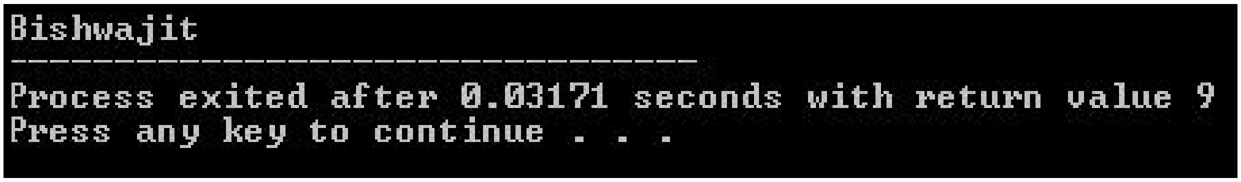
Problem 1 :WACP to print a message on the Screen.

Source Code :

```
#include<stdio.h>
int main()
{
printf("Bishwajit");
}
```

Output :



```
Bishwajit
---------------------------------
Process exited after 0.03171 seconds with return value 9
Press any key to continue . . .
```

Discussion :

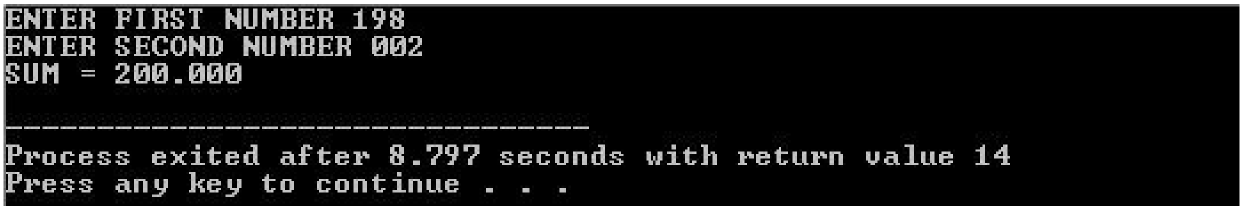
Here in this code the printf(char*) is a function in C that outputs /prints the data to the stdout i.e. the terminal in this case.

Problem 2 :WACP to add two numbers.

Source Code :

```
#include<stdio.h>
int main()
{
double a,b;              //to store the two numbers
printf("ENTER FIRST NUMBER ");
scanf("%lf",&a);       //input of the first number
printf("ENTER SECOND NUMBER ");
scanf("%lf",&b);      //input of the second number
printf("SUM = %.3lf\n",(a+b));        //printing the sum
}                              / /end of main
```

Output:



```
ENTER FIRST NUMBER 198
ENTER SECOND NUMBER 002
SUM = 200.000

--------------------------------
Process exited after 8.797 seconds with return value 14
Press any key to continue . . .
```

Discussion:

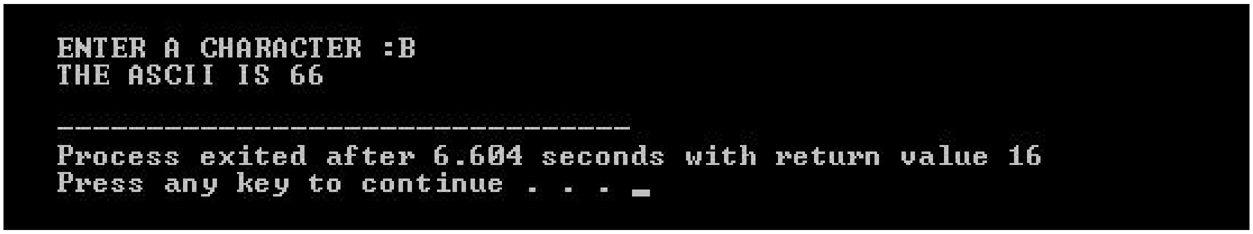
Note that I have used double as the datatype
to store the numbers which is quite logical. The format specifier for double is %lf and .3 between the % and the lf is the fieldwidth specifier which states the number of decimal places to which we want the output.

Problem 3:WACP to read a character and print its ASCII value.

