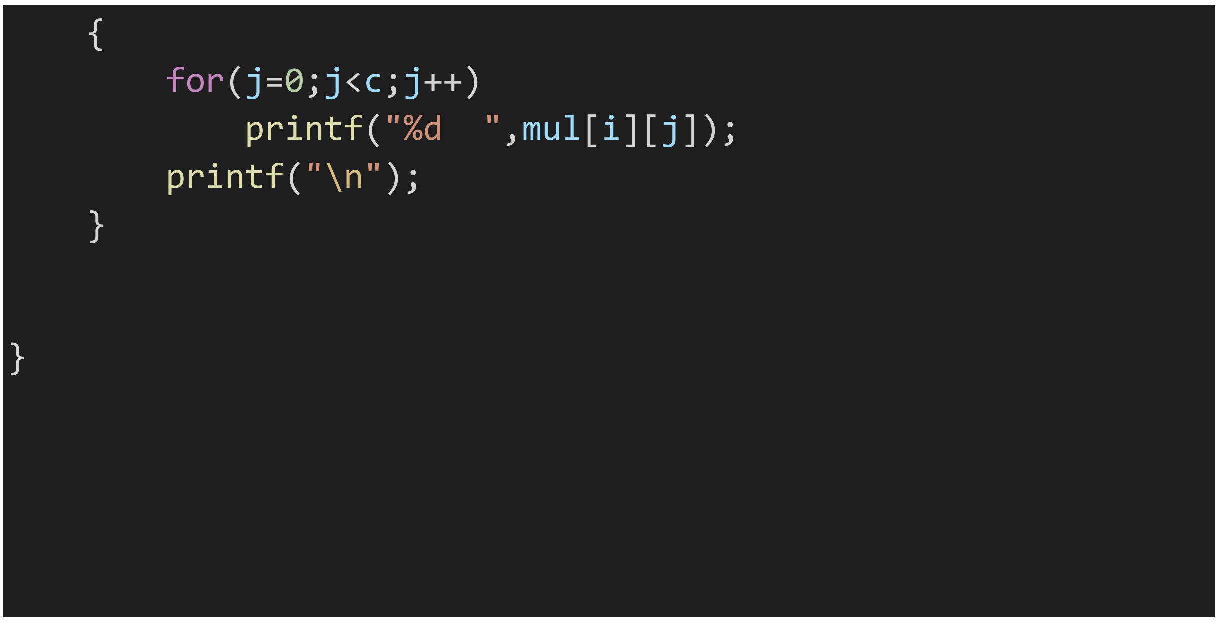
iii) Multiplication

2)Pseudo code:

```
Start
Define a[2][3],b[2][3],c[2][3],i,j
Input the matrix a and b
Then
For(i=0;i<2;i++)
  For(i=0;i<3;j++)
     Mul[i][j]+=a[i][k]*b[k][j]
then output the c matrix
For(i=0;i<2;i++)
For(i=0;i<3;j++)
    Output (mul[i][j])
```

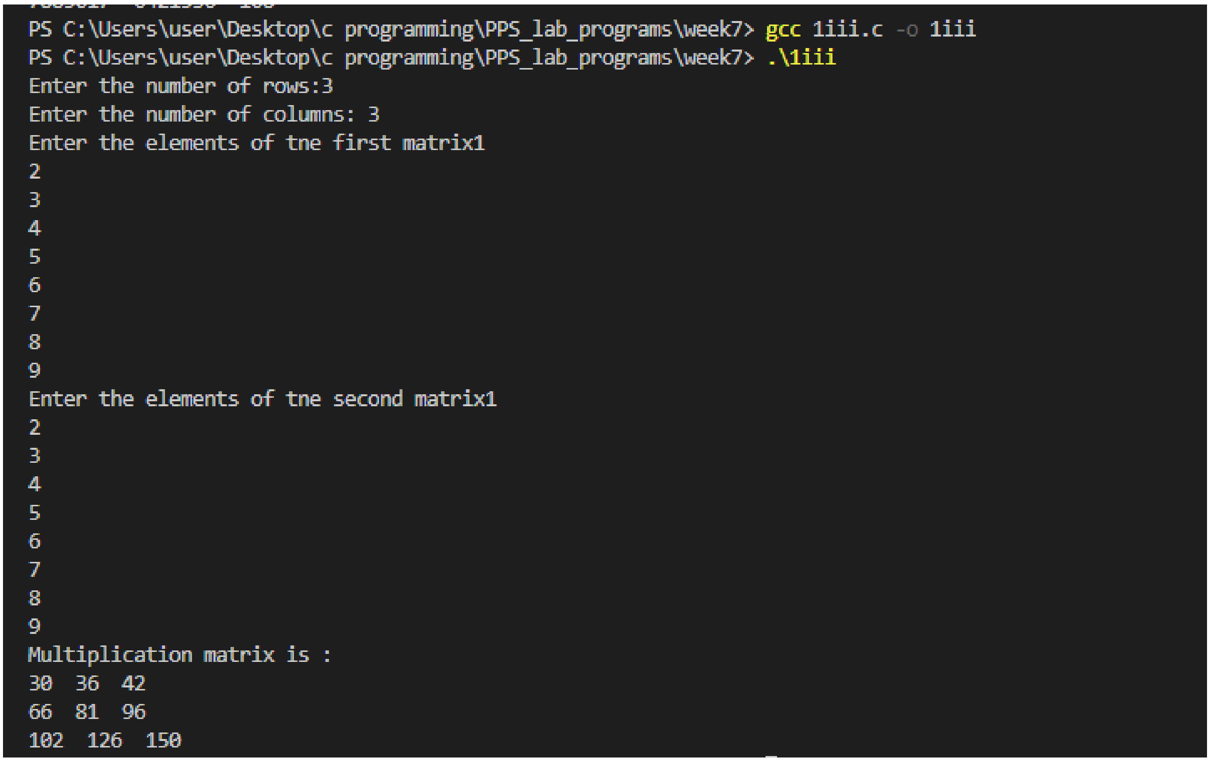
3)Executable C program :

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[10][10],b[10][10],mul[10][10],r,c,i,j,k;
    printf("Enter the number of rows:");
    scanf("%d",&r);
    printf("Enter the number of columns: ");
    scanf("%d",&c);
    printf("Enter the elements of tne first matrix");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            scanf("%d",&a[i][j]);
    }
        printf("Enter the elements of tne second matrix");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            scanf("%d",&b[i][j]);
    }
    printf("Multiplication matrix is :\n");
    for(i=0;i<r;i++)
     {
         for(j=0;j<c;j++)
          {
             mul[i][j]=0;
             for(k=0;k<c;k++)
              {
                 mul[i][j]=mul[i][j]+a[i][k]*b[k][j];
              }
          }
     }
    for(i=0;i<r;i++)
```



```
    {
        for(j=0;j<c;j++)
            printf("%d  ",mul[i][j]);
        printf("\n");
    }


}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> gcc 1iii.c -o 1iii
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\1iii
Enter the number of rows:3
Enter the number of columns: 3
Enter the elements of tne first matrix1
2
3
4
5
6
7
8
9
Enter the elements of tne second matrix1
2
3
4
5
6
7
8
9
Multiplication matrix is :
30  36  42
66  81  96
102  126  150
```

1iv))

1)Problem statement:

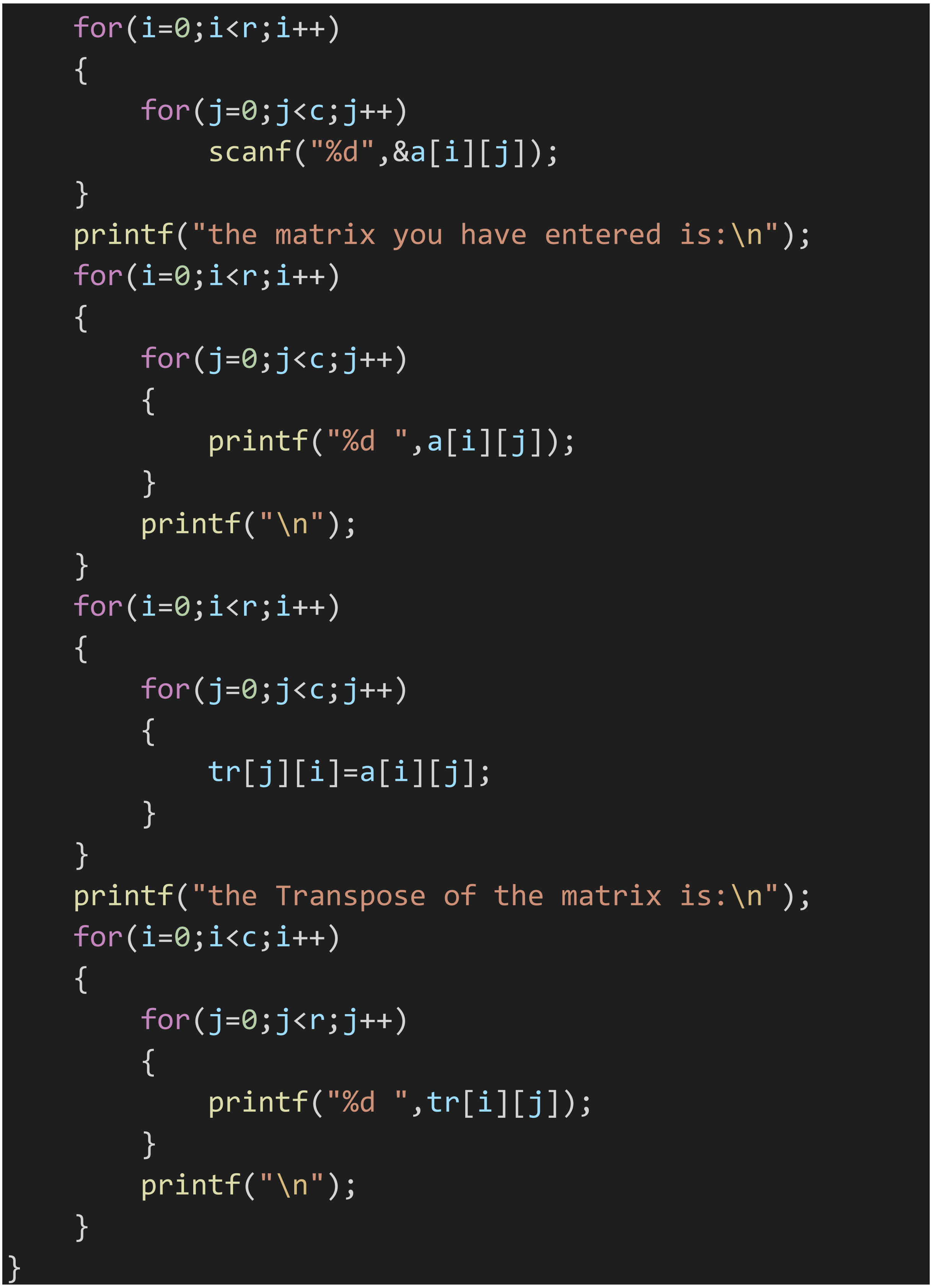
iv) Transpose.

2)Pseudo code:

```
Start
Define a[2][3],b[2][3],c[2][3],i,j
Input the matrix a and temp
Then
For(i=0;i<2;i++)
  For(i=0;i<3;j++)
     Temp[j][i]=a[i][j]
then output the c matrix
For(i=0;i<2;i++)
  For(i=0;i<3;j++)
      Output (temp [i][j])
```

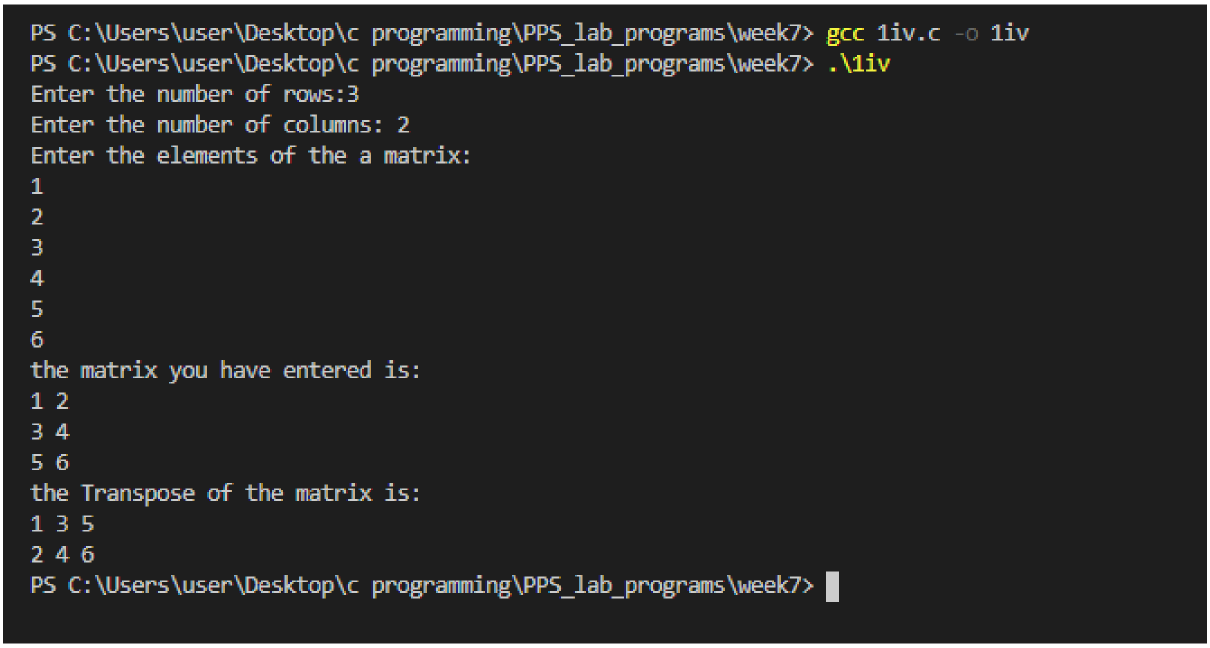
3)Executable C program :

```
//iv)transpose
#include <stdio.h>
#include <conio.h>
void main()
{
    int a[10][10],tr[10][10],r,c,i,j;
    printf("Enter the number of rows:");
    scanf("%d",&r);
    printf("Enter the number of columns: ");
    scanf("%d",&c);
    printf("Enter the elements of the a matrix:\n");
```



```
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
            scanf("%d",&a[i][j]);
    }
    printf("the matrix you have entered is:\n");
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            tr[j][i]=a[i][j];
        }
    }
    printf("the Transpose of the matrix is:\n");
    for(i=0;i<c;i++)
    {
        for(j=0;j<r;j++)
        {
            printf("%d ",tr[i][j]);
        }
        printf("\n");
    }
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> gcc 1iv.c -o 1iv
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\1iv
Enter the number of rows:3
Enter the number of columns: 2
Enter the elements of the a matrix:
1
2
3
4
5
6
the matrix you have entered is:
1 2
3 4
5 6
the Transpose of the matrix is:
1 3 5
2 4 6
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7>
```

2))

1)Problem statement:

Write a C program to determine if the given string is a palindrome or not

2)Pseudo code:

