Lab manual Programs (week 1 - 6)

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 using the formula:
//Area = PI * r2
```

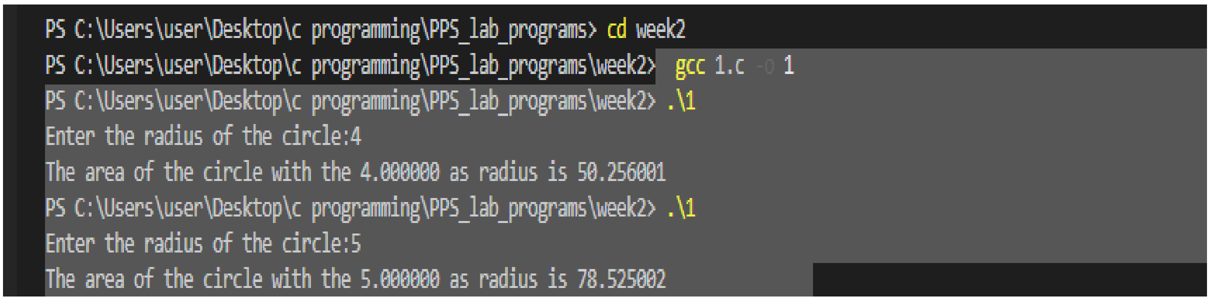
2)Pseudo code:

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

3)Executable C program :

```c
# 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
```

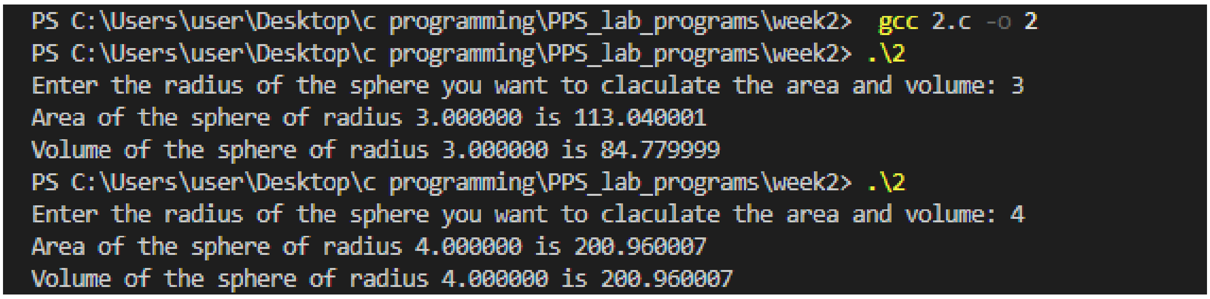
2)Psuedo code:

```
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.

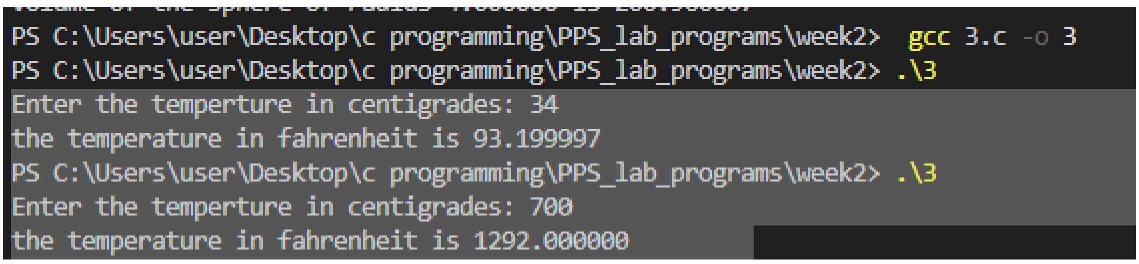
2)Psuedo code:

```
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.
```

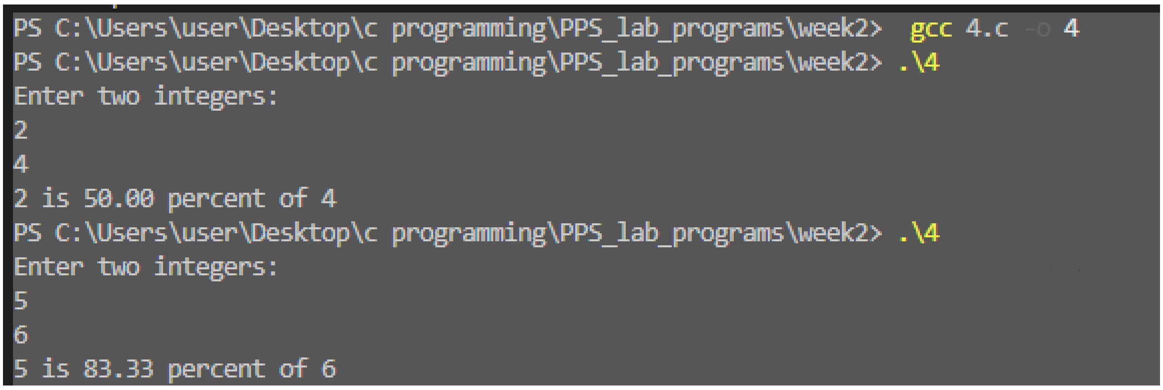
2)Psuedo code:

```
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:

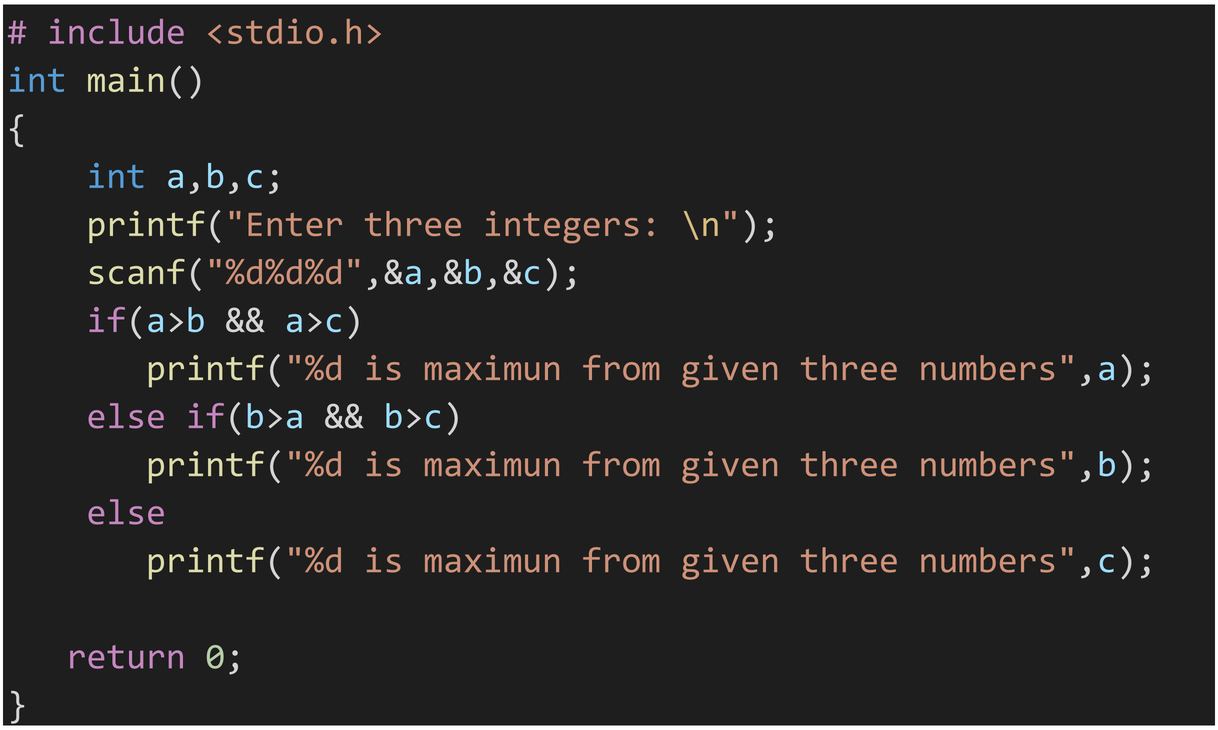
Write a C program to find the maximum from given three nos.

2)Psuedo code:

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

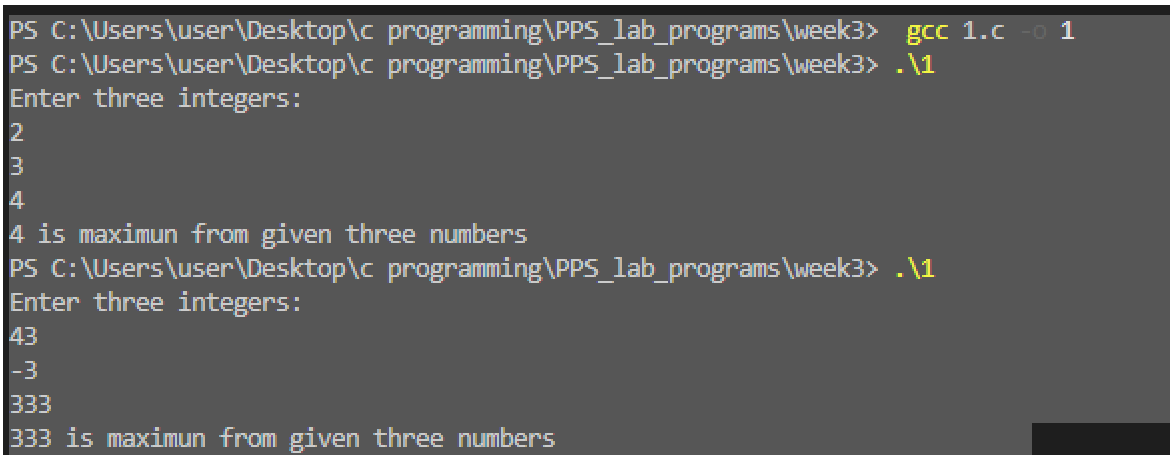
```
Else if b>a && b>c
 Output b is maximum
Else
  Output c is maximum
End
```

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.
```