Source Code :

```
#include <stdio.h>
int main()
{
char ch=0;
printf("ENTER A CHARACTER :");
scanf("%c",&ch);//input of character
printf("THE ASCII IS %d\n",ch);//display of the ASCII
}
```

Output :



```
ENTER A CHARACTER :B
THE ASCII IS 66

--------------------------------
Process exited after 6.604 seconds with return value 16
Press any key to continue . . . _
```

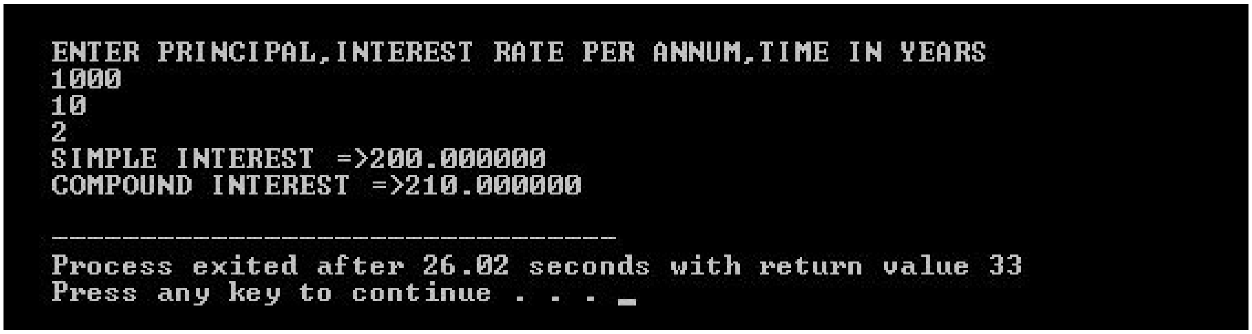
Discussion :Note
the way the ASCII is printed using the %d format specifier however while input we did use the %c specifier.

Problem 4:WACP to calculate Simple and Compound interest.

Source Code :

```
#include<stdio.h>
#include<math.h>
int main()
{
printf("ENTER PRINCIPAL,INTEREST RATE PER ANNUM,TIME IN YEARS
\n");
double principal,rate,time;
scanf("%lf %lf %lf",&principal,&rate,&time);
//SI=p*r*r/100
double si=(principal*rate*time)/100;//the simple interest
printf("SIMPLE INTEREST =>%lf ",si);//printing the simple interest
double ci=principal*(pow((1+(rate/100)),time)-1);
//the compound interest
printf("\nCOMPOUND INTEREST =>%lf \n",ci);//printing the
compound interest
}
```

Output:



```
ENTER PRINCIPAL,INTEREST RATE PER ANNUM,TIME IN YEARS
1000
10
2
SIMPLE INTEREST =>200.000000
COMPOUND INTEREST =>210.000000

--------------------------------
Process exited after 26.02 seconds with return value 33
Press any key to continue . . . _
```

Discussion :

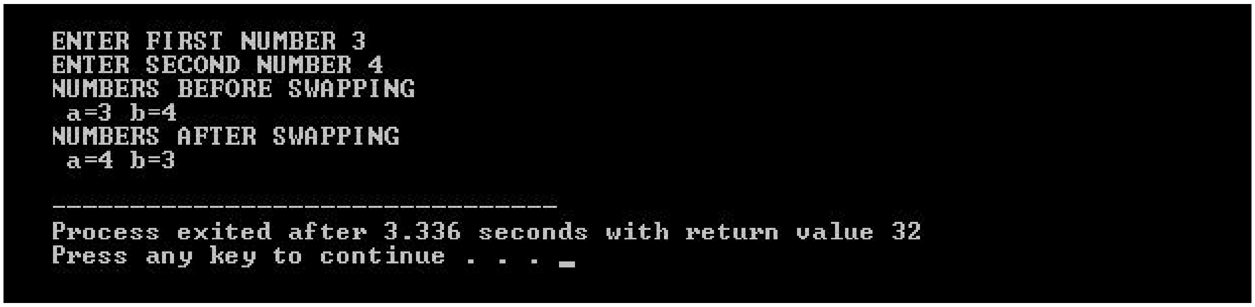
In the above program we have used the library function pow() available by including the math.h header file, which has the following prototype,double pow(double x,double y) and it computes xy and returns it.