```
Start
Declare s[1000]
Input the string
Find the length of  the string
Start for loop
For(i=0;i<n;i++)
   If(s[i]==s[n-i-1]
      Increase c by 1
If c==i
```

Output string is palindrome

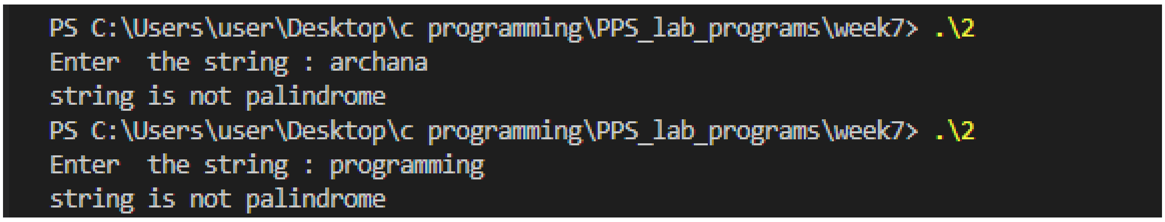
Else

 Output string is not a palindrome

3)Executable C program :

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

int main()
{
    char s[1000];
    int i,n,c=0;

    printf("Enter  the string : ");
    gets(s);
    n=strlen(s);

    for(i=0;i<n;i++)
    {
        if(s[i]==s[n-i-1])
        c++;

    }
    if(c==i)
        printf("string is palindrome");
    else
        printf("string is not palindrome");



  return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\2
Enter  the string : archana
string is not palindrome
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\2
Enter  the string : programming
string is not palindrome
```

3))

1)Problem statement:

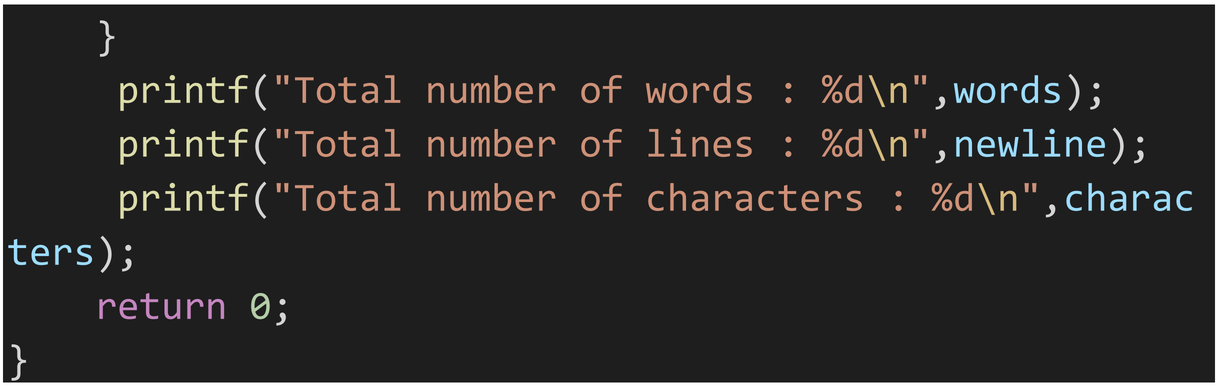
Write a C program to count the lines, words and characters in a given text

2)Pseudo code:

```
Start
Declare str[100],words=0;newline=0;characters=0
Input the string
Start the for loop
For(i=0;str[i]!=’\0’;i++)
  If(str[i]==’ ‘)
    Increments words by 1
Else if(str[i]!=’ ‘ and str[i]!=’\n’
 Increment character++
If character>0
   Increment words by 1
   Increment newline by 1
Then
Output
Total words     Total lines    Total characters
```

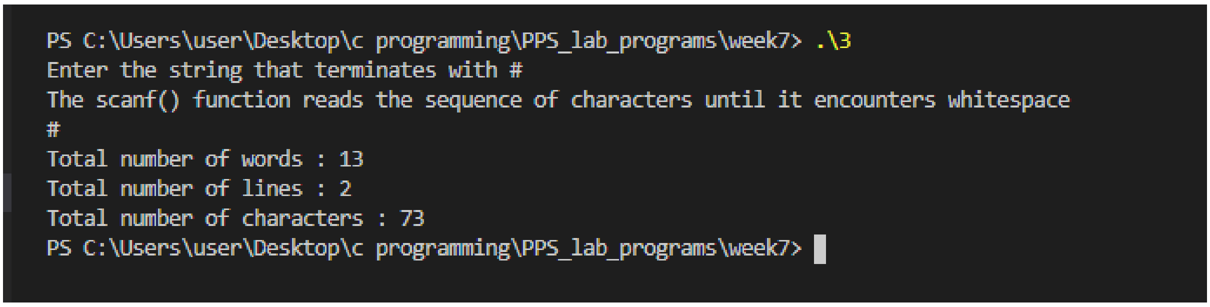
3)Executable C program :

```
#include <stdio.h>
int main()
{
    char str[100];//input string with size 100

    int words=0,newline=0,characters=0; // counter var
iables

    printf("Enter the string that terminates with #\n"
);

    scanf("%[^#]",&str);//scanf formatting

    for(int i=0;str[i]!='\0';i++)
     {
         if(str[i] == ' ')
         {
              words++;
         }
         else if(str[i] == '\n')
         {
             newline++;
              words++;//since with every next line new
 words start. corner case 1
         }
         else if(str[i] != ' ' && str[i] != '\n'){
         characters++;
         }
     }
    if(characters > 0)//Corner case 2,3.
    {
        words++;
        newline++;
```



```
    }
     printf("Total number of words : %d\n",words);
     printf("Total number of lines : %d\n",newline);
     printf("Total number of characters : %d\n",charac
ters);
    return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7> .\3
Enter the string that terminates with #
The scanf() function reads the sequence of characters until it encounters whitespace
#
Total number of words : 13
Total number of lines : 2
Total number of characters : 73
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week7>
```

Week 8

Tutorial 7: Functions, call by value:

Lab 7: Simple functions

1i))

1)Problem statement:

Write a C Function for the following task
i) Calculating Factorial

2)Pseudo code:

Start

Declare num,factorial

Input num

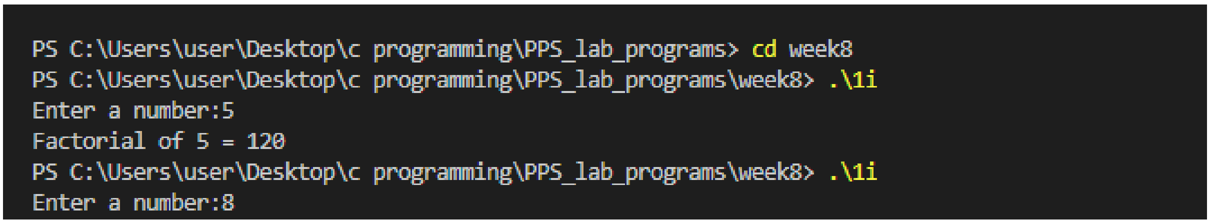
```
Call fact
Factorial=fact(num);
Output factorial
Define fact()
 Define factorial=1
For(i=1;i<=num;i++)
     Factorial=factorial*i
Return(factorial);
End
```

3)Executable C program :

```c
//i) Calculating Factorial
#include <stdio.h>
int fact(int);
int main()
{
    int num;
    int factorial;
    printf("Enter a number:");
    scanf("%d",&num);
    factorial=fact(num);
    printf("Factorial of %d = %ld\n",num,factorial);
    return 0;
}
int fact(int num)
{
    int i;
    int factorial=1;
    for(i=1;i<=num;i++)
        factorial=factorial*i;
    return(factorial);
```

```
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs> cd week8
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1i
Enter a number:5
Factorial of 5 = 120
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1i
Enter a number:8
```

1ii))

1)Problem statement:

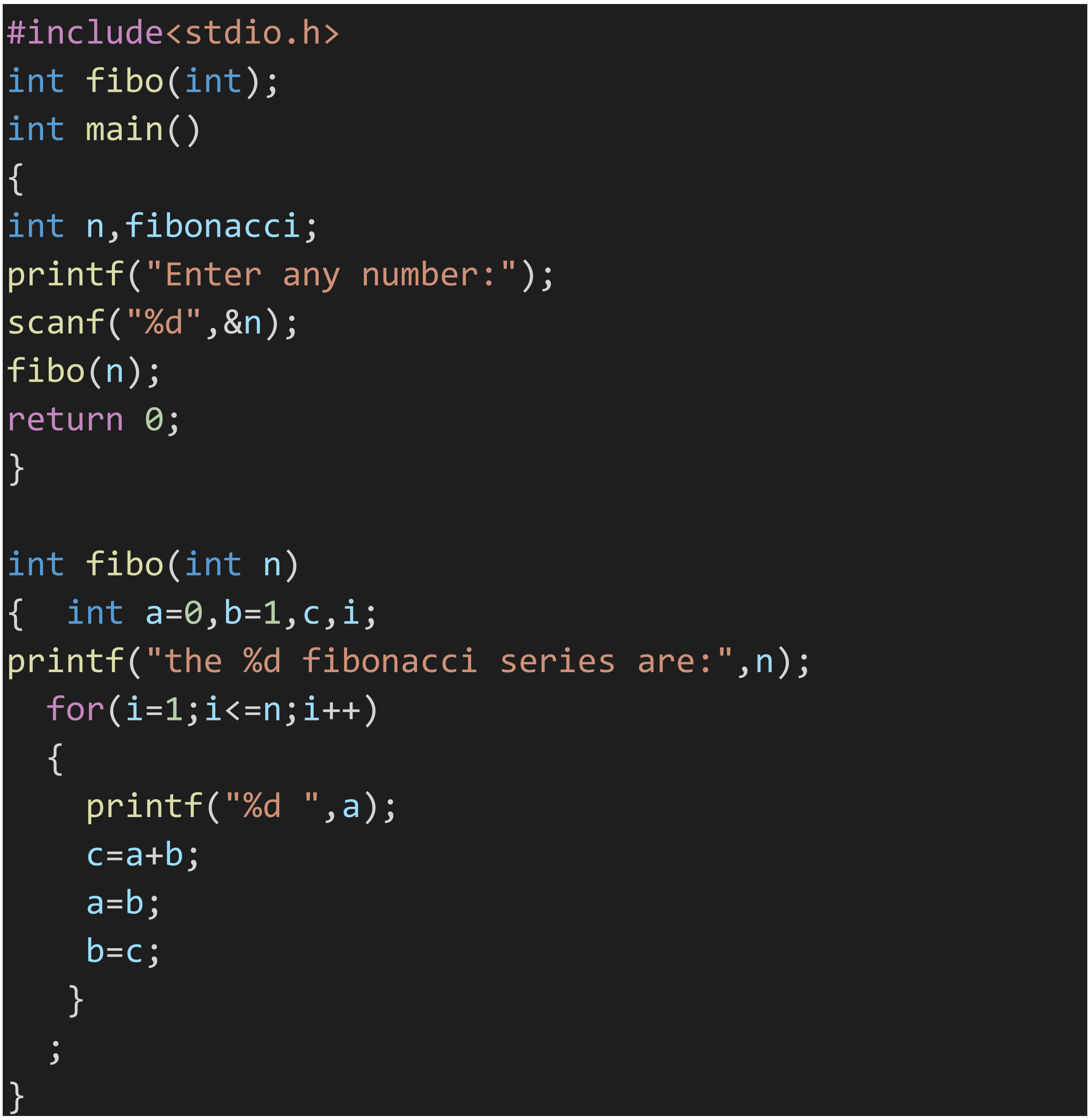
ii) Find value of a given Fibonacci term

2)Pseudo code:

```
Start
Declare n,Fibonacci
Input n
Call fibo(n);
Define fibo(n)
Declare a=0,b=1,c,i
Start for loop
For(i=1;i<n;i++)
Output a
c=a+b
 a=b
b=c
```

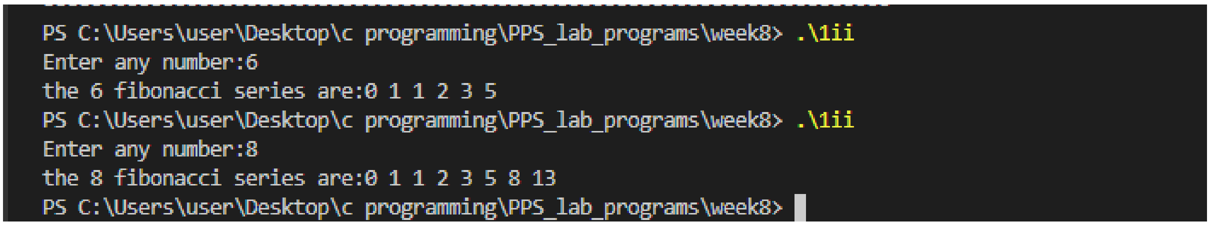
3)Executable C program :

```
//ii) Find value of a given Fibonacci term
```



```c
#include<stdio.h>
int fibo(int);
int main()
{
int n,fibonacci;
printf("Enter any number:");
scanf("%d",&n);
fibo(n);
return 0;
}

int fibo(int n)
{  int a=0,b=1,c,i;
printf("the %d fibonacci series are:",n);
  for(i=1;i<=n;i++)
  {
    printf("%d ",a);
    c=a+b;
    a=b;
    b=c;
   }
  ;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1ii
Enter any number:6
the 6 fibonacci series are:0 1 1 2 3 5
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1ii
Enter any number:8
the 8 fibonacci series are:0 1 1 2 3 5 8 13
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8>
```

1iii))

1)Problem statement:

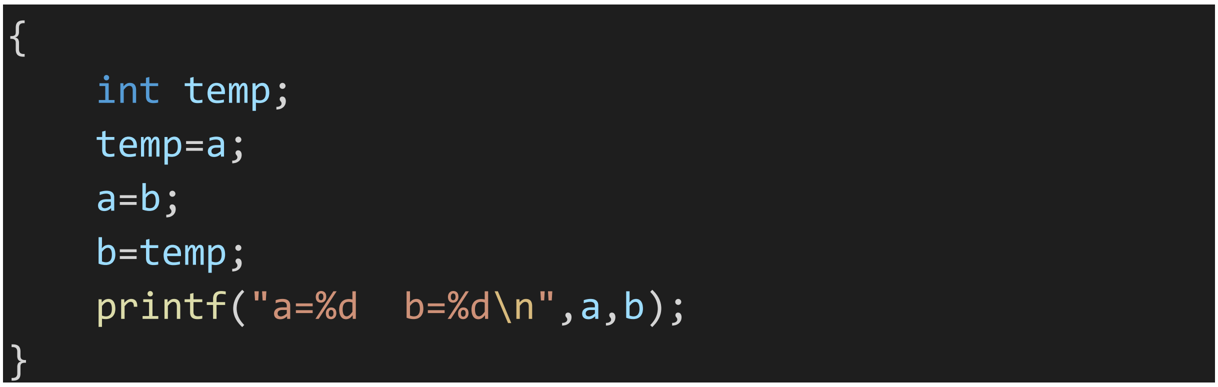
iii) Swapping the values of two variable