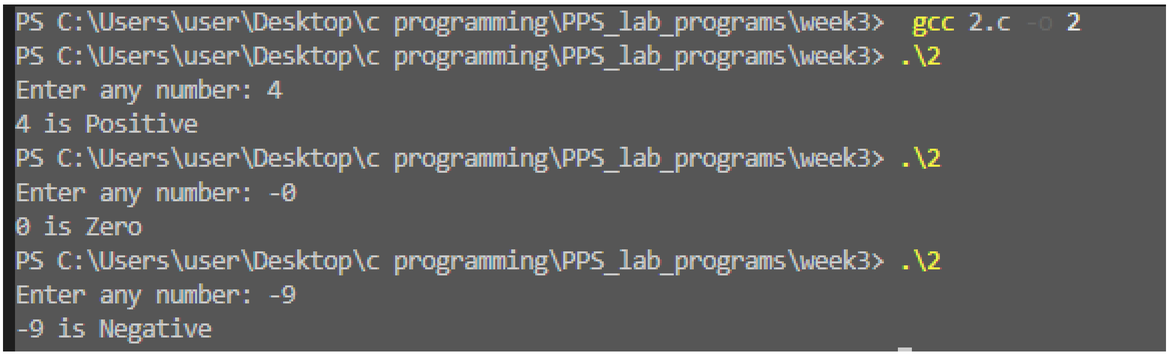
2)Psuedo code:

```
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:

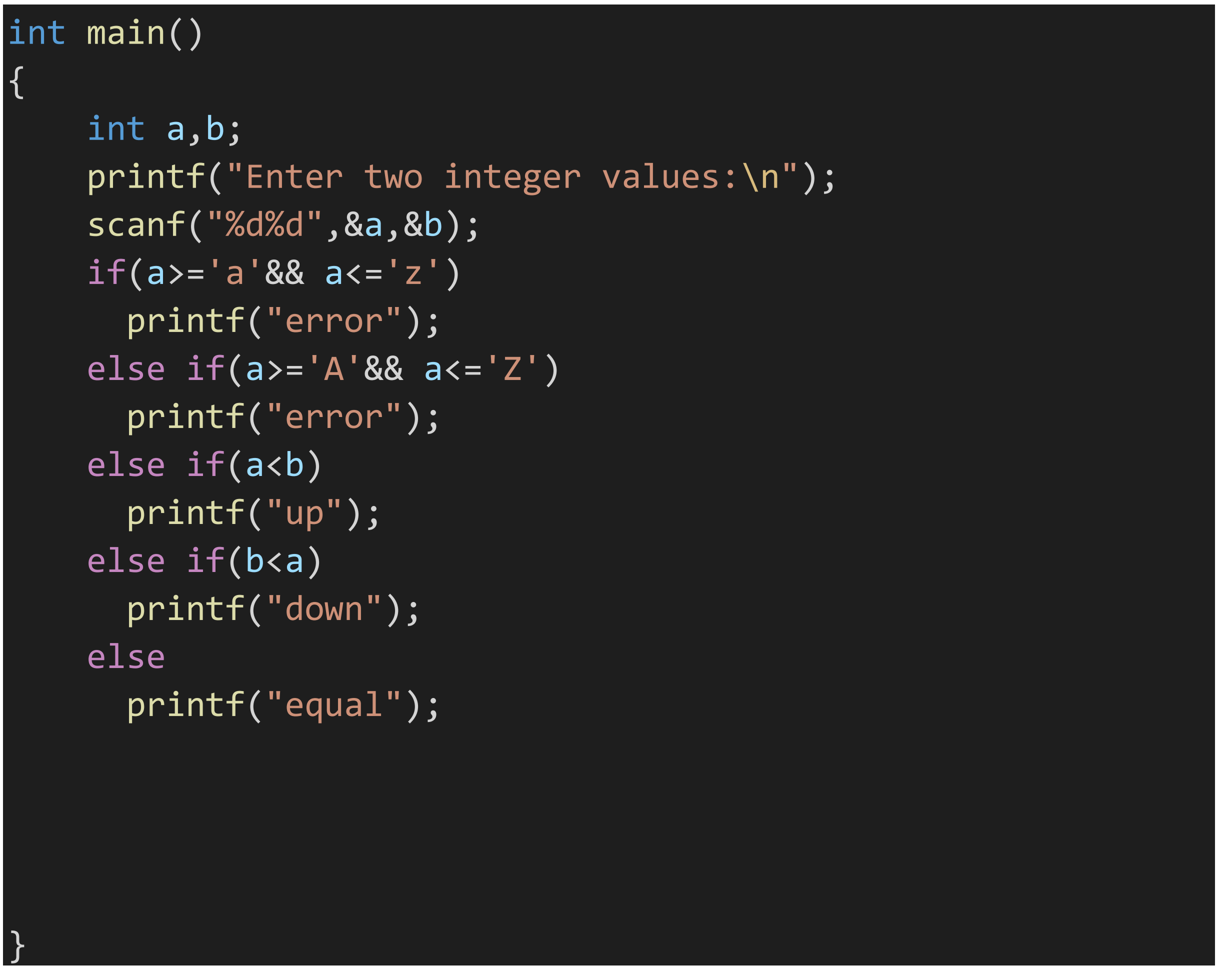
Write a program which reads two integer values. If the first islesser print the message “up”. If the second is lesser, print themessage “down” if they are equal, print the message “equal” ifthere is an error reading the data, print a message containing the word “Error”

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
```

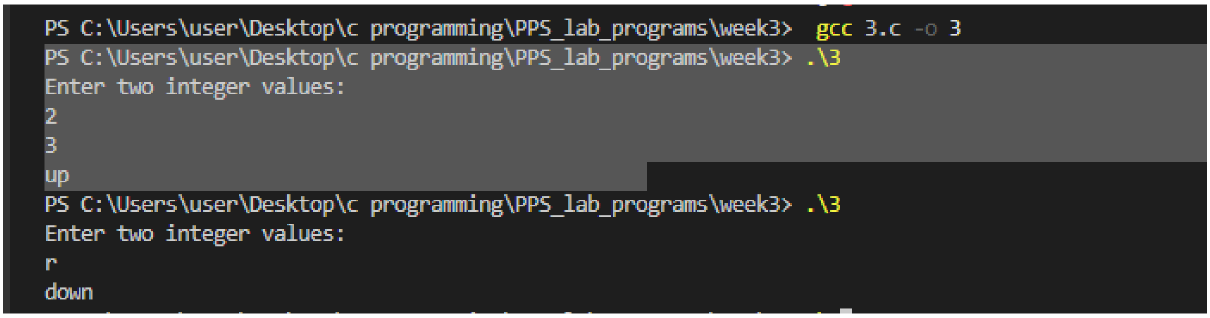
```
Else if b<a
  Output down
Else
Equal
End
```