Problem 5:WACP to swap two integer numbers without using 3rd Variable

Source Code :

```
#include<stdio.h>
int main()
{
int a,b;
printf("ENTER FIRST NUMBER ");
scanf("%d",&a);//input of first number
printf("ENTER SECOND NUMBER ");
scanf("%d",&b);//input of the second number
printf("NUMBERS BEFORE SWAPPING\n a=%d b=%d\n",a,b);
b=a+b;
a=b-a;//the swapping
b=b-a;
printf("NUMBERS AFTER SWAPPING\n a=%d b=%d\n",a,b);
}
```

Output:



```
ENTER FIRST NUMBER 3
ENTER SECOND NUMBER 4
NUMBERS BEFORE SWAPPING
 a=3 b=4
NUMBERS AFTER SWAPPING
 a=4 b=3

--------------------------------
Process exited after 3.336 seconds with return value 32
Press any key to continue . . . _
```

Discussion:

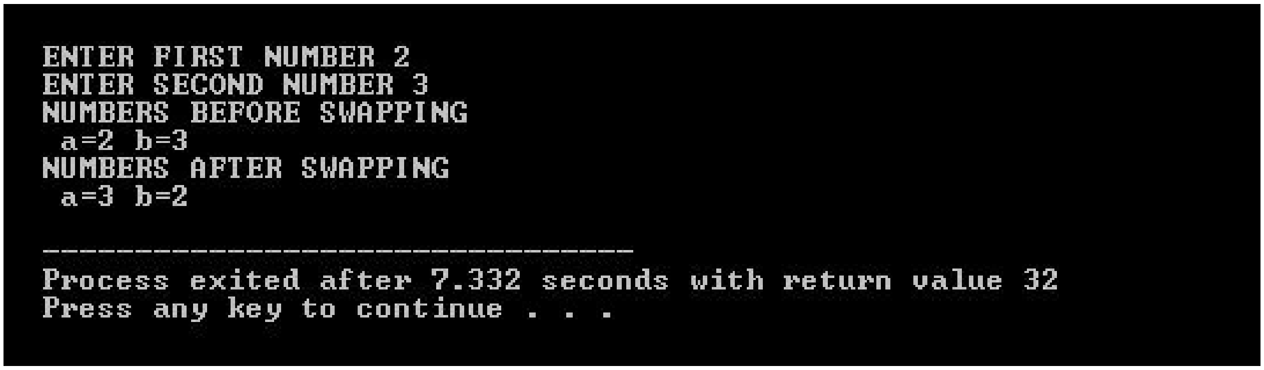
This concept of swapping is purely mathematical and has connection with the real word swapping. For example, you can't possibly swap the contents of a glass of milk and water without a third class !!
But look the way we have achieved swapping the contents of two variables in C using simple statements a bit of mathematics.

Problem 6:WACP to swap two integer numbers using 3rd Variable.

Source Code :

```
#include<stdio.h>
int main()
{
int a,b;
printf("ENTER FIRST NUMBER ");
scanf("%d",&a);//input of the first number
printf("ENTER SECOND NUMBER ");
scanf("%d",&b);//input of the second number
printf("NUMBERS BEFORE SWAPPING\n a=%d b=%d\n",a,b);
int temp;//temporary variable for swapping
temp=a;
a=b; //swapping
b=temp;
printf("NUMBERS AFTER SWAPPING\n a=%d b=%d\n",a,b);
}
```

Output:



```
ENTER FIRST NUMBER 2
ENTER SECOND NUMBER 3
NUMBERS BEFORE SWAPPING
 a=2 b=3
NUMBERS AFTER SWAPPING
 a=3 b=2

--------------------------------
Process exited after 7.332 seconds with return value 32
Press any key to continue . . .
```

Discussions:

This implements the way we have performed swapping in the real world. For example, if we have a glass of milk and that of water then we shall definitely require the third glass for swapping the contents,i.e. Milk to the third glass and pour water in the glass which cointained milk initially and then pour back milk from thw third glass to the glass which previously had water. And you are done with the swapping.

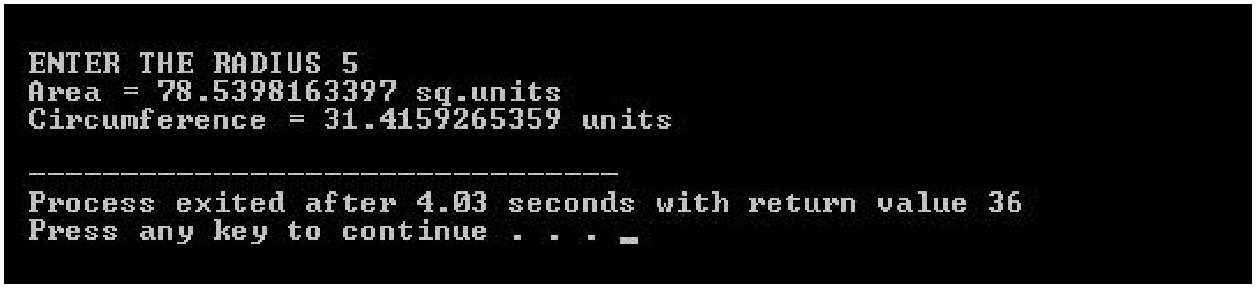
Problem 7 :WACP to find the area and circumference of a circle.