2)Pseudo code:

```
Start
Declare a,b
Call function swap(a,b)
Go to function definition
Swap(int a,int b)
Declare Int temp
temp=a
a=b
b=temp
output a,b
```

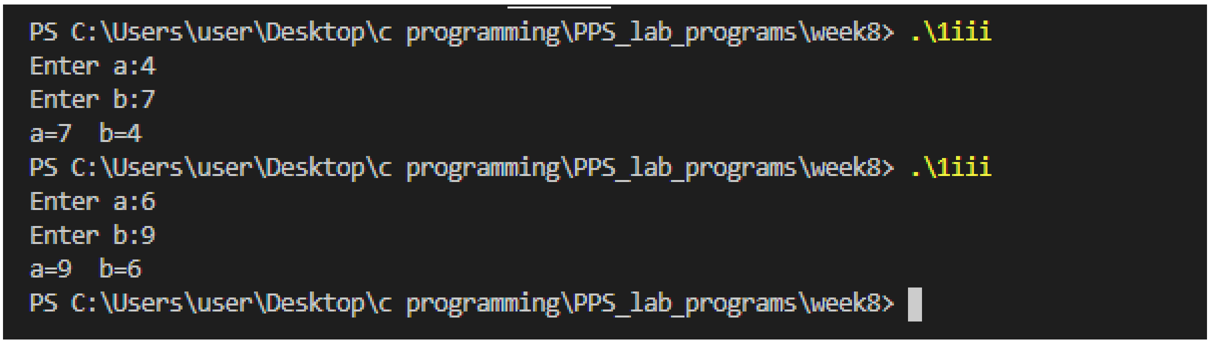
3)Executable C program :

```c
//iii) Swapping the values of two variable
#include <stdio.h>
void swap(int,int);
int main()
{
    int a,b;
    printf("Enter a:");
    scanf("%d",&a);
    printf("Enter b:");
    scanf("%d",&b);
    swap(a,b);
    return 0;
}
void swap(int a,int b)
```



```
{
    int temp;
    temp=a;
    a=b;
    b=temp;
    printf("a=%d  b=%d\n",a,b);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1iii
Enter a:4
Enter b:7
a=7  b=4
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\1iii
Enter a:6
Enter b:9
a=9  b=6
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> 
```

2i))

1)Problem statement:

2. Write a C program that uses functions to perform the following operations:
i) To insert a sub-string in to a given main string from a given position.

2)Pseudo code:

Start

Declare i,pos,str[100],substr[30]

Input i,pos

Call substring(str,substr,i);

Defining substring(str,substr,int)

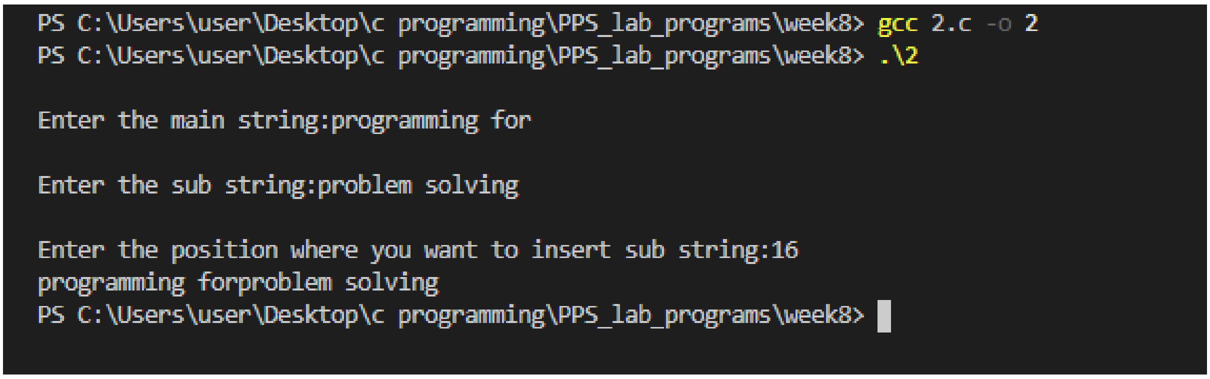
Declare temp[100],m,n,k,j

```
m=strlen(str)
n=strlen(substr)
for(j=0;j<i;j++)
    temp[j]=str[j]
for(j=1,k=0;j<m,k<m;j++,k++)
    temp[j]=subtr[k]
for(j=n+I,k=i;j<m,k<m;j++,k++)
   temp[j]=str[k]
output temp
```

3)Executable C program :

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

//Declaration of function substring to insert substring into main string
void substring(char str[100],char substr[30],int i);

//Main function to execute program
void main()
{
char str[100],substr[30];
int i,pos;
printf("\nEnter the main string:");
gets(str);
printf("\nEnter the sub string:");
gets(substr);
printf("\nEnter the position where you want to insert sub string:");
scanf("%d",&pos);
i=pos-1;
//Call to substring() function to perform required task
substring(str,substr,i);
}

//Definition of substring() function
void substring(char str[100],char substr[30],int i)
{
char temp[100];
int m,n,k,j;

//strlen(str) function to measure length of the string
m=strlen(str);
n=strlen(substr);

//str is copied into temp from 0 to i
for(j=0;j<i;j++)
{
temp[j]=str[j];
}

//substr is copied into temp from i to n
for(j=i,k=0;j<n+i,k<n;j++,k++)
temp[j]=substr[k];

//remaining str is copied into temp from n+i to m
for(j=n+i,k=i;j<m,k<m;j++,k++)
temp[j]=str[k];

//puts() function to print temp
puts(temp);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> gcc 2.c -o 2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\2

Enter the main string:programming for

Enter the sub string:problem solving

Enter the position where you want to insert sub string:16
programming forproblem solving
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> 
```

2ii))

1)Problem statement:

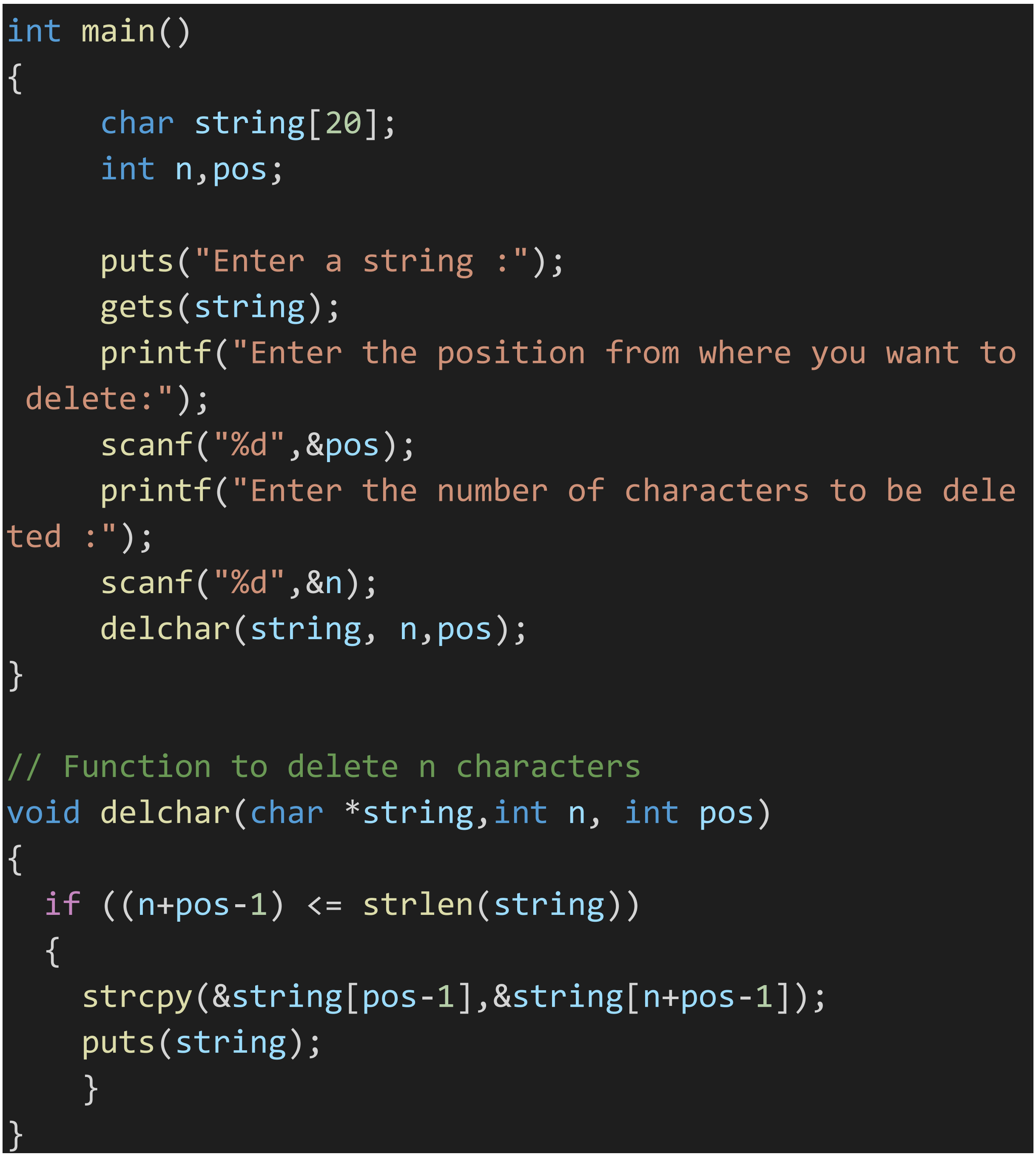
ii) To delete n Characters from a given position in a given string.

2)Pseudo code:

```
Start
Declare string[20],pos,n
Input string[20],pos,n
Call delchar(string,n,pos)
Define delchar(char *string,int n,int pos)
 If n+pos-1<=strlen(string)
   Strcpy(&string[pos-1],&string[n+pos-1])
    Output string
End
```

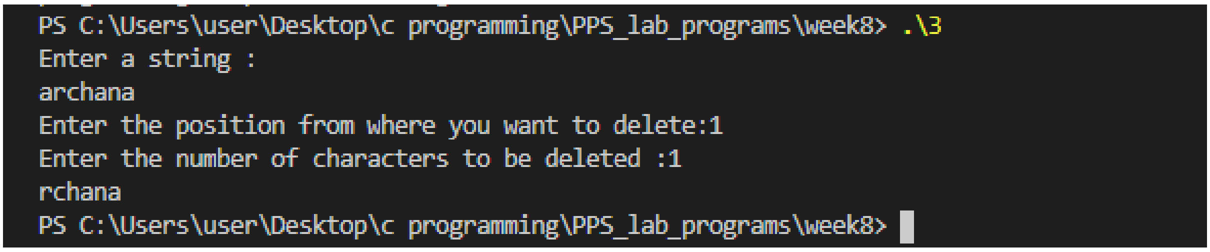
3)Executable C program :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
 void delchar(char *string,int n, int pos);
```



```c
int main()
{
     char string[20];
     int n,pos;

     puts("Enter a string :");
     gets(string);
     printf("Enter the position from where you want to
 delete:");
     scanf("%d",&pos);
     printf("Enter the number of characters to be dele
ted :");
     scanf("%d",&n);
     delchar(string, n,pos);
}

// Function to delete n characters
void delchar(char *string,int n, int pos)
{
  if ((n+pos-1) <= strlen(string))
  {
    strcpy(&string[pos-1],&string[n+pos-1]);
    puts(string);
    }
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8> .\3
Enter a string :
archana
Enter the position from where you want to delete:1
Enter the number of characters to be deleted :1
rchana
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week8>
```

Week 9

Tutorial 10: Recursion, structure of recursive calls

Lab 10: Recursive functions

1))

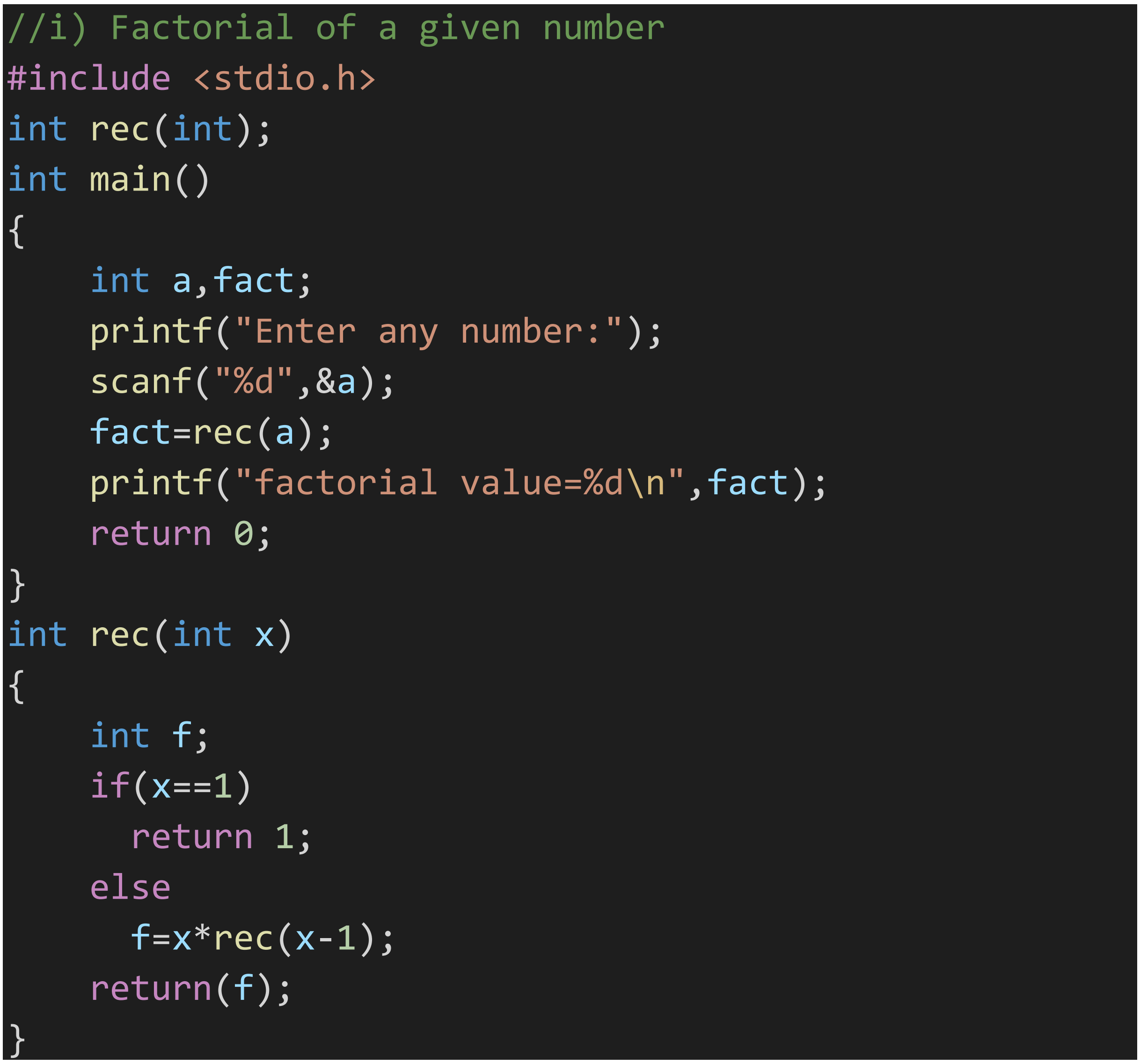
1)Problem statement:

1) Write the following recursive C Function
i) Factorial of a given number

2)Pseudo code:

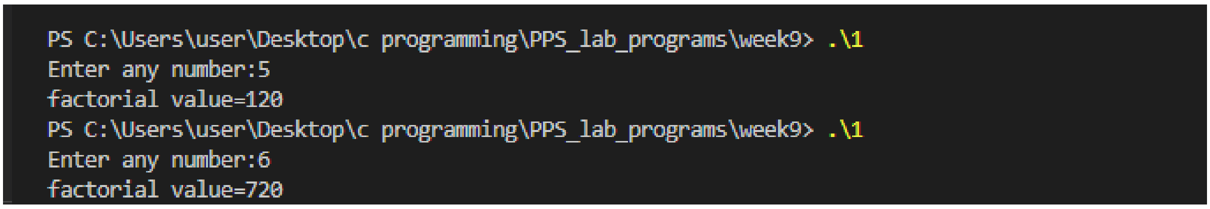
```
Start
Declare a,fact
Input a
Call rec(a);
Define rec(int a)
Declare int f
If x==1
  Return 1
Else
  F=x*rec(x-1)
Return f
End
```

3)Executable C program :



```c
//i) Factorial of a given number
#include <stdio.h>
int rec(int);
int main()
{
    int a,fact;
    printf("Enter any number:");
    scanf("%d",&a);
    fact=rec(a);
    printf("factorial value=%d\n",fact);
    return 0;
}
int rec(int x)
{
    int f;
    if(x==1)
      return 1;
    else
      f=x*rec(x-1);
    return(f);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1
Enter any number:5
factorial value=120
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1
Enter any number:6
factorial value=720
```

1ii))

1)Problem statement:

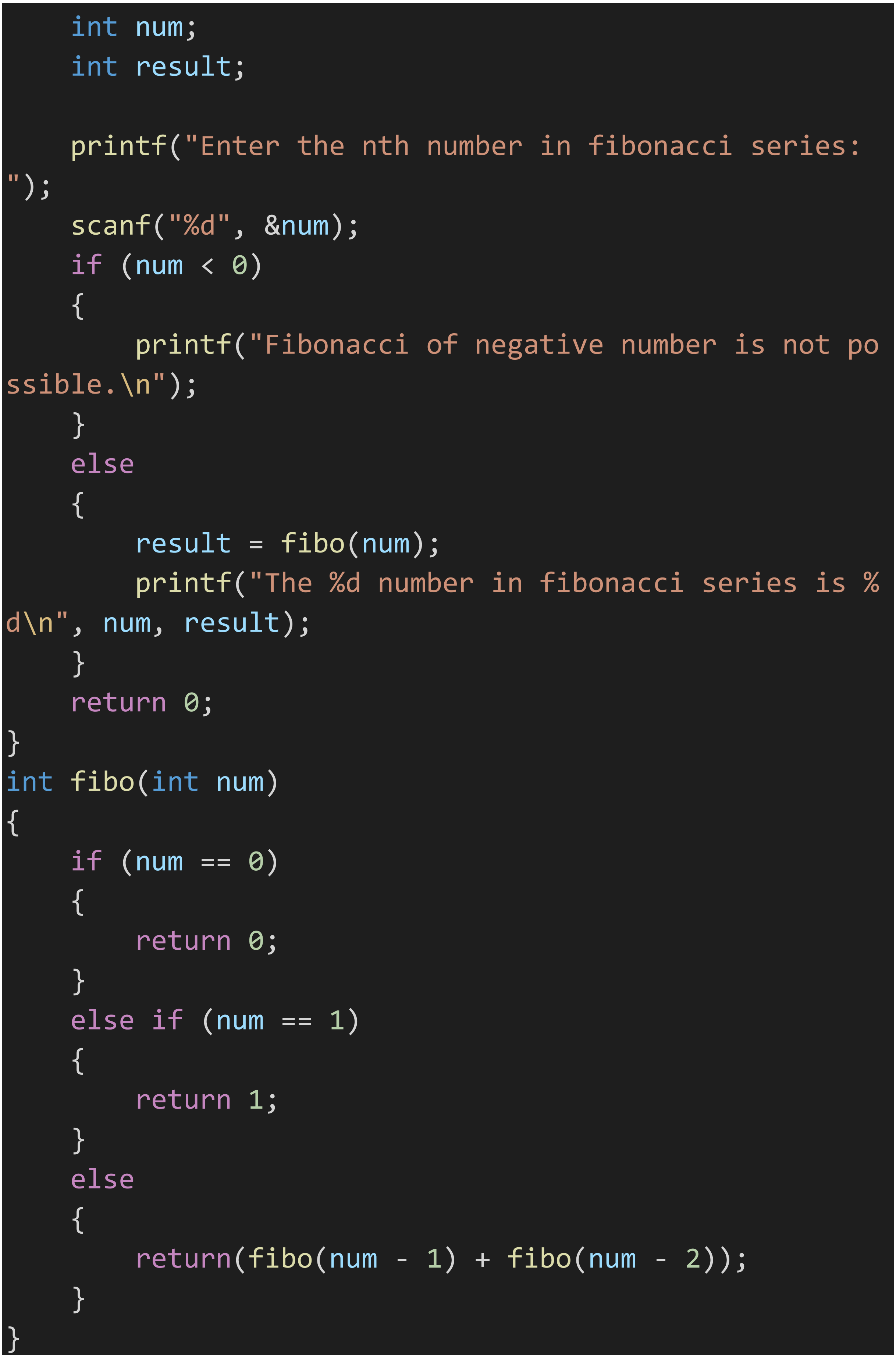
ii) Nth Fibonacci number

2)Pseudo code:

```
Start
Declare num,result
Input num
If num<0
  Output Fibonacci of negative number is not possible
Else
  Result=fibo(num);
  Output the Fibonacci series nth number is result
Define fibo(num)
If num==0
 Return 0
Else if num==1
  Return 1
Else
  return(fibo(num-1)+fibo(num-2));
```

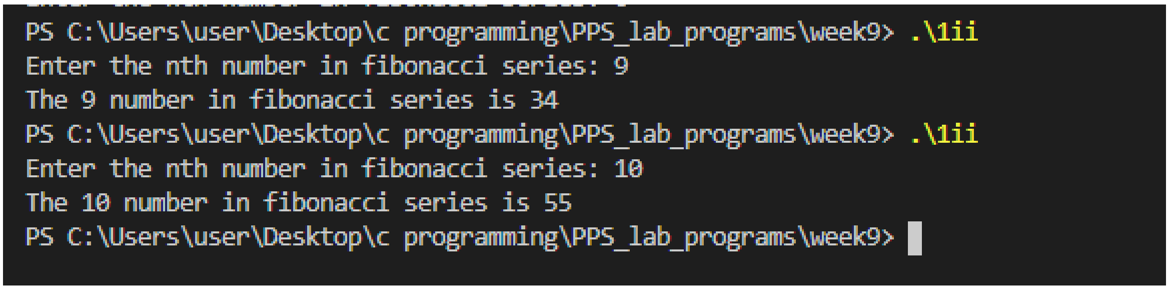
3)Executable C program :

```
//Nth Fibonacci number

#include <stdio.h>
int fibo(int);

int main()
{
```



```c
    int num;
    int result;

    printf("Enter the nth number in fibonacci series: ");
    scanf("%d", &num);
    if (num < 0)
    {
        printf("Fibonacci of negative number is not possible.\n");
    }
    else
    {
        result = fibo(num);
        printf("The %d number in fibonacci series is %d\n", num, result);
    }
    return 0;
}
int fibo(int num)
{
    if (num == 0)
    {
        return 0;
    }
    else if (num == 1)
    {
        return 1;
    }
    else
    {
        return(fibo(num - 1) + fibo(num - 2));
    }
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1ii
Enter the nth number in fibonacci series: 9
The 9 number in fibonacci series is 34
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1ii
Enter the nth number in fibonacci series: 10
The 10 number in fibonacci series is 55
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9>
```

1iii))

1)Problem statement:

iii) Reverse of a given String

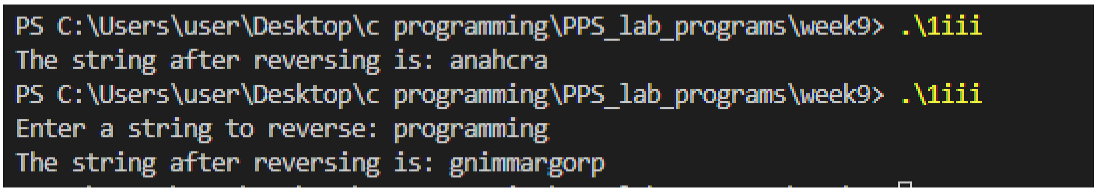
2)Pseudo code:

```
Start
Declare str[20],size
Input str
Size=strlen(str)
Call reverse(str,0,size-1);
Output
  Reverse_string
Define reverse()
   Declare temp
   Temp=str[index]
   str[index]=str[size-index]
   if(index==size/2)
      return
  reversr(str,index+1,size)
end
```

3)Executable C program :

```
/*
 * C Program to Reverse the String using Recursion
 */
#include <stdio.h>
#include <string.h>

void reverse(char [], int, int);
int main()
{
    char str1[20];
    int size;

    printf("Enter a string to reverse: ");
    scanf("%s", str1);
    size = strlen(str1);
    reverse(str1, 0, size - 1);
    printf("The string after reversing is: %s\n", str1
);
    return 0;
}

void reverse(char str1[], int index, int size)
{
    char temp;
    temp = str1[index];
    str1[index] = str1[size - index];
    str1[size - index] = temp;
    if (index == size / 2)
    {
        return;
    }
    reverse(str1, index + 1, size);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1iii
The string after reversing is: anahcra
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1iii
Enter a string to reverse: programming
The string after reversing is: gnimmargorp
```

1iv))

1)Problem statement:

iv) Reverse of a give Number

2)Pseudo code:

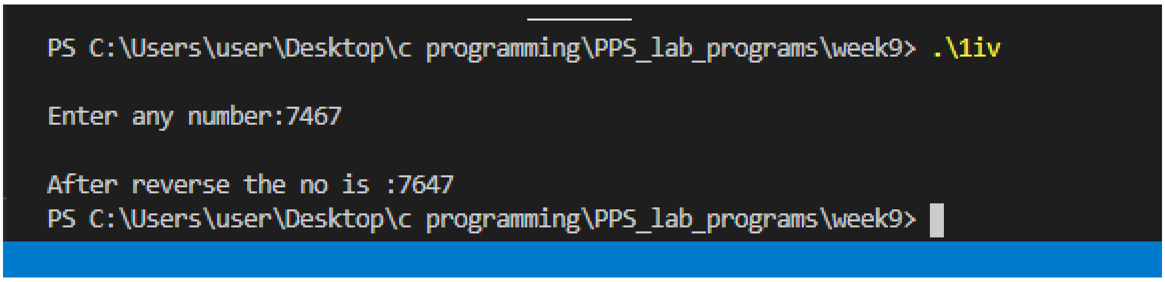
```
Start
Declare num,reverse_number
Input num
Reverse_number=reverse_function(num);
Output reverse_number
Define  reverse_function
Define num
Rem=num%10
Sum=sum*10+rem
Reverse_function(num/10)
Return sum;
```

End

3)Executable C program :

```
#include<stdio.h>
int reverse_number(int);
int main(){
   int num,reverse_number;

   //User would input the number
   printf("\nEnter any number:");
   scanf("%d",&num);

   //Calling user defined function to perform reverse
   reverse_number=reverse_function(num);
   printf("\nAfter reverse the no is :%d",reverse_number);
   return 0;
}
int sum=0,rem;
 int reverse_function(int num){
   if(num){
      rem=num%10;
      sum=sum*10+rem;
      reverse_function(num/10);
   }
   else
      return sum;
   return sum;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9> .\1iv

Enter any number:7467

After reverse the no is :7647
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week9>
```

Week 10

Tutorial 11: Pointers, structures and dynamic memory allocation

Lab 11: Pointers and structures

1))

1)Problem statement:

1. Write a C program to maintain a record of "n" student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Assume appropriate data type for each field. Print the marks of the student, given the student name as input.