3)Executable C program :



```
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:

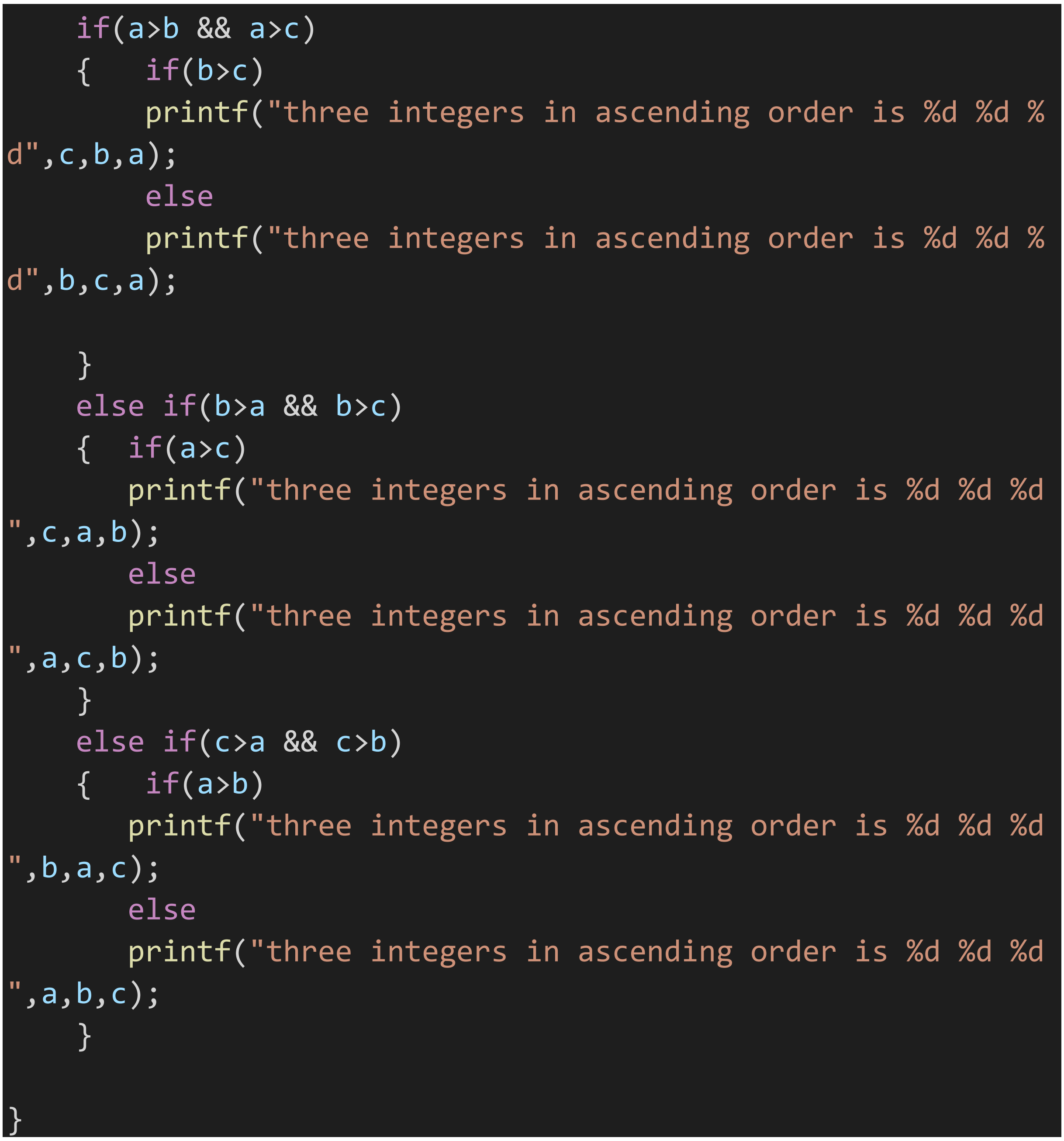
```
 Write a C program that prints the given three integers in
Ascending order using if - else.
```

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
```

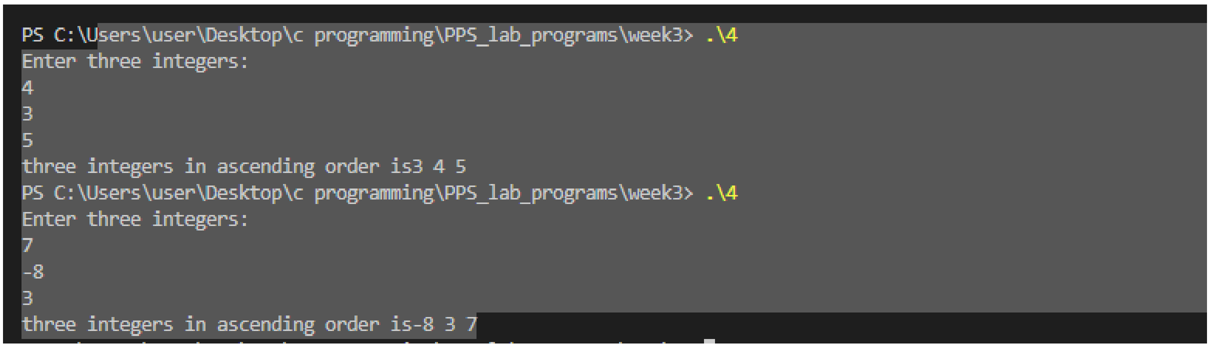
3)Executable C program :

```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:

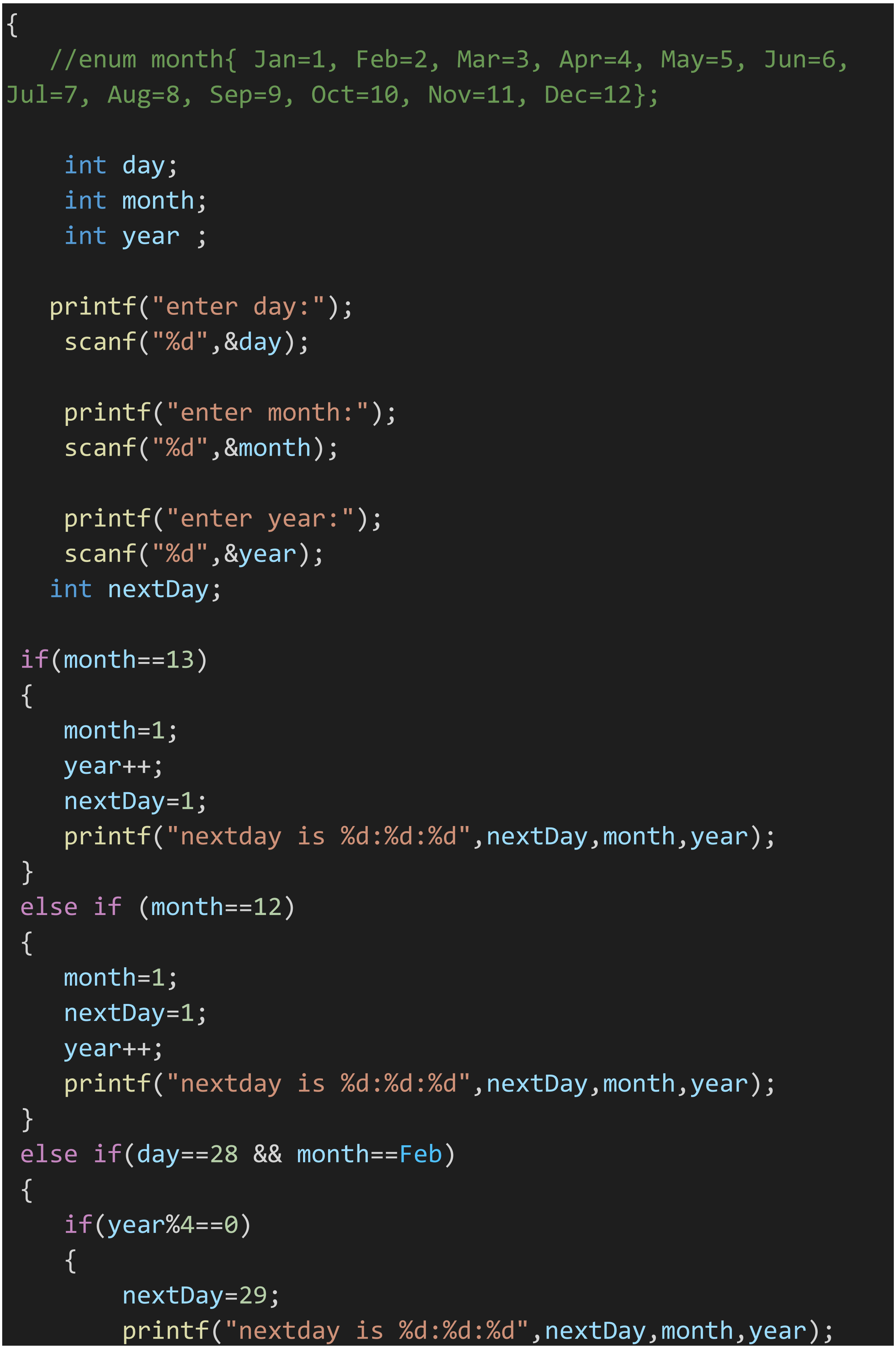
```
Given as input three integers representing a date as day,
month,year, print the number day, month and year for the nex
t day's date.
Typical input: “28 2 1992” Typical output: “Date following
28:02:1992 is 29:02:1992
```

2)Psuedo code:

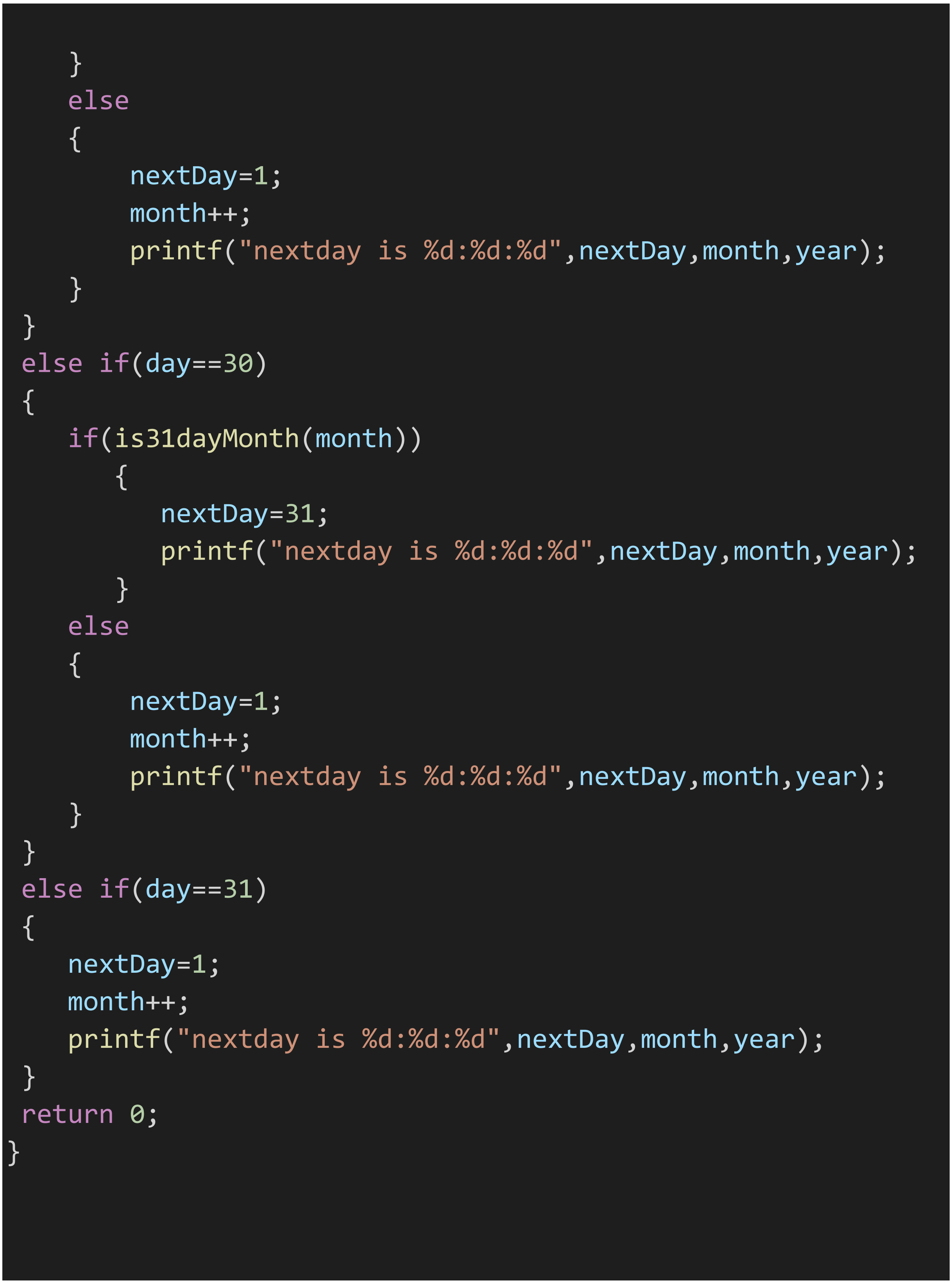
```
Start
Declare enum jan=1,feb,mar,apr,may,Jun,jul,aug,sep,oct,nov,dec
Declare _Bool is 31dayMonth()
                If   m==jan or m==mar or m==may or m==jul or m==aug or m==oct
                        or m==dec
                    return 1
                else
                    return 0
Declare 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()
```

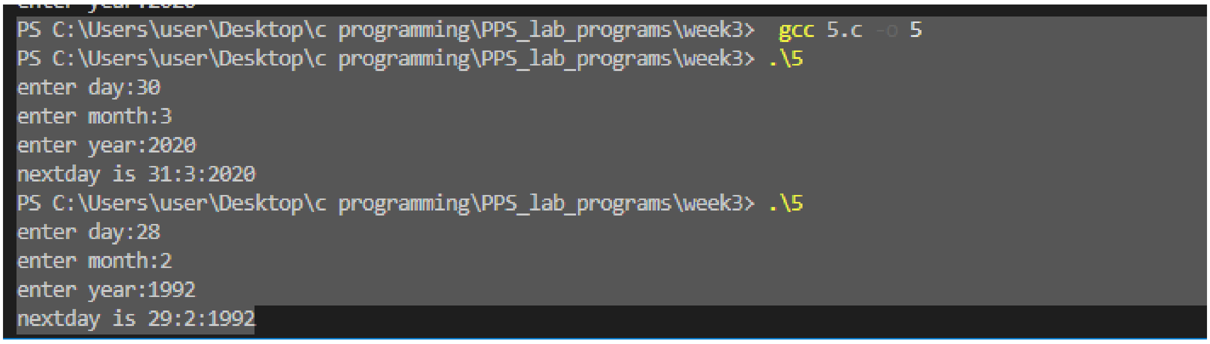