(value of radius must be given by user and use the concept of symbolic constants)

Source Code:

```
#include<stdio.h>
#include <math.h>
int main()
{
double radius;//to store the radius
printf("ENTER THE RADIUS ");
scanf("%lf",&radius);//input of the radius
printf("Area = %.10lf sq.units\n",(pow(radius,2)*M_PI));
//display of the area(area=pi*(radius)^2)
printf("Circumference = %.10lf units\n",(2*M_PI*radius));
//display of the circumference (circumference=2*pi*r)
}//end of main
```

Output :



```
ENTER THE RADIUS 5
Area = 78.5398163397 sq.units
Circumference = 31.4159265359 units

--------------------------------
Process exited after 4.03 seconds with return value 36
Press any key to continue . . .
```

Discussion:

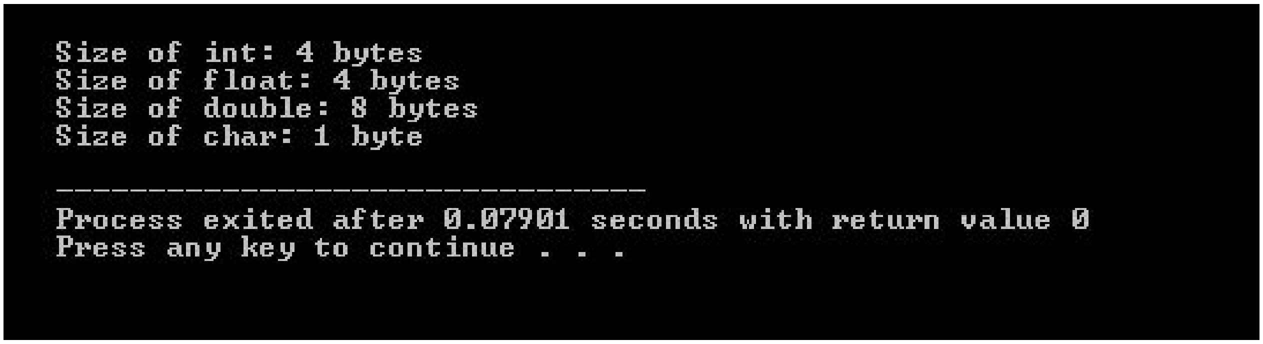
In the above program we have used the library function pow() available by including the math.h header file, which has the following prototype, double pow(double x,double y) and it computes xy and returns it. We have also used the macro M_PI available under math.h for the constant PI

Problem 8:WACP to print the sizes and range of various data types.

Source Code :

```
#include <stdio.h>
int main()
{
   int integerType;
   float floatType;
   double doubleType;
   char charType;

   // Sizeof operator is used to evaluate the size of a variable
   printf("Size of int: %ld bytes\n",sizeof(integerType));
   printf("Size of float: %ld bytes\n",sizeof(floatType));
   printf("Size of double: %ld bytes\n",sizeof(doubleType));
   printf("Size of char: %ld byte\n",sizeof(charType));

   return 0;
}
```

Output :



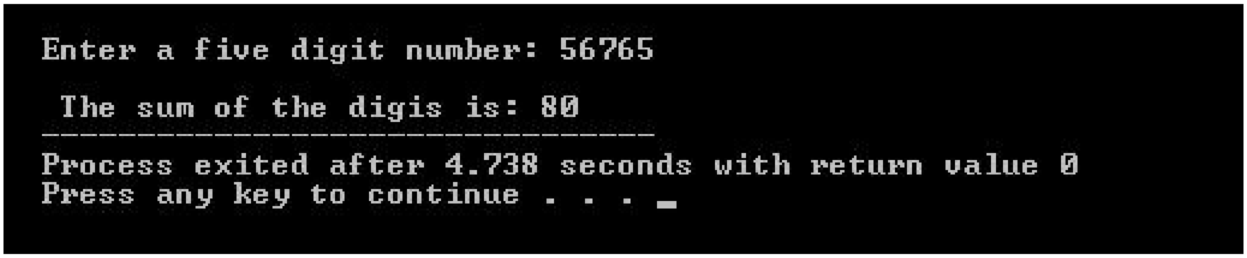
```
Size of int: 4 bytes
Size of float: 4 bytes
Size of double: 8 bytes
Size of char: 1 byte

--------------------------------
Process exited after 0.07901 seconds with return value 0
Press any key to continue . . .
```