2)Pseudo code:

```
Start
Define a structure of name roll marks and grade
Then start a loop for the n times as the no. suudents
```

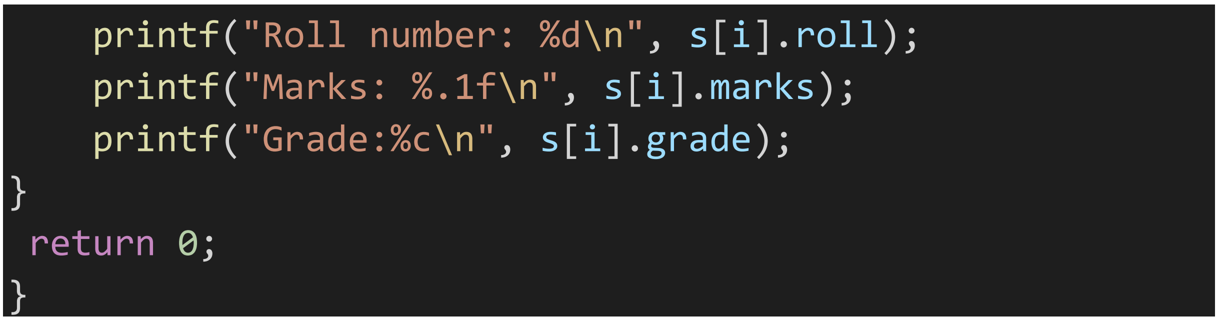
Input the values into structure

Output the values on to the screen

End

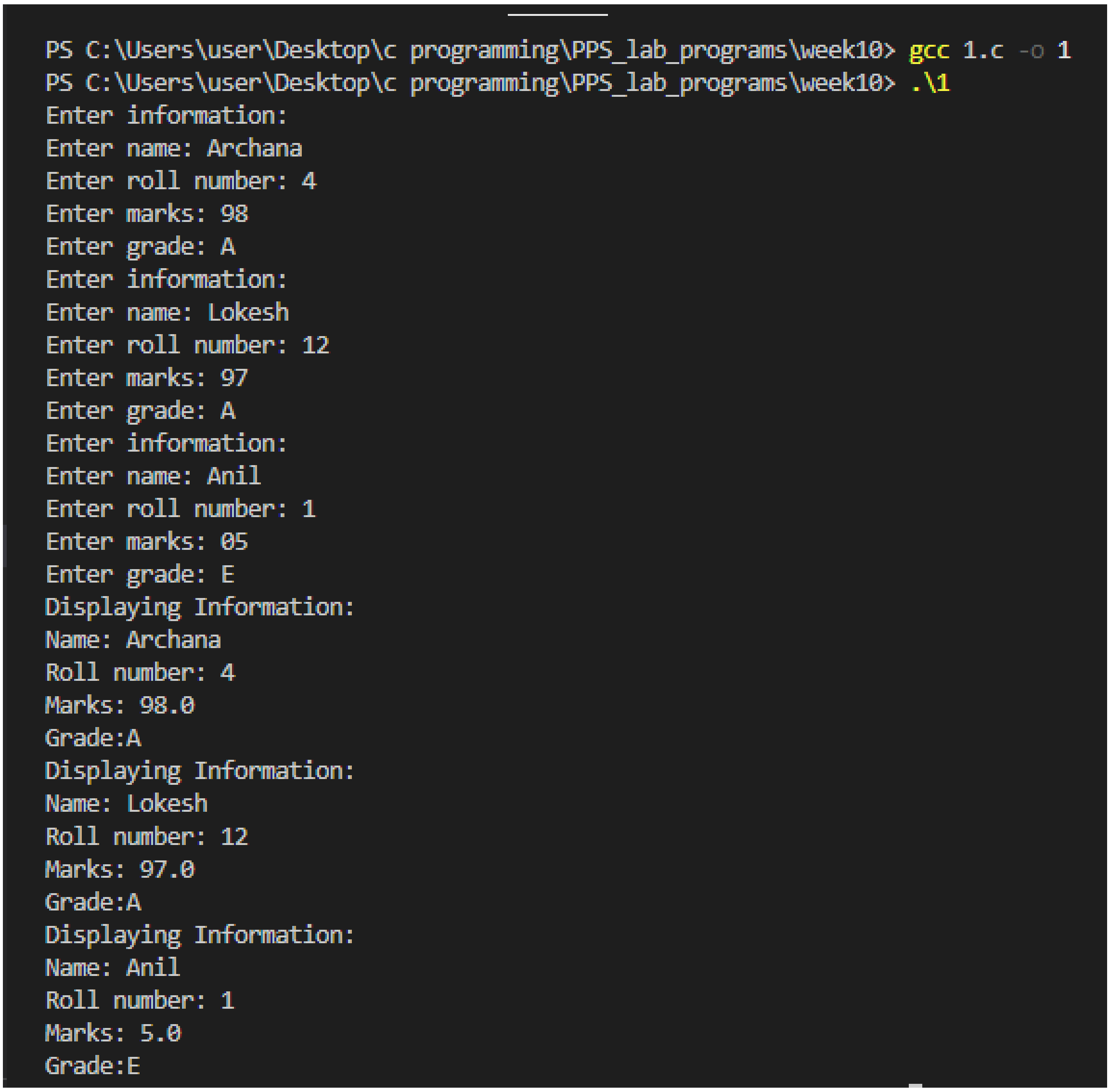
3)Executable C program :

```
#include <stdio.h>
struct student {
    char name[50];
    int roll;
    float marks;
    char grade;
} ;

int main()
{   struct student s[3];
    int i;

  for(i=0;i<3;i++)
  {
    printf("Enter information:\n");
    printf("Enter name: ");
    scanf("%s",&s[i].name);
    printf("Enter roll number: ");
    scanf("%d", &s[i].roll);
    printf("Enter marks: ");
    scanf("%f", &s[i].marks);
    printf("Enter grade: ");
    scanf("%s", &s[i].grade);
  }
for(i=0;i<3;i++)
{
    printf("Displaying Information:\n");
    printf("Name: ");
    printf("%s\n", s[i].name);
```



```
    printf("Roll number: %d\n", s[i].roll);
    printf("Marks: %.1f\n", s[i].marks);
    printf("Grade:%c\n", s[i].grade);
}
 return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> gcc 1.c -o 1
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> .\1
Enter information:
Enter name: Archana
Enter roll number: 4
Enter marks: 98
Enter grade: A
Enter information:
Enter name: Lokesh
Enter roll number: 12
Enter marks: 97
Enter grade: A
Enter information:
Enter name: Anil
Enter roll number: 1
Enter marks: 05
Enter grade: E
Displaying Information:
Name: Archana
Roll number: 4
Marks: 98.0
Grade:A
Displaying Information:
Name: Lokesh
Roll number: 12
Marks: 97.0
Grade:A
Displaying Information:
Name: Anil
Roll number: 1
Marks: 5.0
Grade:E
```

2))

1)Problem statement:

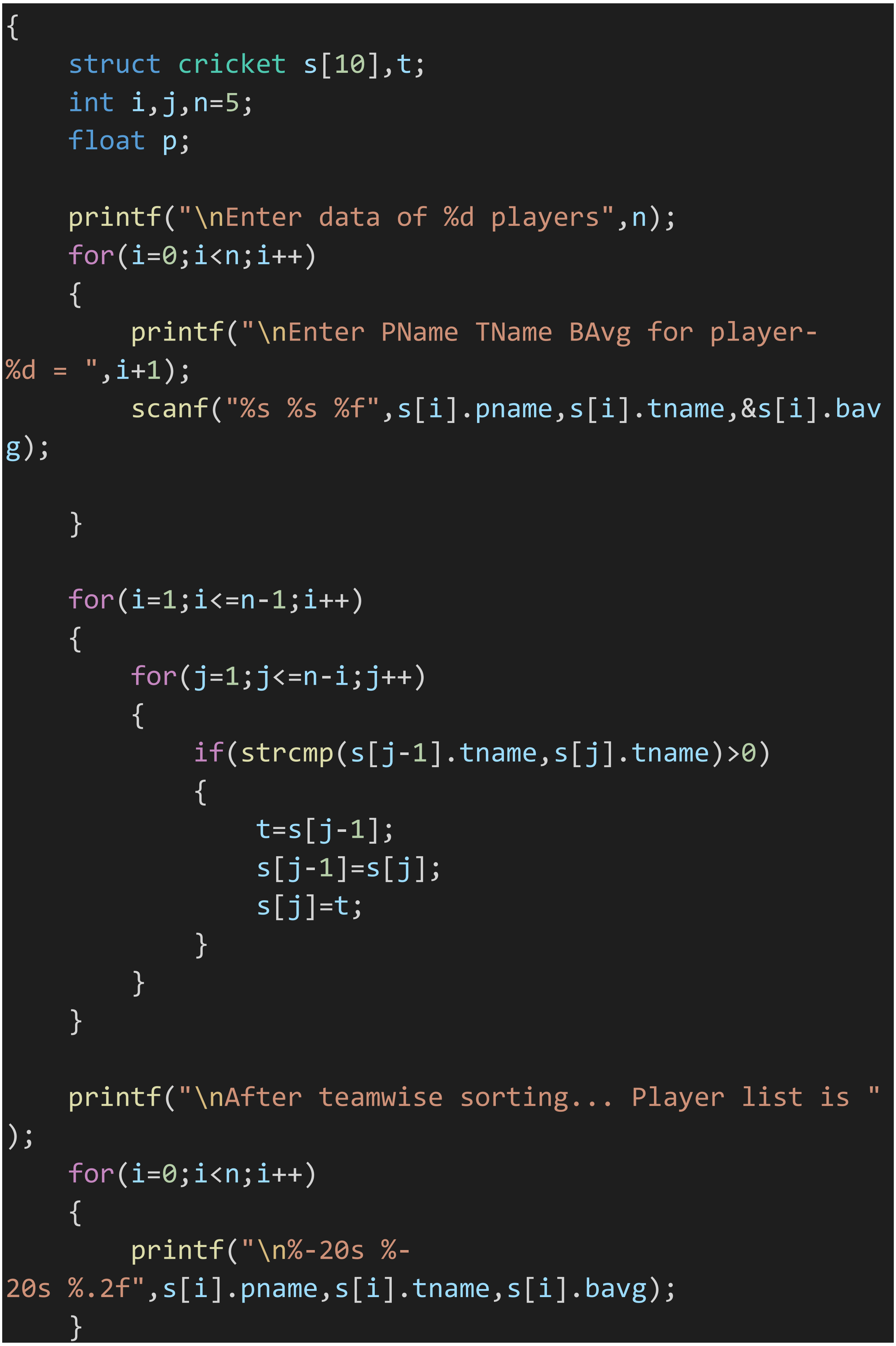
2. Define structure called cricket that will describe the information player name, team name,batting aveg.Using cricket ,declare an array player with 10 elements and write program to read information about all 10 players and print team wise list containing names of the player with their batting avg

2)Pseudo code:

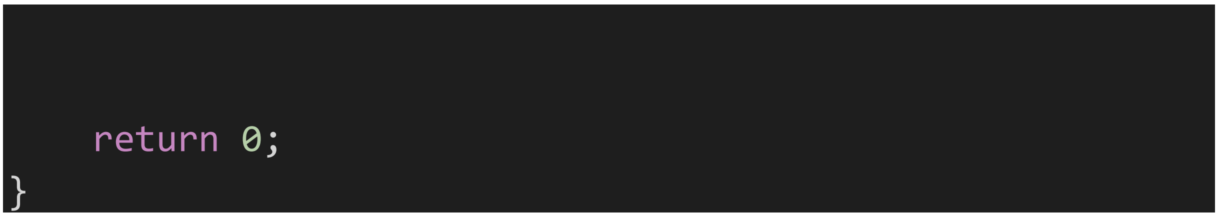
```
Start

Define structure called cricket that will describe the information
player name, team name,batting aveg.
Using cricket ,declare an array player with 10 elements
Input information about all 10 players
output
team wise list containing names of the player with their batting avg
```

3)Executable C program :

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

struct cricket
{
    char pname[20];
    char tname[20];
    float bavg;
};

int main()
```

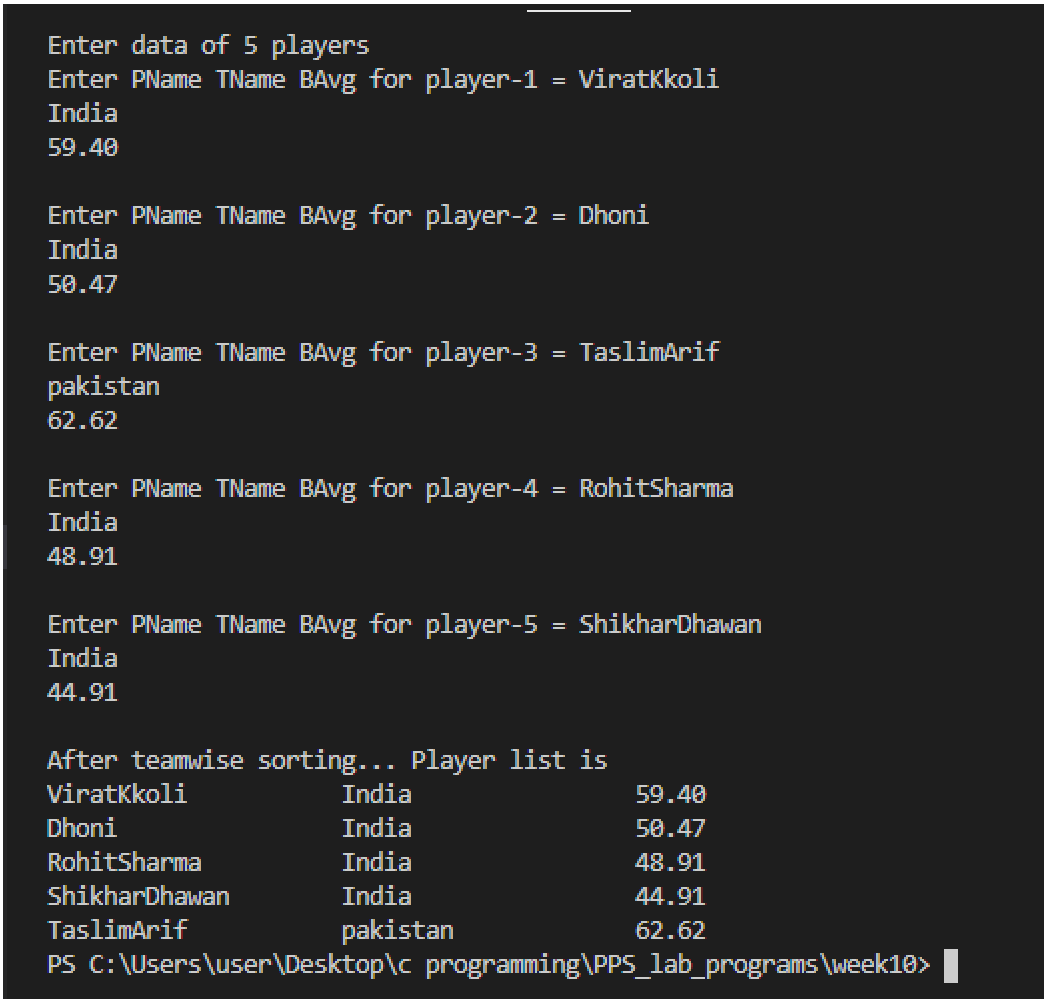


```
{
    struct cricket s[10],t;
    int i,j,n=5;
    float p;

    printf("\nEnter data of %d players",n);
    for(i=0;i<n;i++)
    {
        printf("\nEnter PName TName BAvg for player-
%d = ",i+1);
        scanf("%s %s %f",s[i].pname,s[i].tname,&s[i].bav
g);