```
{
   //enum month{ Jan=1, Feb=2, Mar=3, Apr=4, May=5, Jun=6,
Jul=7, Aug=8, Sep=9, Oct=10, Nov=11, Dec=12};

    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:



```
enter year:2020
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:

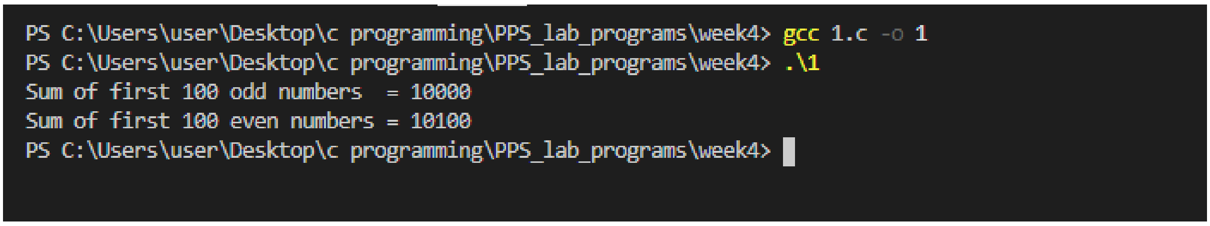
```
Start
Declare i,odd_sum,even_sum,count
Start for loop
 For i=1;i<200;i++
    If  i%2==0
       even_sum=even__sum+i
       count++
    else
       odd_sum=odd_sum+i
        count++
output
   sum of first 100 odd numbers=odd_sum
```

  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))

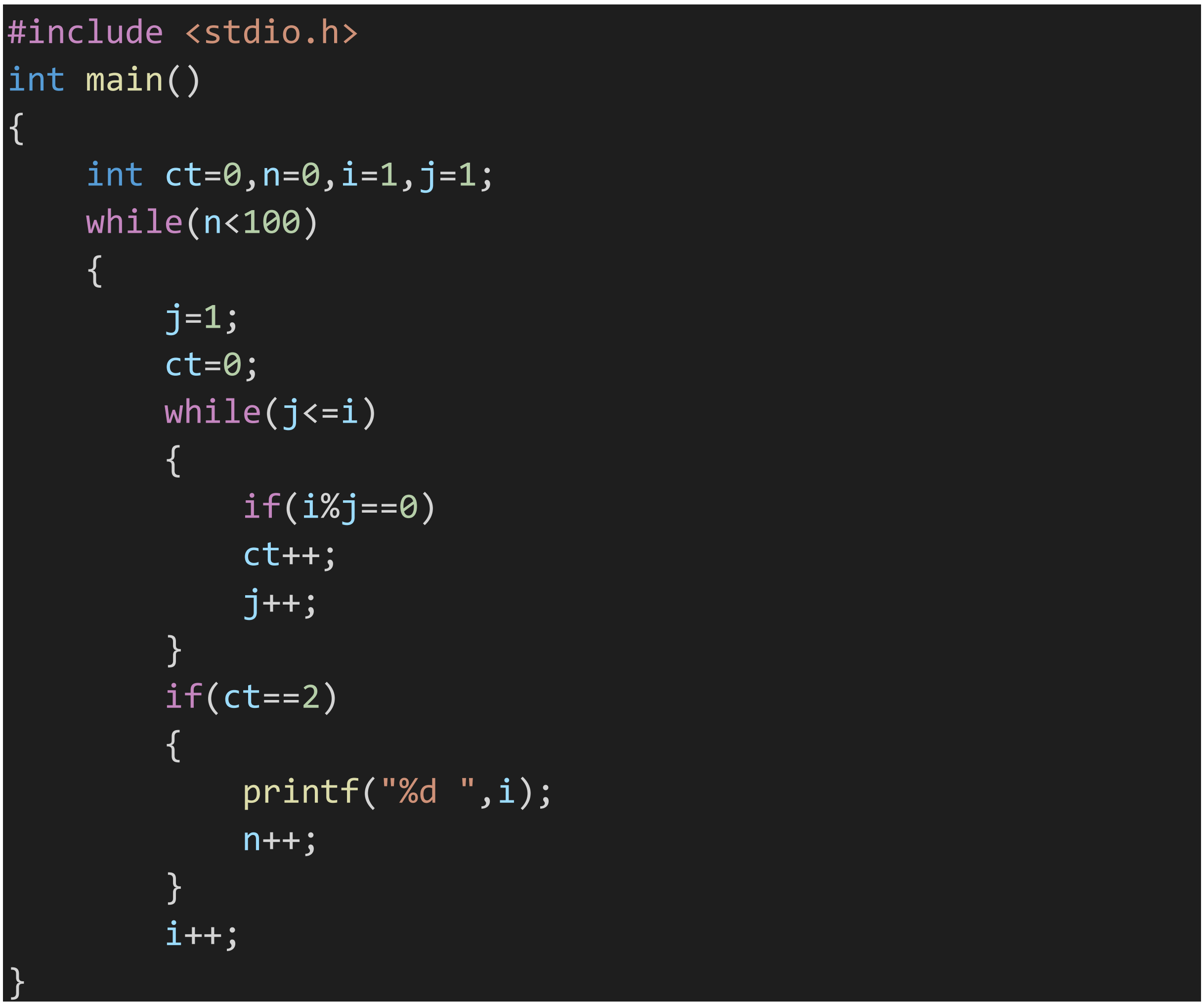
1)Problem statement:

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

2)Psuedo code:

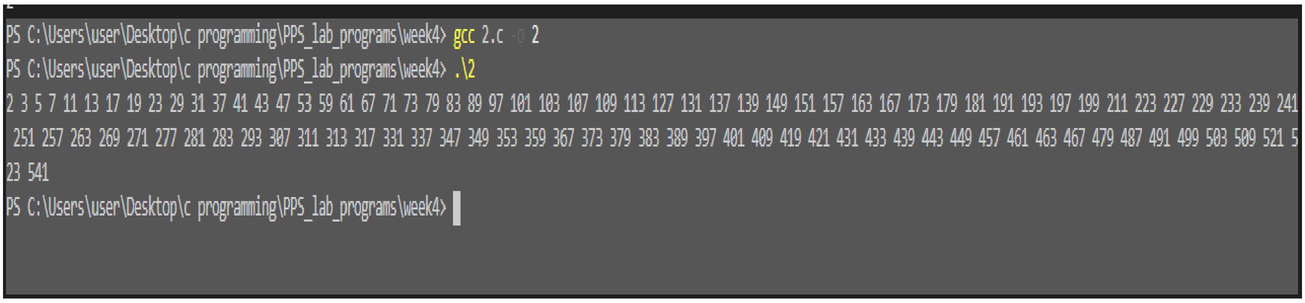
```
Start
Declare ct,n,i,j
 While n<100
    j=1
   ct=0
   while j<=i
      if  i%j==0
      ct++
      j++
  if ct==2
  output i
  n++
i++
End
```

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
```