Problem 9:WACP to calculate sum of digits of a five digit numbers. (do not use any loop construct)

Source Code :

```
#include<stdio.h>
int main()
{
   int d1,d2,d3,d4,d5,sum;
   long num;
   printf("Enter a five digit number: ");
   scanf("%ld",&num);
   d1=(num%10);
   d2=(num%100)/10;
   d3=(num%1000)/100;
   d4=(num%10000)/1000;
   d5=(num/1000);
   sum=d1+d2+d3+d4+d5;
   printf("\n The sum of the digis is: %d",sum);
   return 0;
}
```

Output:



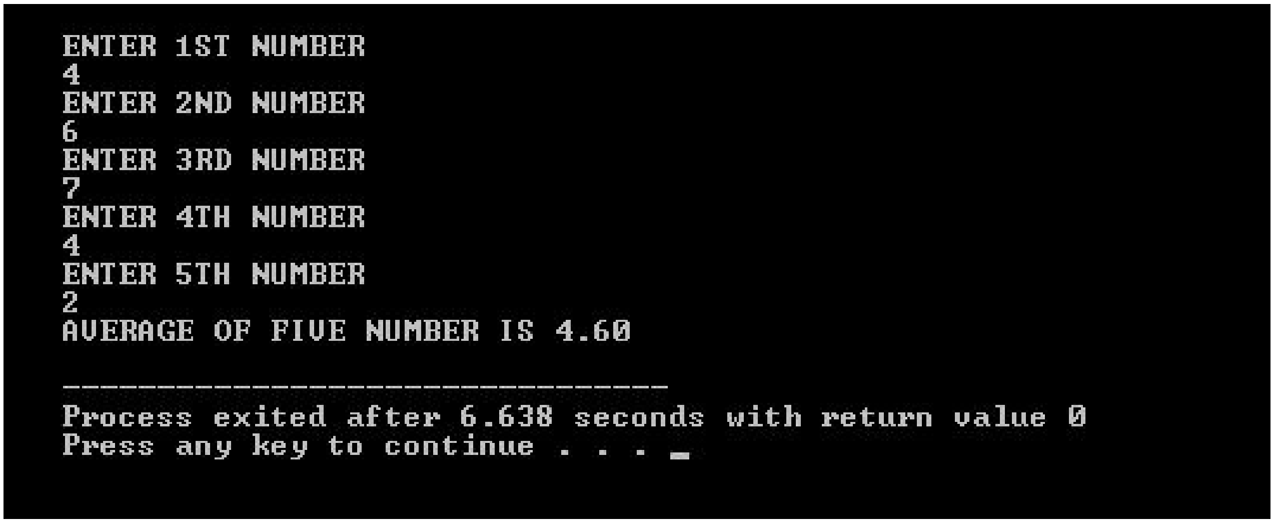
```
Enter a five digit number: 56765

 The sum of the digis is: 80
--------------------------------
Process exited after 4.738 seconds with return value 0
Press any key to continue . . .
```

Problem 10 :WACP to find the average of five numbers.

Source Code :

```
#include<stdio.h>
int main()
{
int a,b,c,d,e;
float f;
printf("ENTER 1ST NUMBER\n");
scanf("%d",&a);
printf("ENTER 2ND NUMBER\n");
scanf("%d",&b);
printf("ENTER 3RD NUMBER\n");
scanf("%d",&c);
printf("ENTER 4TH NUMBER\n");
scanf("%d",&d);
printf("ENTER 5TH NUMBER\n");
scanf("%d",&e);
f=(float)(a+b+c+d+e)/5;
printf("AVERAGE OF FIVE NUMBER IS %.2f\n",f);
return 0;
}
```

Output :



```
ENTER 1ST NUMBER
4
ENTER 2ND NUMBER
6
ENTER 3RD NUMBER
7
ENTER 4TH NUMBER
4
ENTER 5TH NUMBER
2
AVERAGE OF FIVE NUMBER IS 4.60

--------------------------------
Process exited after 6.638 seconds with return value 0
Press any key to continue . . . _
```

Discussions:Note
that we have store the loop variable i in the register for faster operations. Moreover the second run of the program shows how the input
buffer has been cleared after each input.