    }

    for(i=1;i<=n-1;i++)
    {
        for(j=1;j<=n-i;j++)
        {
            if(strcmp(s[j-1].tname,s[j].tname)>0)
            {
                t=s[j-1];
                s[j-1]=s[j];
                s[j]=t;
            }
        }
    }

    printf("\nAfter teamwise sorting... Player list is "
);
    for(i=0;i<n;i++)
    {
        printf("\n%-20s %-
20s %.2f",s[i].pname,s[i].tname,s[i].bavg);
    }
```



```
    return 0;
}
```

4)Output:



```
Enter data of 5 players
Enter PName TName BAvg for player-1 = ViratKkoli
India
59.40

Enter PName TName BAvg for player-2 = Dhoni
India
50.47

Enter PName TName BAvg for player-3 = TaslimArif
pakistan
62.62

Enter PName TName BAvg for player-4 = RohitSharma
India
48.91

Enter PName TName BAvg for player-5 = ShikharDhawan
India
44.91

After teamwise sorting... Player list is
ViratKkoli           India                59.40
Dhoni                India                50.47
RohitSharma          India                48.91
ShikharDhawan        India                44.91
TaslimArif           pakistan             62.62
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> 
```

3))

1)Problem statement:

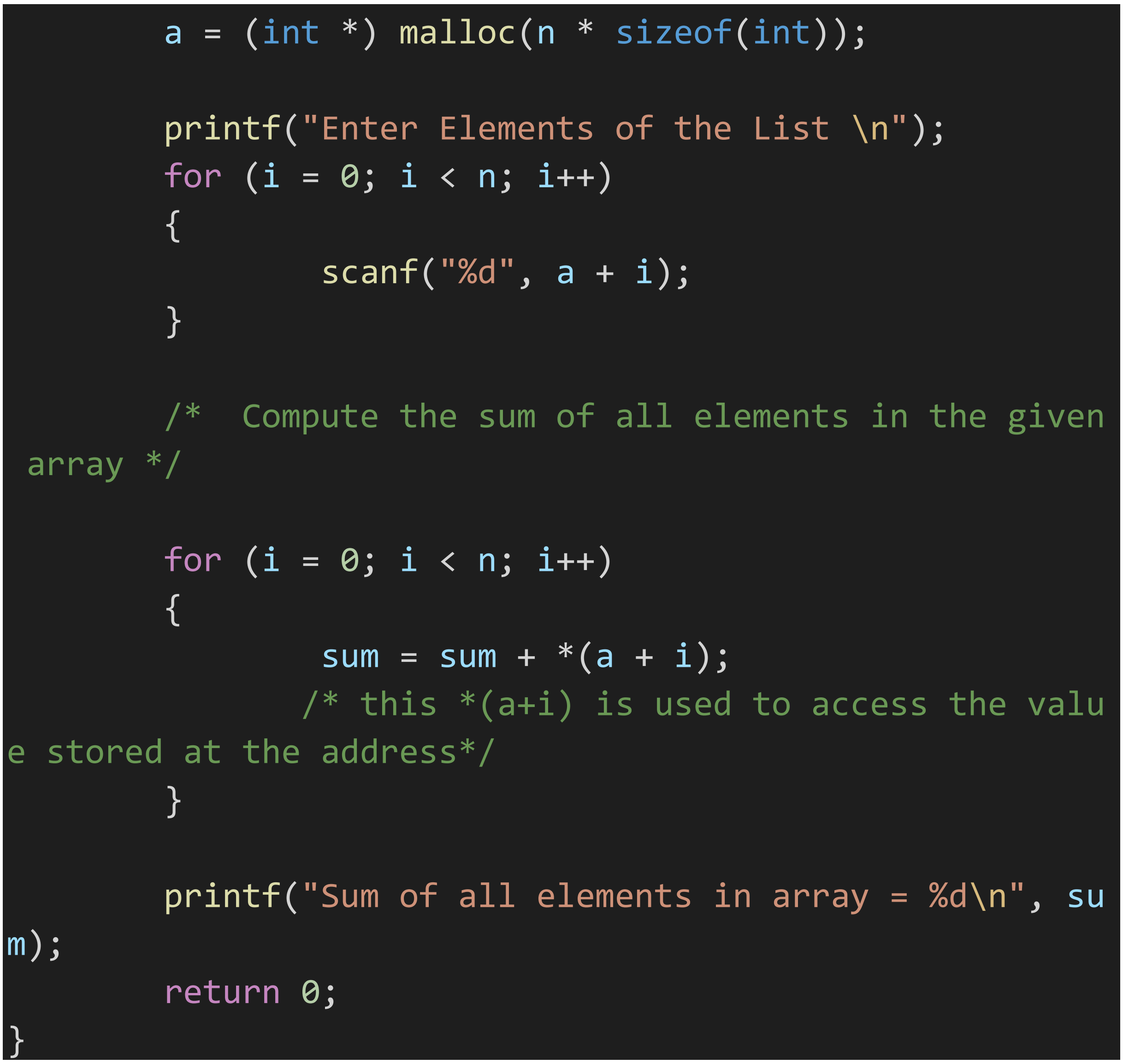
3. Write a program using pointers to compute the sum of all elements sorted in an array

2)Pseudo code:

```
Start
Declare i,n,sum=0
Declare  pointer a
Start  for loop(n)
Input the elements of array
Output
Start for loop(n)
   sum=sum+*(a+i)


End
```

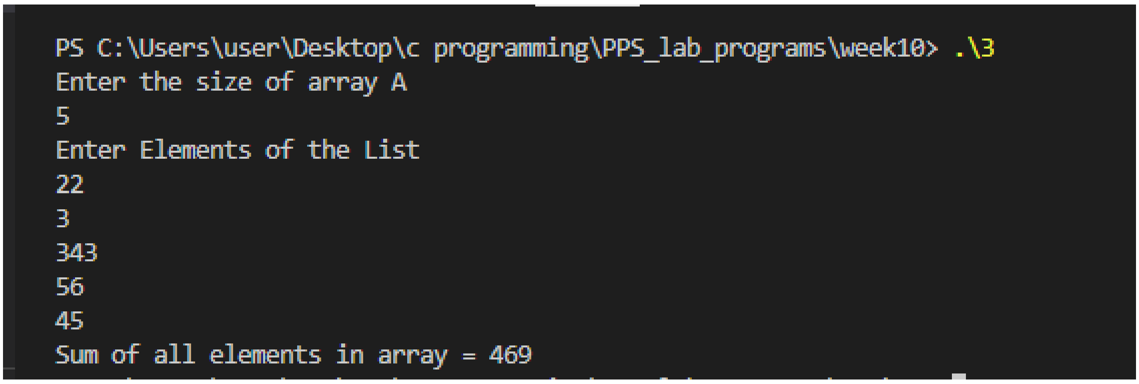
3)Executable C program :

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

int  main()
{
        int i, n, sum = 0;
        int *a;

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

```



```c
        a = (int *) malloc(n * sizeof(int));

        printf("Enter Elements of the List \n");
        for (i = 0; i < n; i++)
        {
                scanf("%d", a + i);
        }

        /*  Compute the sum of all elements in the given array */

        for (i = 0; i < n; i++)
        {
                sum = sum + *(a + i);
               /* this *(a+i) is used to access the value stored at the address*/
        }

        printf("Sum of all elements in array = %d\n", sum);
        return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> .\3
Enter the size of array A
5
Enter Elements of the List
22
3
343
56
45
Sum of all elements in array = 469
```

4))

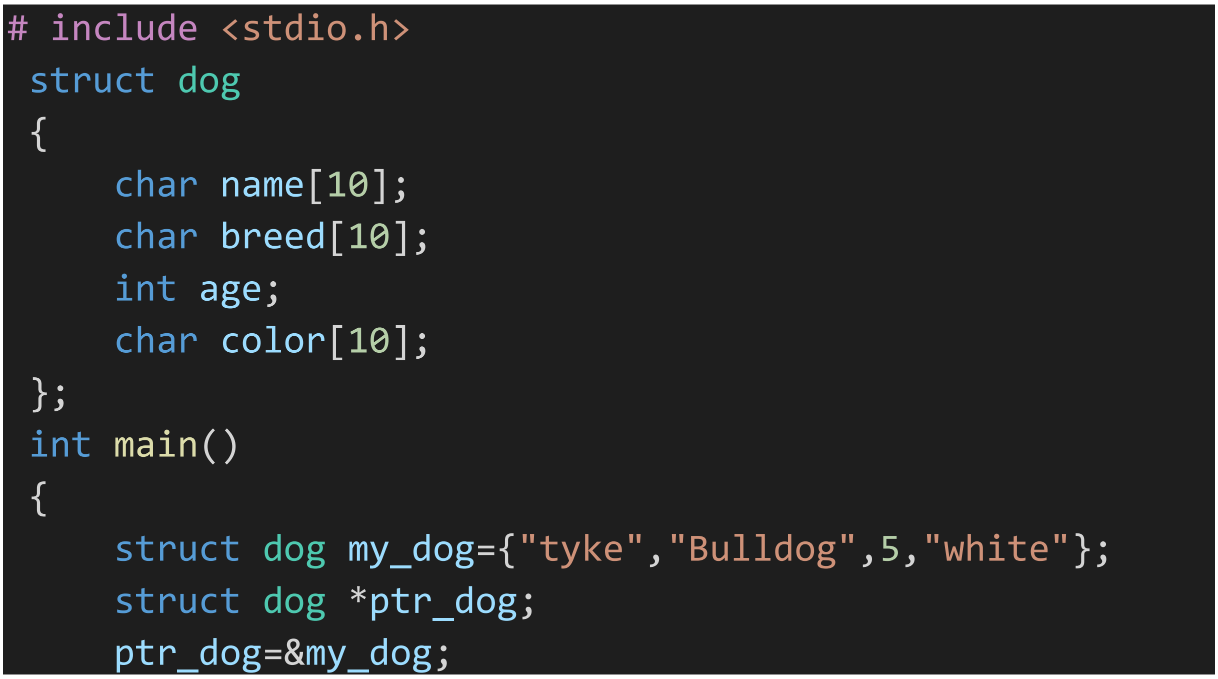
1)Problem statement:

4. Write a program to print the elements of a structure using pointers.

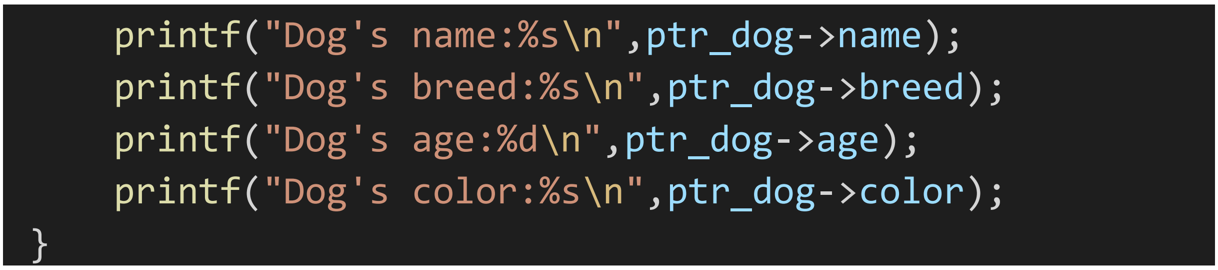
2)Pseudo code:

```
Start
Declare the structure
Input the data into the structures
Output using the pointer
End
```

3)Executable C program :

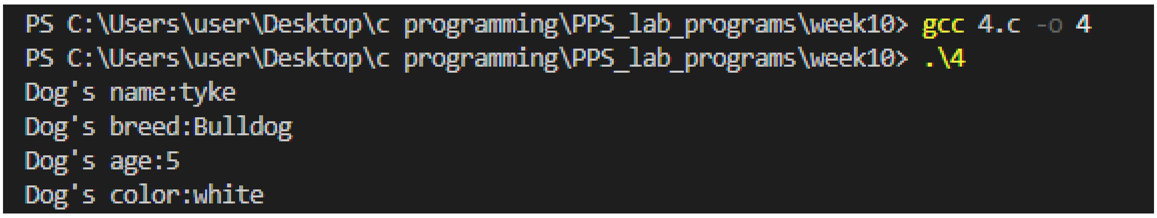


```
# include <stdio.h>
 struct dog
 {
     char name[10];
     char breed[10];
     int age;
     char color[10];
 };
 int main()
 {
     struct dog my_dog={"tyke","Bulldog",5,"white"};
     struct dog *ptr_dog;
     ptr_dog=&my_dog;
```



```
    printf("Dog's name:%s\n",ptr_dog->name);
    printf("Dog's breed:%s\n",ptr_dog->breed);
    printf("Dog's age:%d\n",ptr_dog->age);
    printf("Dog's color:%s\n",ptr_dog->color);
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> gcc 4.c -o 4
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week10> .\4
Dog's name:tyke
Dog's breed:Bulldog
Dog's age:5
Dog's color:white
```

Week 11

Tutorial 12: File handling:

Lab 12: File operations

1))

1)Problem statement:

1. Write a C program that creates an Employee text file? Records Are empid, empname, designation, qualification, salary, experience, Research work, address, city phone?