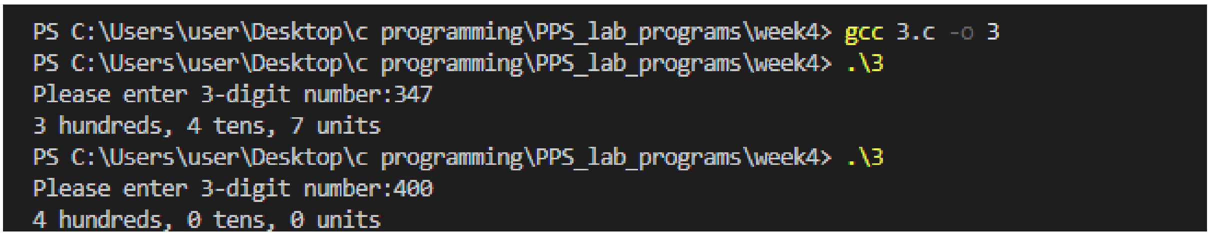
2)Psuedo code:

```
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:

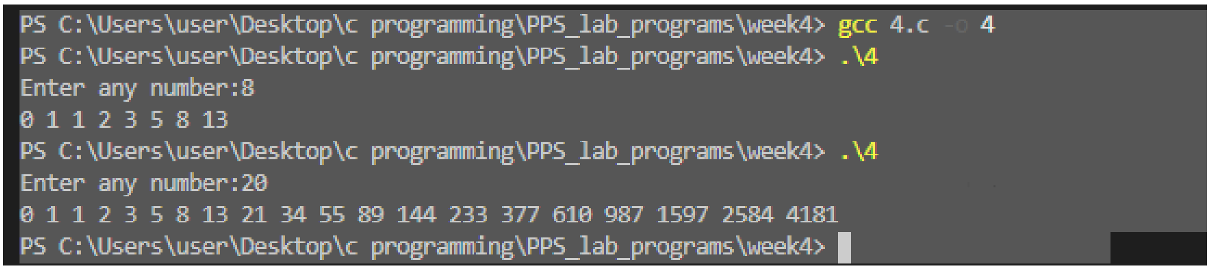
```
Start
Declare n,a=0,b=1,c,i
Input n
Start for loop
For i=1;i<=n;i++
 Output a
 c=a+b
 a=b
b=c
```

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:

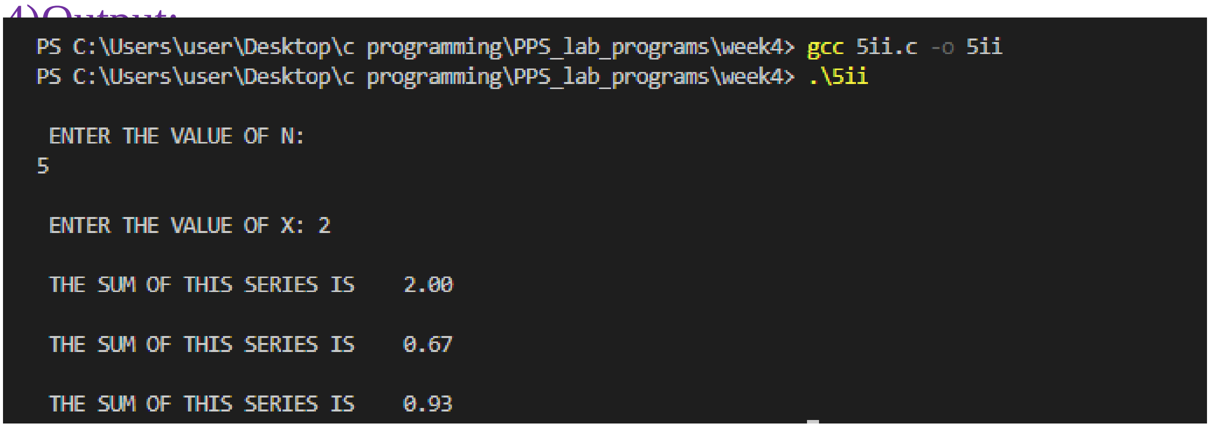
```
Start
Declare n,f_coun
Declare sum,power,fact
For n=0,power=0; power<=10;n++,power=power+2
```

```
  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 :

```c
#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:

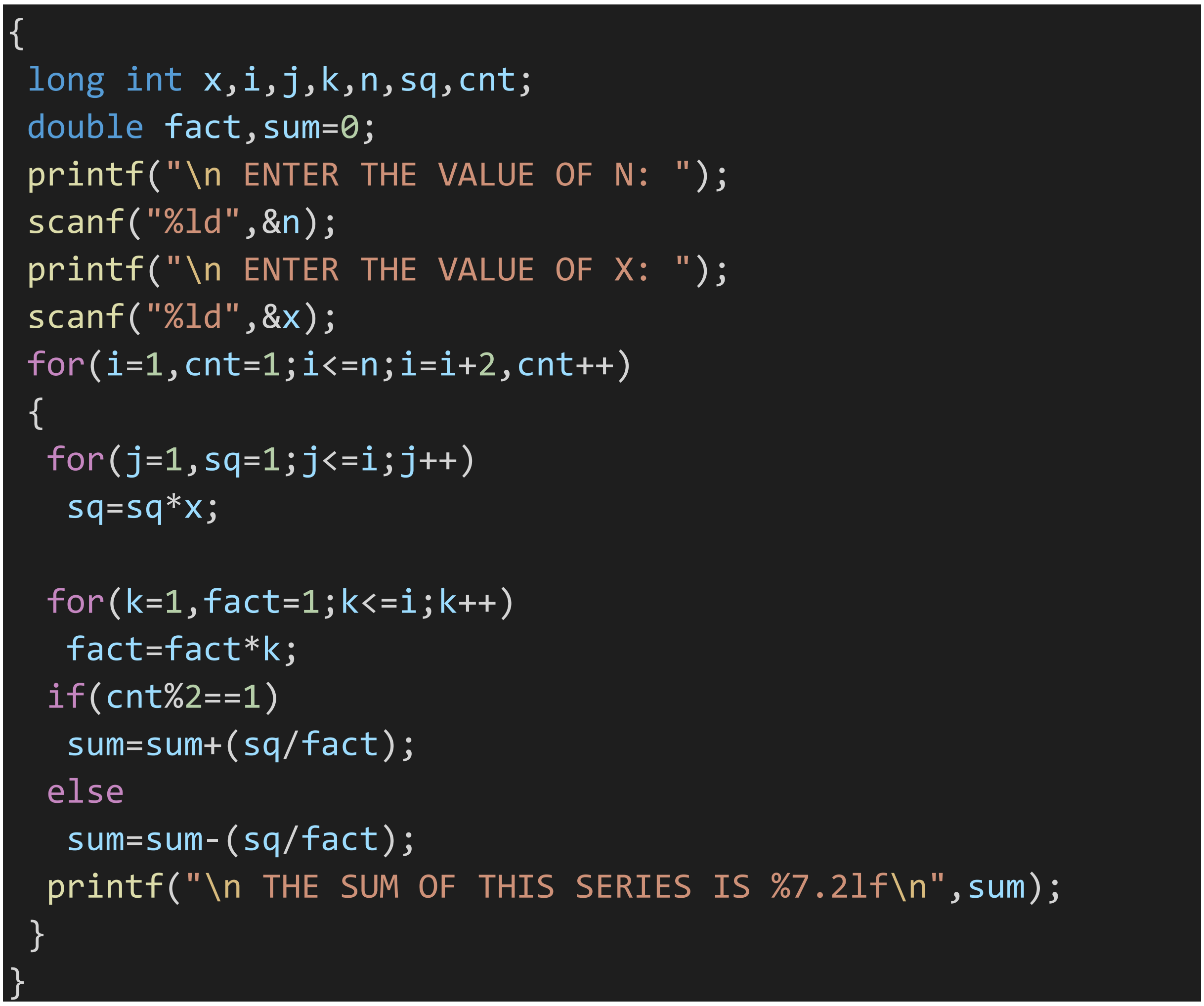
```
ii.sum=x-x3/3!+x5/5!.........................,
```

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
```