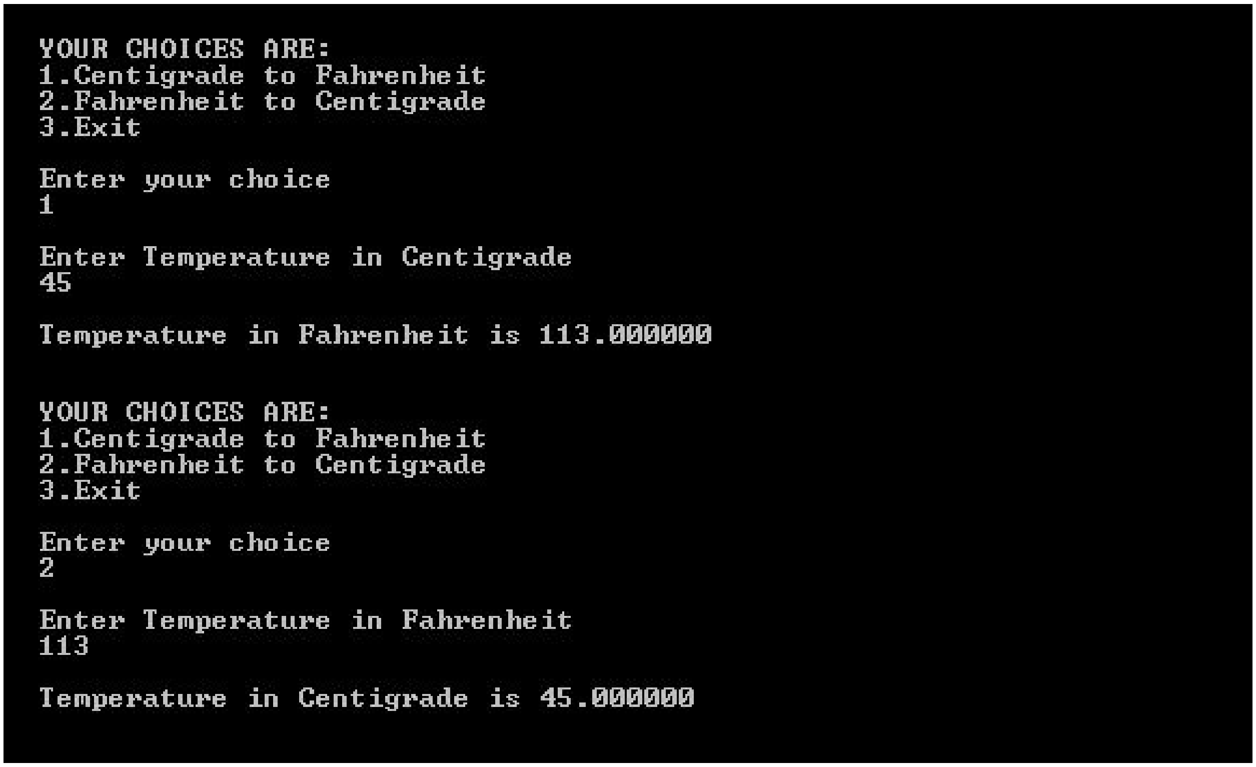
Problem 11 :WACP to convert temperature in degree F to degree C and vice versa.