2)Pseudo code:

Start

Declare structure name,empid,salary

Declare file pointer fp

Fopen

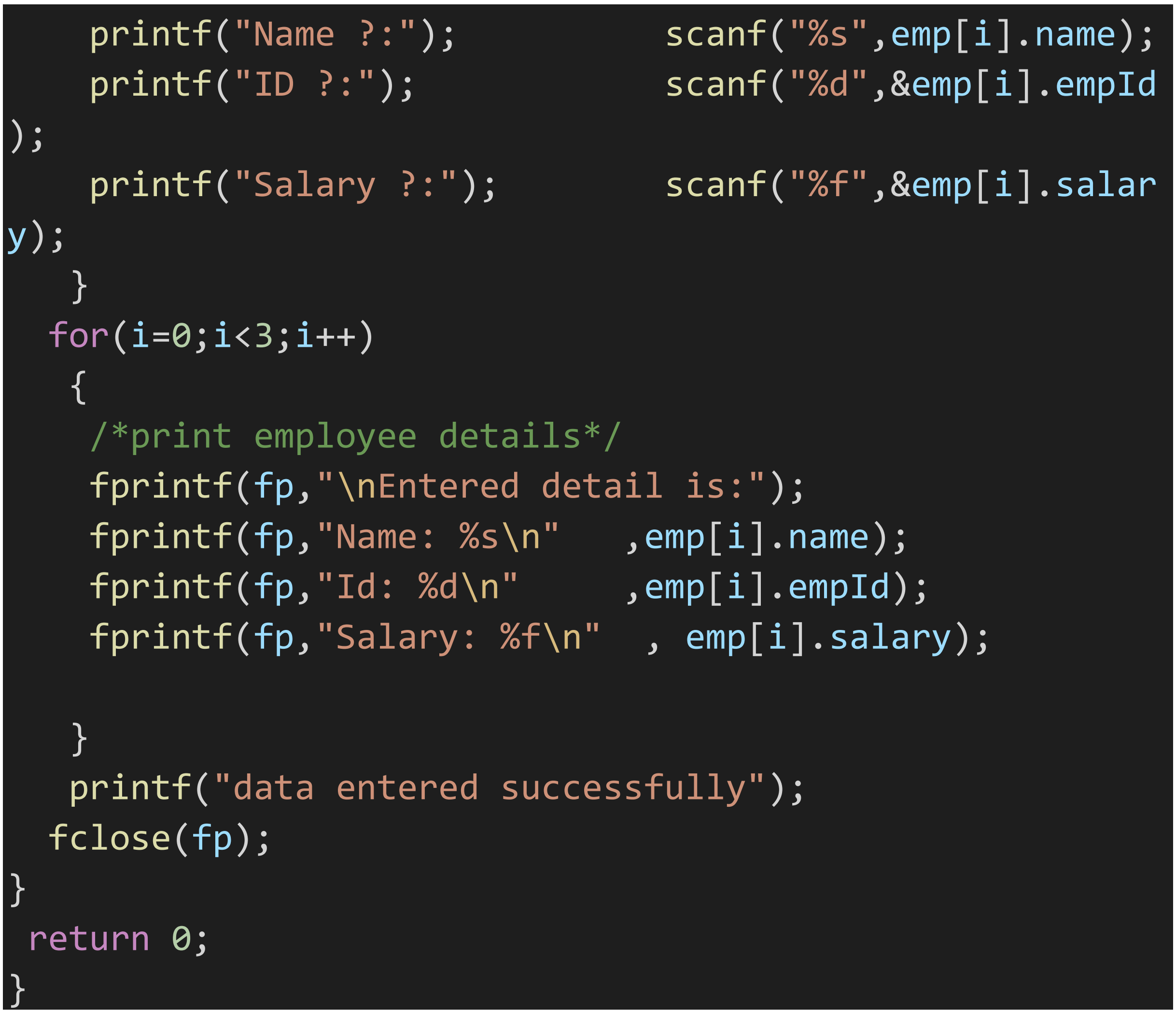
Write onto the files

Fclose

End

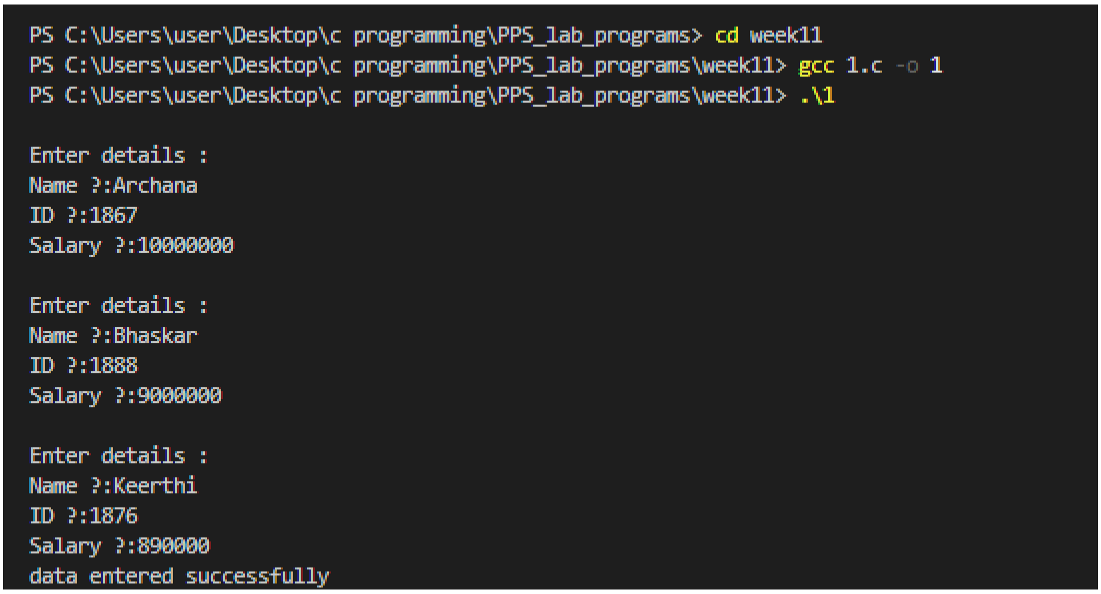
3)Executable C program :

```
#include <stdio.h>

/*structure declaration*/
struct employee{
    char    name[30];
    int     empId;
    float   salary;

};

int main()
{     int i;
    struct employee emp[3];
    FILE *fp;
    fp=fopen("emp.txt","w+");
    if(fp==NULL)
    {
        puts("cannot open the file");
    }
    else
{
   for(i=0;i<3;i++)
   {
    /*read employee details*/
    printf("\nEnter details :\n");
```

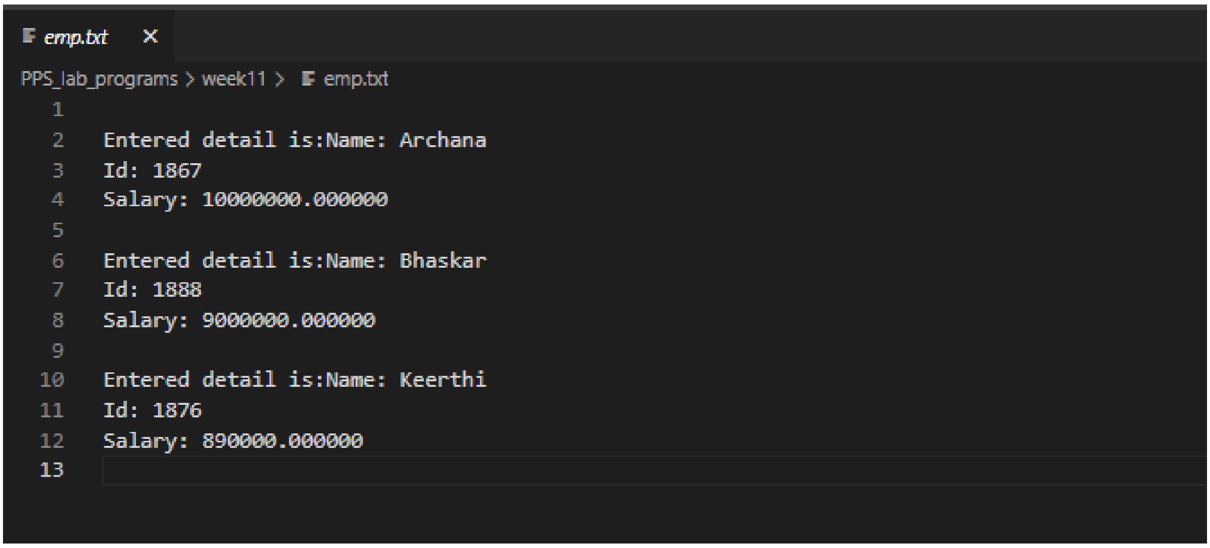


```
    printf("Name ?:");          scanf("%s",emp[i].name);
    printf("ID ?:");            scanf("%d",&emp[i].empId
);
    printf("Salary ?:");        scanf("%f",&emp[i].salar
y);
   }
  for(i=0;i<3;i++)
   {
    /*print employee details*/
    fprintf(fp,"\nEntered detail is:");
    fprintf(fp,"Name: %s\n"   ,emp[i].name);
    fprintf(fp,"Id: %d\n"     ,emp[i].empId);
    fprintf(fp,"Salary: %f\n"  , emp[i].salary);

   }
   printf("data entered successfully");
  fclose(fp);
}
 return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs> cd week11
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week11> gcc 1.c -o 1
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week11> .\1

Enter details :
Name ?:Archana
ID ?:1867
Salary ?:10000000

Enter details :
Name ?:Bhaskar
ID ?:1888
Salary ?:9000000

Enter details :
Name ?:Keerthi
ID ?:1876
Salary ?:890000
data entered successfully
```



```

Entered detail is:Name: Archana
Id: 1867
Salary: 10000000.000000

Entered detail is:Name: Bhaskar
Id: 1888
Salary: 9000000.000000

Entered detail is:Name: Keerthi
Id: 1876
Salary: 890000.000000

```

2))

1)Problem statement:

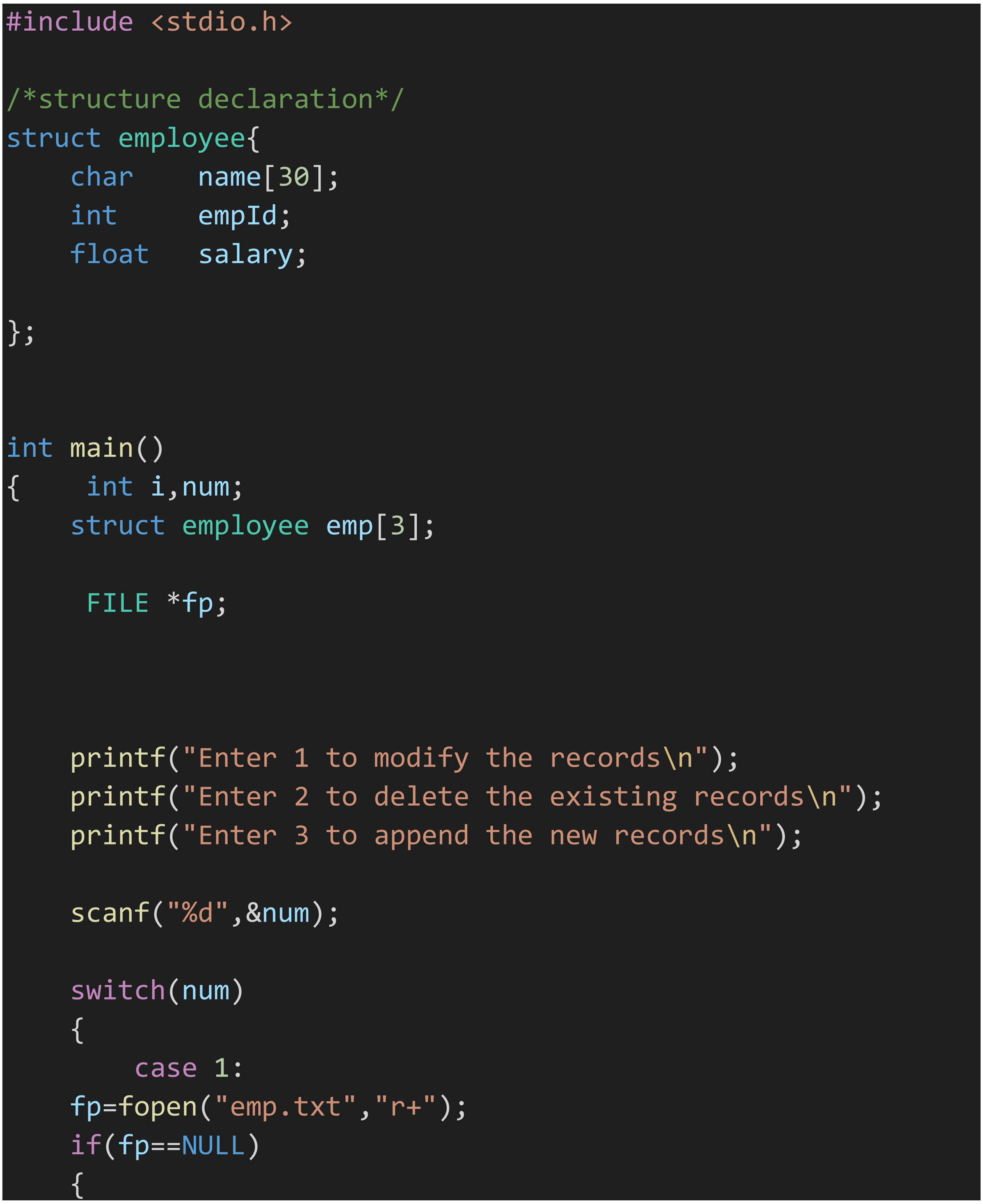
2. Write a C program that manipulates the above text file. The program must implements the operation to modify a record, delete a record and append new records

2)Pseudo code:

```
Start
Declare structure name,empid,salary
Declare file pointer fp
Write swith cases for modify,delete,append
Fopen
Write onto the files
Fclose
```

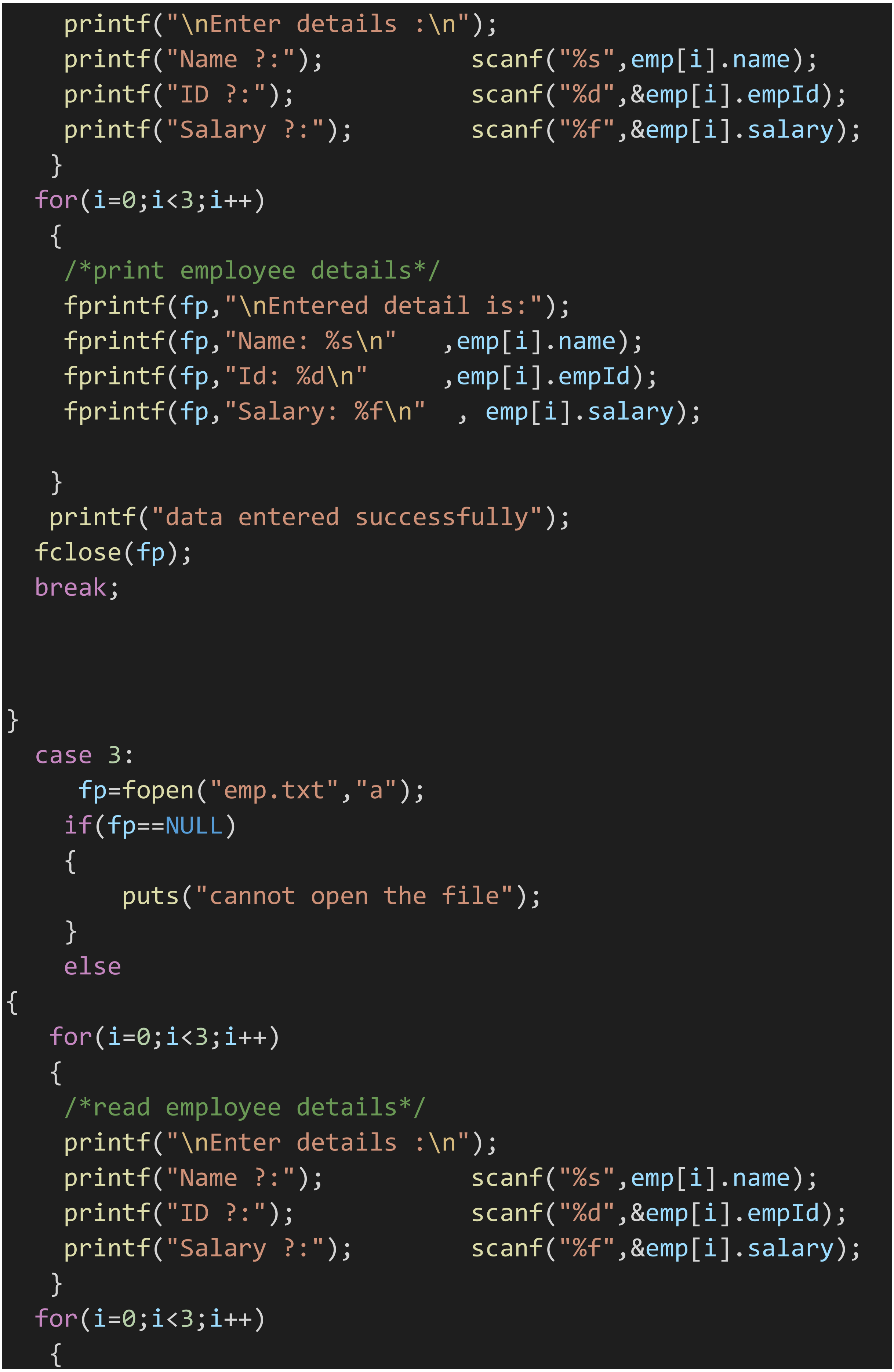
End

3)Executable C program :