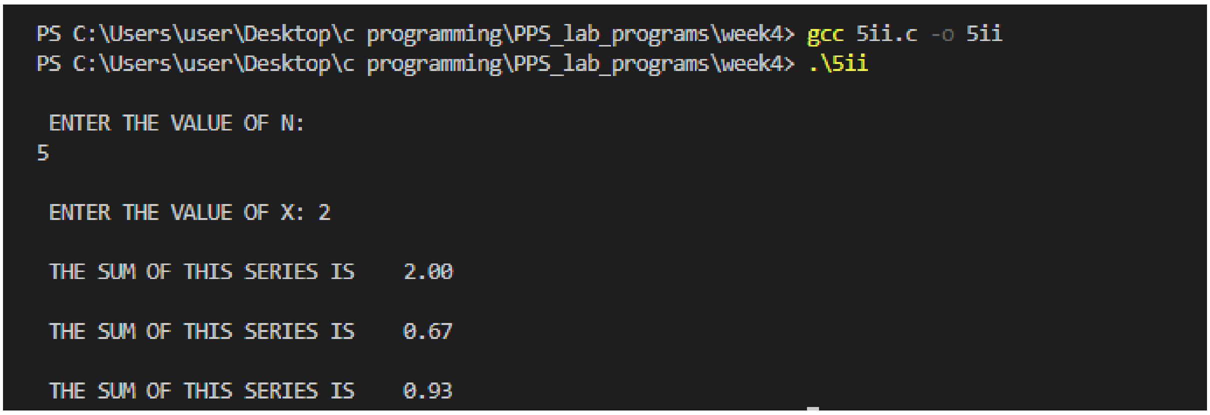
3)Executable C program :

```
#include<stdio.h>

int main()
```



```
{
 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!..............,
```

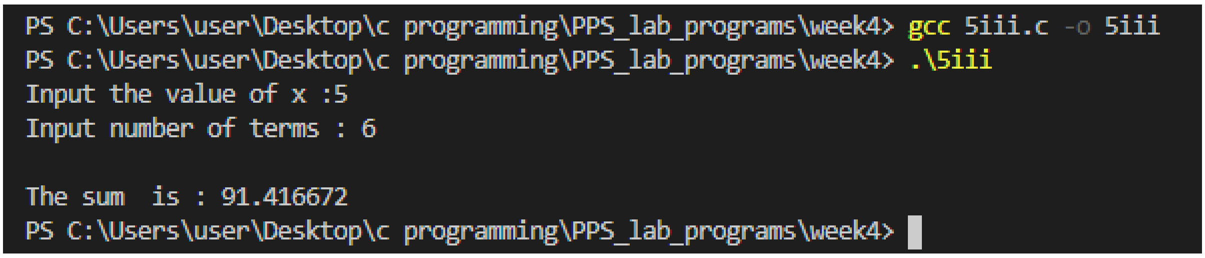
2)Psuedo code:

```
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 :

```
#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.
```

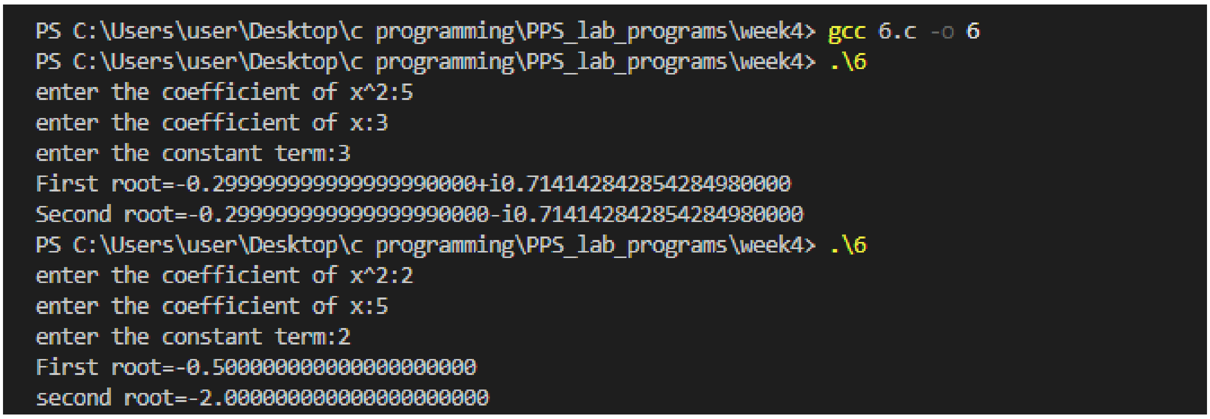
2)Psuedo code:

```
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 :

```
# 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
```