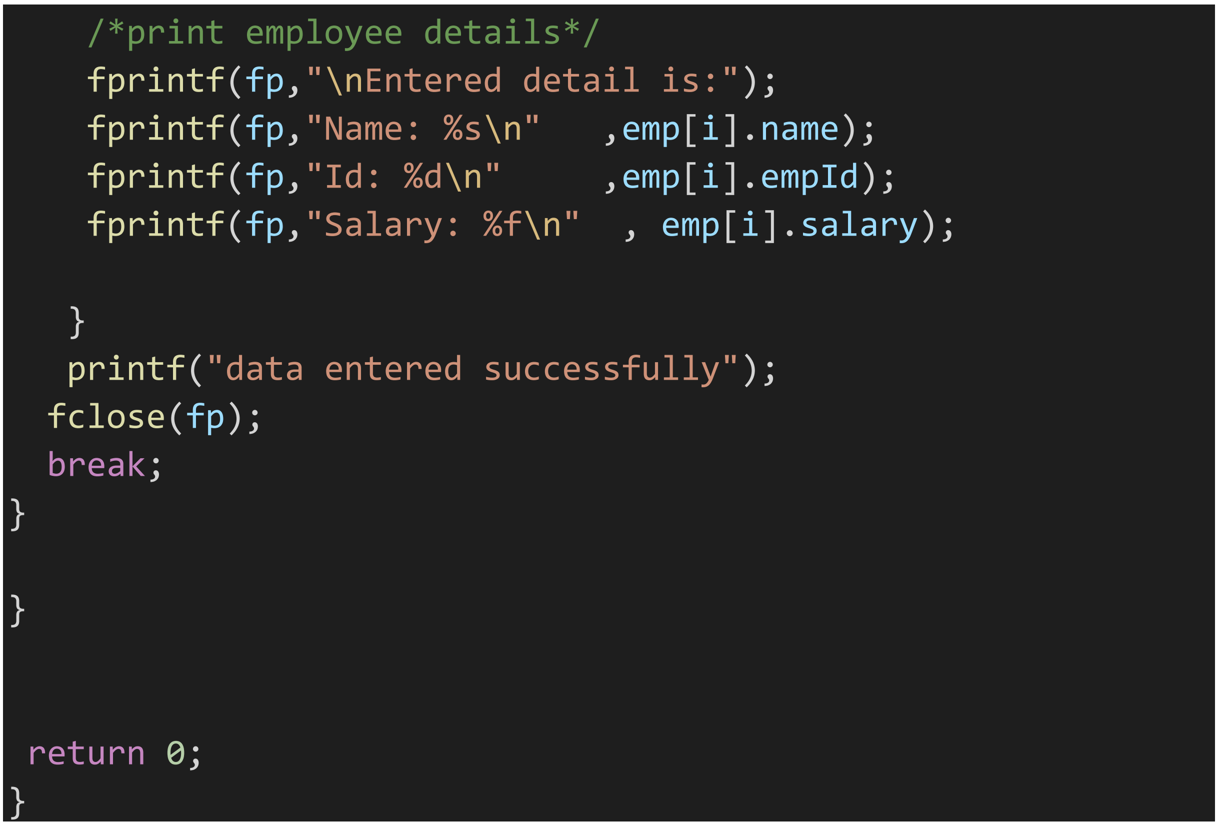


```
#include <stdio.h>

/*structure declaration*/
struct employee{
    char    name[30];
    int     empId;
    float   salary;

};


int main()
{    int i,num;
    struct employee emp[3];

     FILE *fp;



    printf("Enter 1 to modify the records\n");
    printf("Enter 2 to delete the existing records\n");
    printf("Enter 3 to append the new records\n");

    scanf("%d",&num);

    switch(num)
    {
        case 1:
    fp=fopen("emp.txt","r+");
    if(fp==NULL)
    {
```

```
        puts("cannot open the file");
    }
    else
{
   for(i=0;i<3;i++)
   {
    /*read employee details*/
    printf("\nEnter details :\n");
    printf("Name ?:");          scanf("%s",emp[i].name);
    printf("ID ?:");            scanf("%d",&emp[i].empId);
    printf("Salary ?:");        scanf("%f",&emp[i].salary);
   }
  for(i=0;i<3;i++)
   {
    /*print employee details*/
    fprintf(fp,"\nEntered detail is:");
    fprintf(fp,"Name: %s\n"   ,emp[i].name);
    fprintf(fp,"Id: %d\n"     ,emp[i].empId);
    fprintf(fp,"Salary: %f\n"  , emp[i].salary);

   }
   printf("data entered successfully");
  fclose(fp);
  break;



}
     case 2:
        fp=fopen("emp.txt","w");
    if(fp==NULL)
    {
        puts("cannot open the file");
    }
    else
{
   for(i=0;i<3;i++)
   {
    /*read employee details*/
```

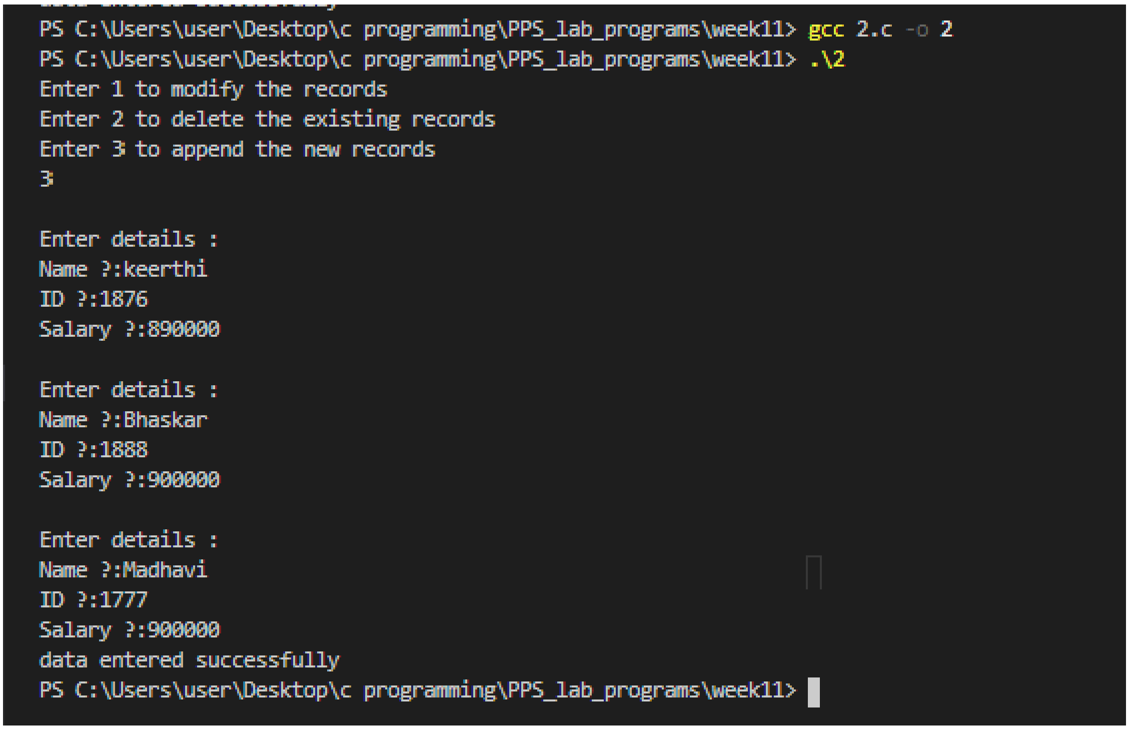


```
    printf("\nEnter details :\n");
    printf("Name ?:");          scanf("%s",emp[i].name);
    printf("ID ?:");            scanf("%d",&emp[i].empId);
    printf("Salary ?:");        scanf("%f",&emp[i].salary);
   }
  for(i=0;i<3;i++)
   {
    /*print employee details*/
    fprintf(fp,"\nEntered detail is:");
    fprintf(fp,"Name: %s\n"   ,emp[i].name);
    fprintf(fp,"Id: %d\n"     ,emp[i].empId);
    fprintf(fp,"Salary: %f\n"  , emp[i].salary);

   }
   printf("data entered successfully");
  fclose(fp);
  break;



}
  case 3:
     fp=fopen("emp.txt","a");
    if(fp==NULL)
    {
        puts("cannot open the file");
    }
    else
{
   for(i=0;i<3;i++)
   {
    /*read employee details*/
    printf("\nEnter details :\n");
    printf("Name ?:");          scanf("%s",emp[i].name);
    printf("ID ?:");            scanf("%d",&emp[i].empId);
    printf("Salary ?:");        scanf("%f",&emp[i].salary);
   }
  for(i=0;i<3;i++)
   {
```

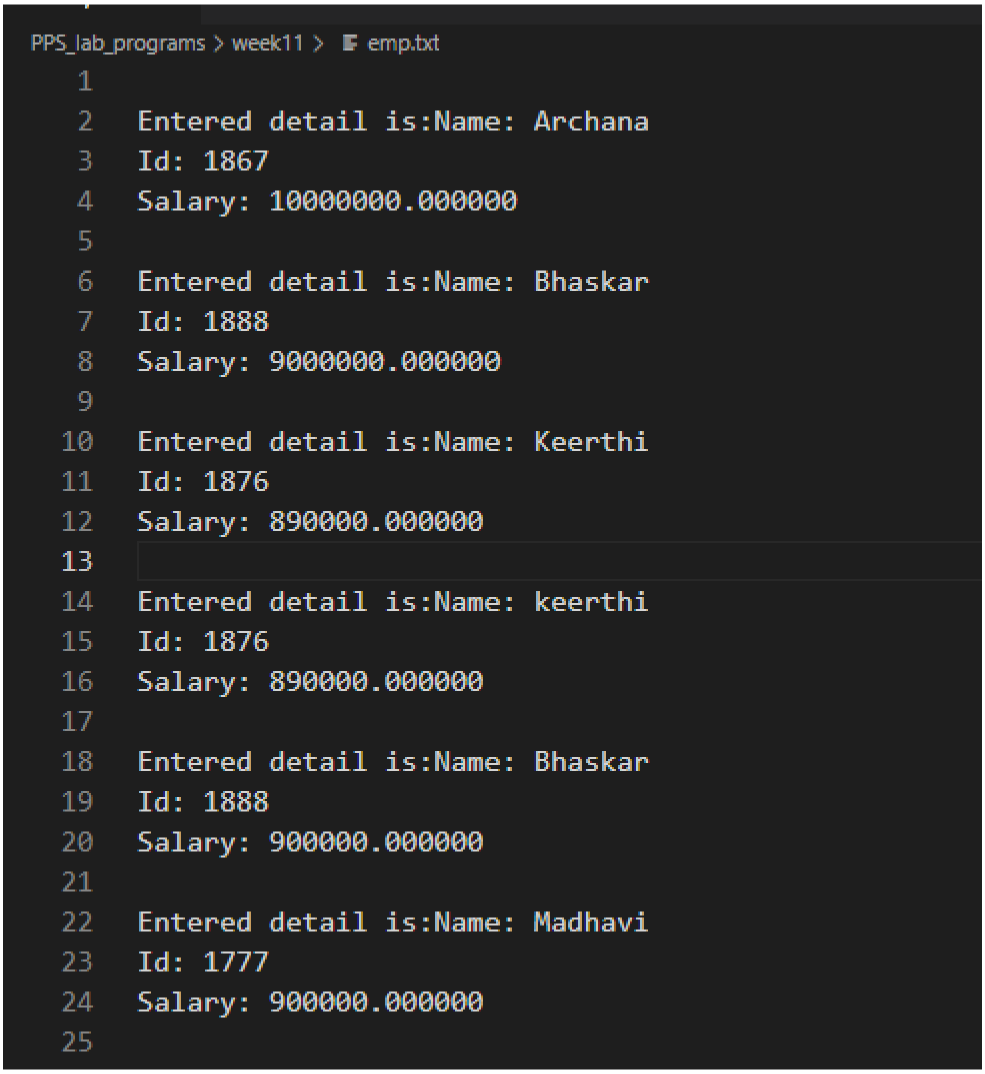


```
    /*print employee details*/
    fprintf(fp,"\nEntered detail is:");
    fprintf(fp,"Name: %s\n"    ,emp[i].name);
    fprintf(fp,"Id: %d\n"      ,emp[i].empId);
    fprintf(fp,"Salary: %f\n"   , emp[i].salary);

   }
   printf("data entered successfully");
  fclose(fp);
  break;
}

}


 return 0;
}
```

4)Output:



```
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week11> gcc 2.c -o 2
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week11> .\2
Enter 1 to modify the records
Enter 2 to delete the existing records
Enter 3 to append the new records
3

Enter details :
Name ?:keerthi
ID ?:1876
Salary ?:890000

Enter details :
Name ?:Bhaskar
ID ?:1888
Salary ?:900000

Enter details :
Name ?:Madhavi
ID ?:1777
Salary ?:900000
data entered successfully
PS C:\Users\user\Desktop\c programming\PPS_lab_programs\week11>
```



```

Entered detail is:Name: Archana
Id: 1867
Salary: 10000000.000000

Entered detail is:Name: Bhaskar
Id: 1888
Salary: 9000000.000000

Entered detail is:Name: Keerthi
Id: 1876
Salary: 890000.000000

Entered detail is:Name: keerthi
Id: 1876
Salary: 890000.000000

Entered detail is:Name: Bhaskar
Id: 1888
Salary: 900000.000000

Entered detail is:Name: Madhavi
Id: 1777
Salary: 900000.000000

```

-----End-----