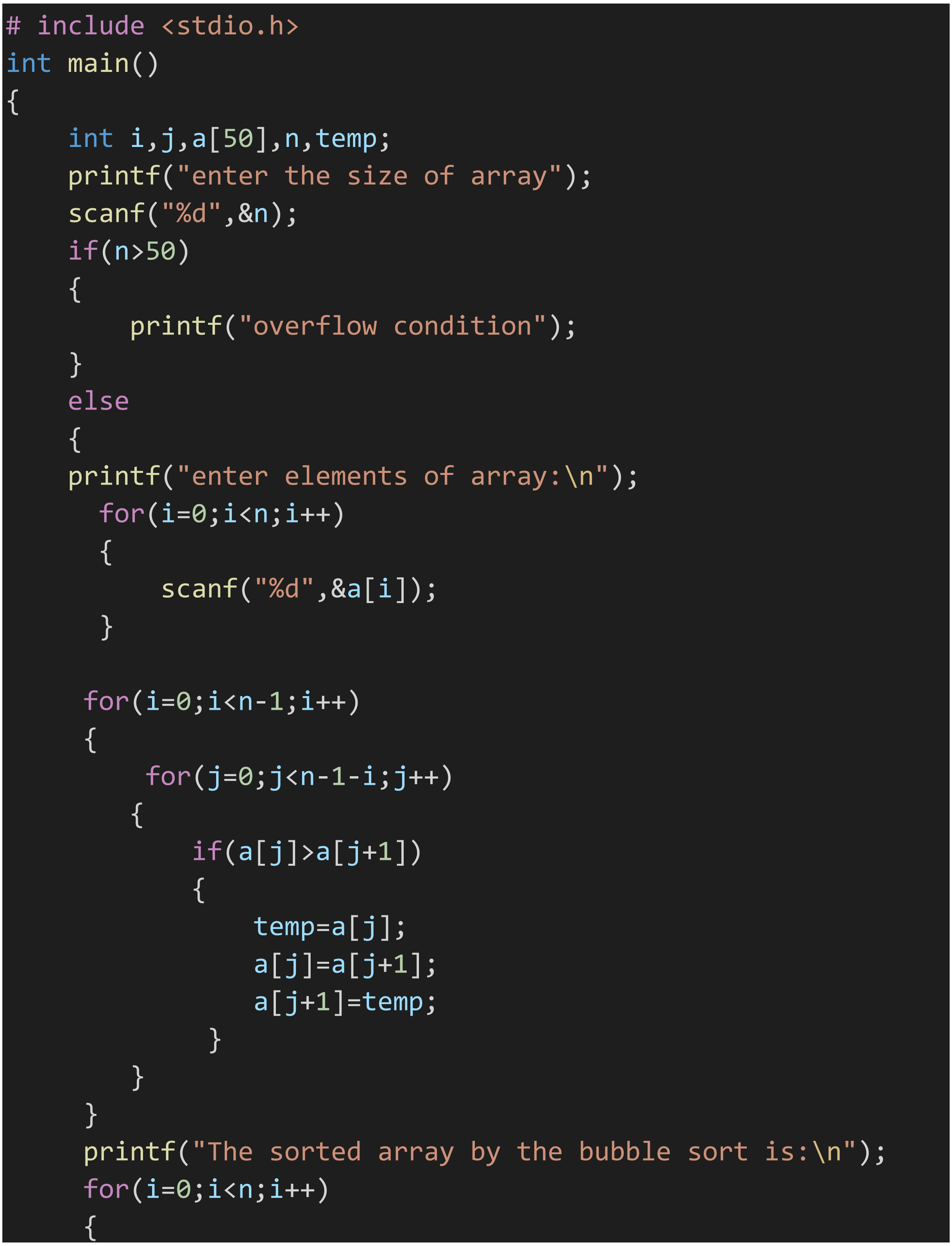
2)Psuedo code:

```
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

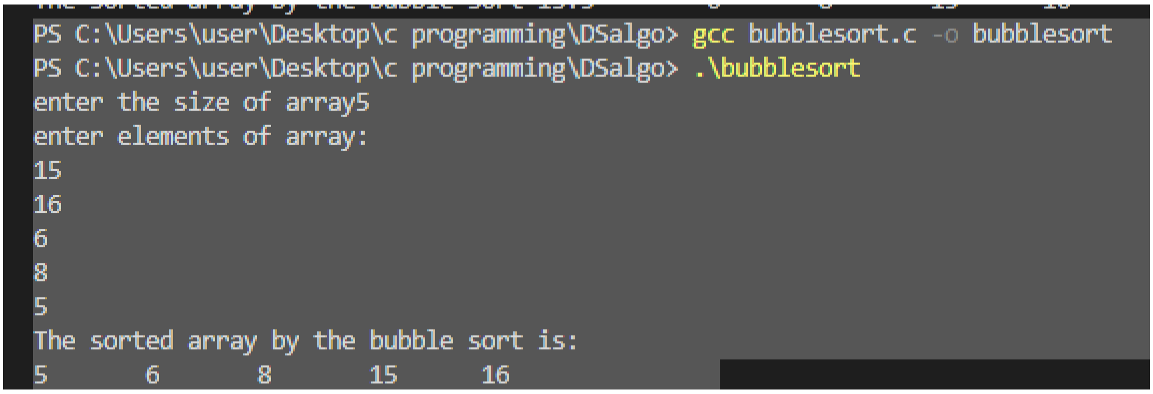
End

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
```

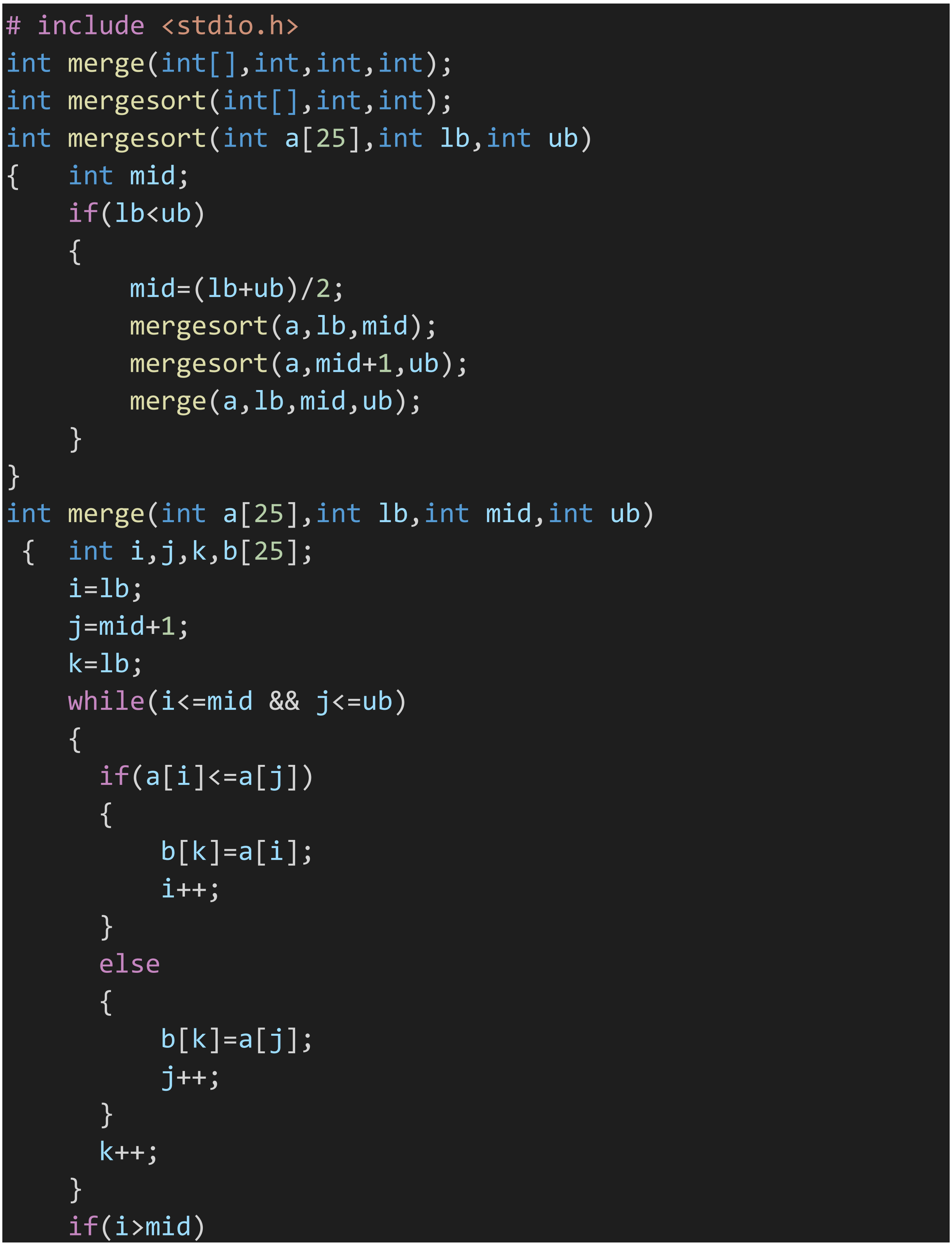
return mid=(left+right)/2

MergeSort(arr,left,mid)

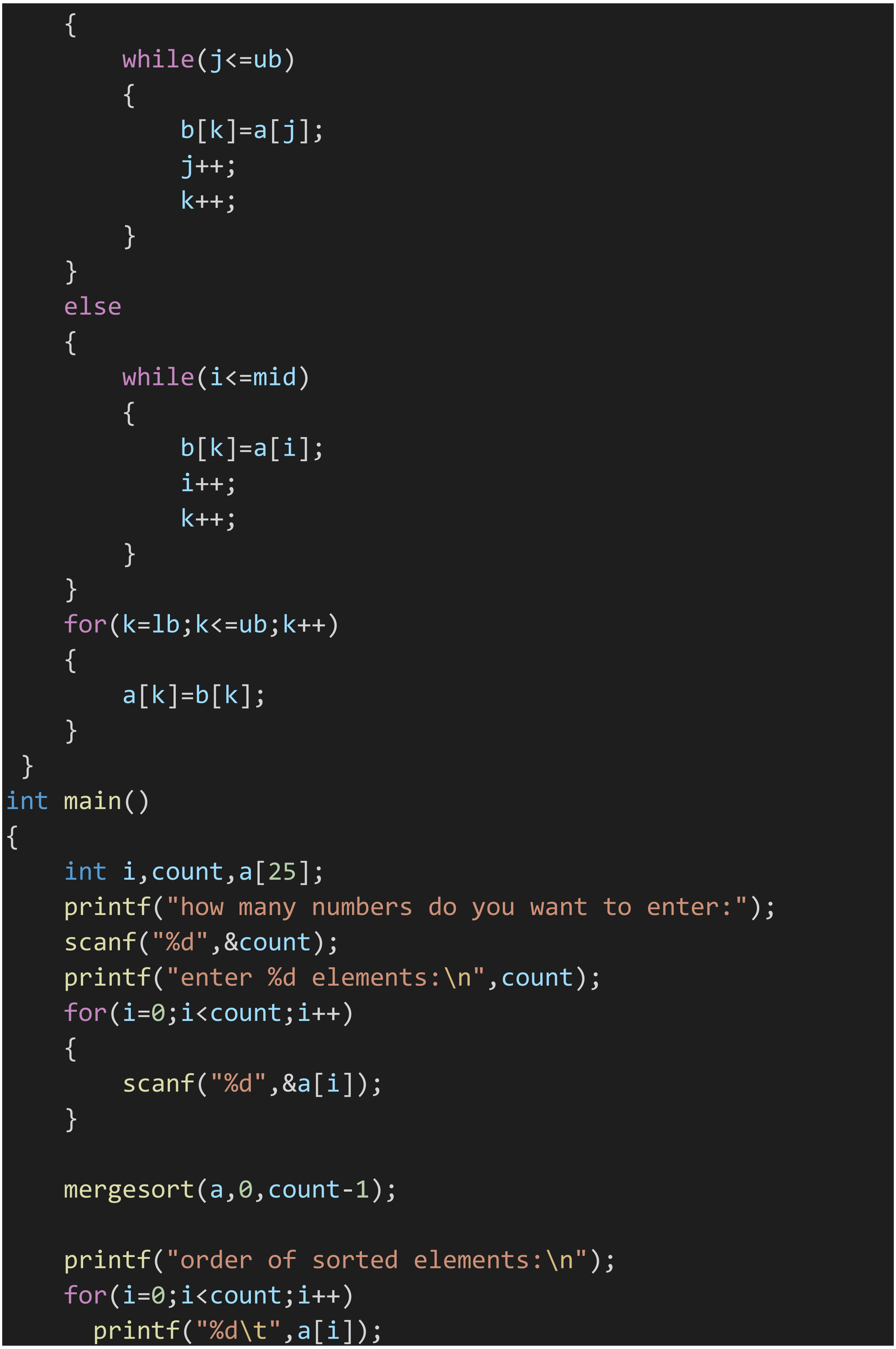
MergeSort(arr,mid,right)

End

3)Executable C program :



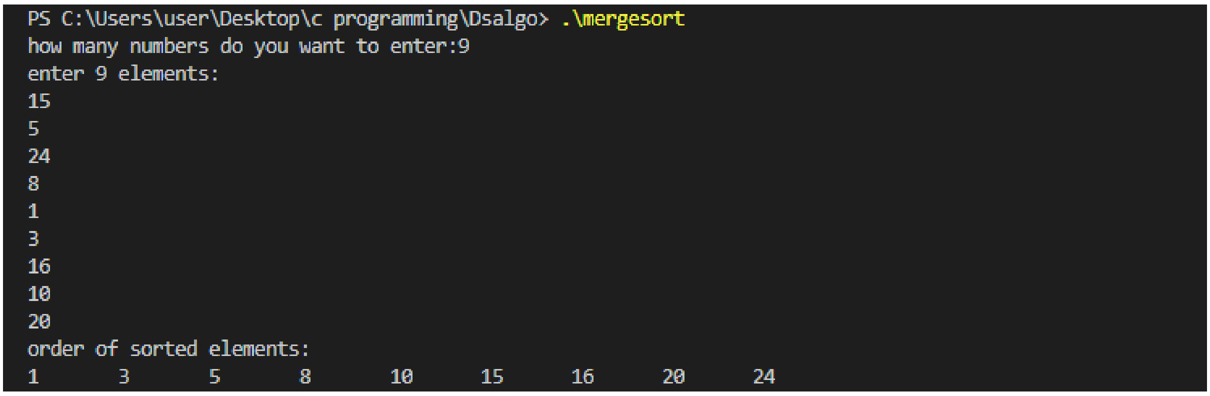
```
# include <stdio.h>
int merge(int[],int,int,int);
int mergesort(int[],int,int);
int mergesort(int a[25],int lb,int ub)
{   int mid;
    if(lb<ub)
    {
        mid=(lb+ub)/2;
        mergesort(a,lb,mid);
        mergesort(a,mid+1,ub);
        merge(a,lb,mid,ub);
    }
}
int merge(int a[25],int lb,int 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:

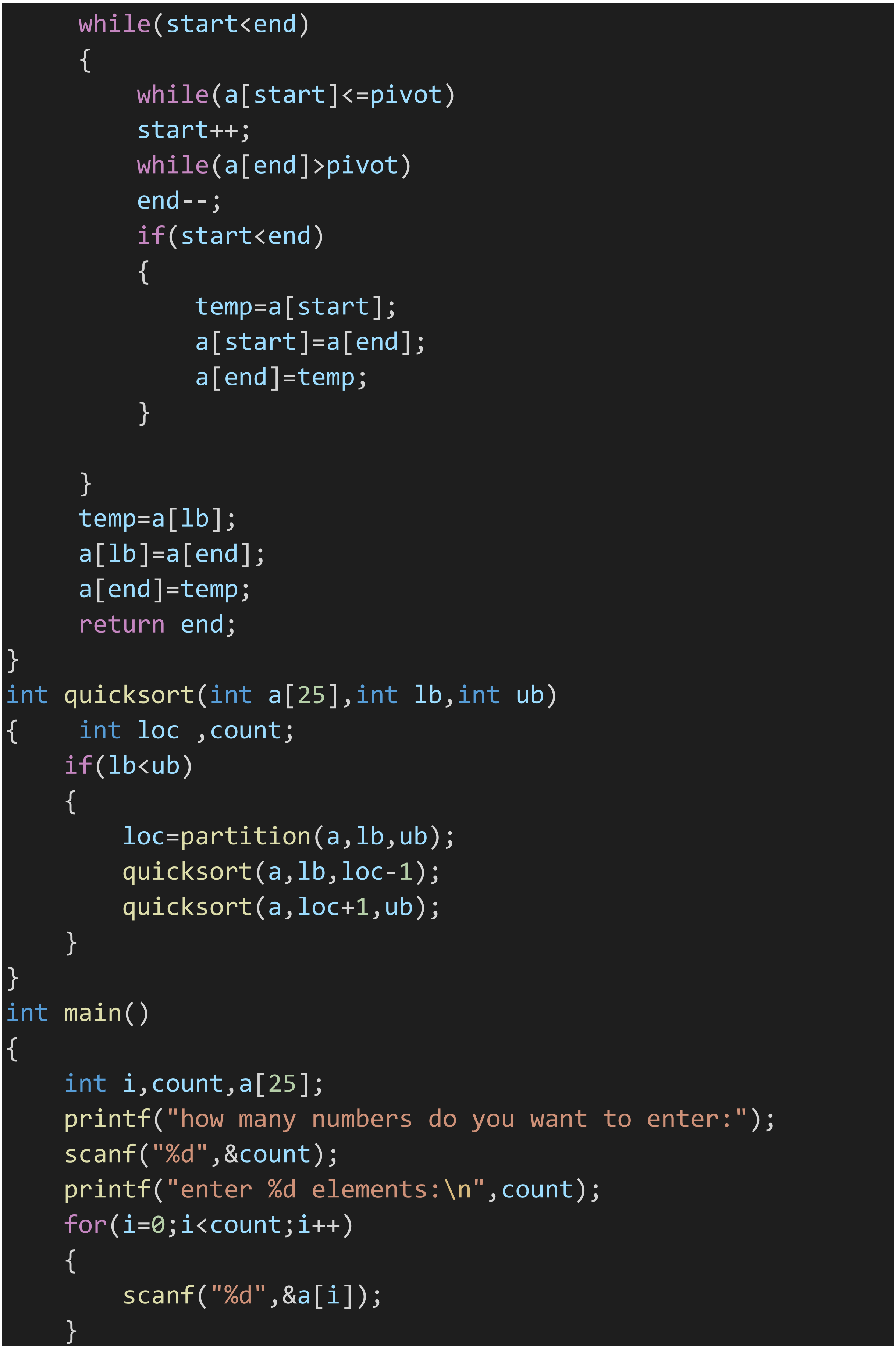
```
C program that reads N integer numbers and arrange them in
ascending order using Quick Sort
```

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
```

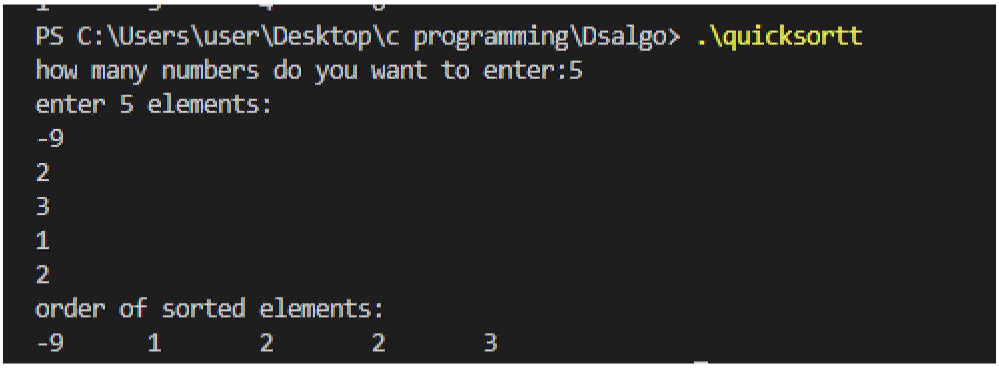
3)Executable C program :

```
# 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..**