Source Code:

```
#include <stdio.h>
void centigradeToFahrenheit();
//asks for a temperature in Centigrade and converts it to Fahrenheit
void fahrenheitToCentigrade();
//asks for a temperature in Fahrenheit and converts it to Centigrade
int main()
{
int ch;//to store the choice entered by user
while(1)//a infinite loop for the menu
{
//which runs till choice is entered to end the program
printf("\n\nYOUR CHOICES ARE:\n");
printf("1.Centigrade to Fahrenheit\n");
printf("2.Fahrenheit to Centigrade\n");
printf("3.Exit\n\n");
printf("Enter your choice\n");
scanf("%d",&ch);//input of choice
switch(ch)
{
case 1:centigradeToFahrenheit();
break;
case 2:fahrenheitToCentigrade();
break;
case 3: printf("\n~~~~~~~~~~~~~~END~~~~~~~~~~~~~~~~\n");
return 0;
default:printf("\nOOPS! Wrong choice. Please try again\n\n");
}//end of switch
}//end of while
}//end of main
/*THE FORMULA USED FOR CONVERSION IS
C/5=(F32)/9
*/
void centigradeToFahrenheit()
{
printf("\nEnter Temperature in Centigrade\n");
double c;//to the temperature in Centigrade
scanf("%lf",&c);//input of temperature in Centigrade
double f=(9*c/5)+32;//conversion of the temperature to that in F
printf("\nTemperature in Fahrenheit is %lf\n",f);
//printing the temperature in Fahrenheit
}
13
void fahrenheitToCentigrade()
{
```

```
printf("\nEnter Temperature in Fahrenheit\n");
double f;//to store the temperature in Fahrenheit
scanf("%lf",&f);//input of temperature in Fahrenheit
double c=5*(f32)/
9;//conversion of temperature in F to that in C
printf("\nTemperature in Centigrade is %lf\n",c);
//printing the temperature in Centigrade
}
```

Output:



```
YOUR CHOICES ARE:
1.Centigrade to Fahrenheit
2.Fahrenheit to Centigrade
3.Exit

Enter your choice
1

Enter Temperature in Centigrade
45

Temperature in Fahrenheit is 113.000000


YOUR CHOICES ARE:
1.Centigrade to Fahrenheit
2.Fahrenheit to Centigrade
3.Exit

Enter your choice
2

Enter Temperature in Fahrenheit
113

Temperature in Centigrade is 45.000000
```

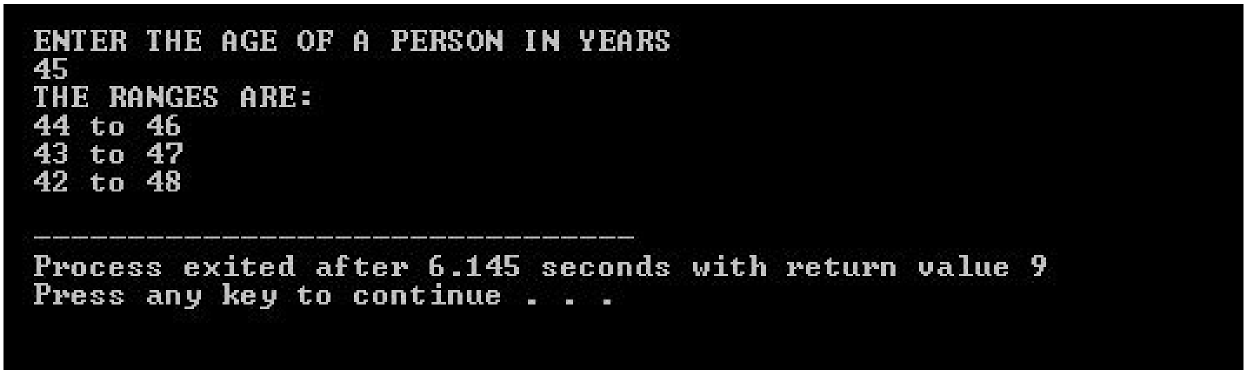
Discussions:Here
the above code implements a menu driven program, one can also do it without a menu by doing two seperate codes or by finding the Centigrade value assuming the input is in Fahrenheit and also considering the same input to be in Fahrenheit and convert it to Centigrade.

Problem 12:WACP to read in the age of a person as an integer number in years and print the three immediate integral range of the age entered. For example if the user enters 25, then the output will print the three immediate integral ranges: 24 to 26, 23 to 27 and 22 to 28.

Source Code :

```
#include <stdio.h>
void range(int age,int diff);//accepts the range and the
//difference of the upper or the lower limit of the range
//from the age
int main()
{
//displays a prompt on the screen
printf("ENTER THE AGE OF A PERSON IN YEARS\n");
int age;// to store the age
scanf("%d",&age);//input of age
printf("THE RANGES ARE:\n");
range(age,1);//prints the first range
range(age,2);//prints the second range
range(age,3);//prints the third range
}
void range(int age,int diff)
{
printf("%d to %d\n",agediff,
age+diff);
//displays the range on the screen
//as per the difference entered in arguments
}//end of range(int,int)
```

Output:



```
ENTER THE AGE OF A PERSON IN YEARS
45
THE RANGES ARE:
44 to 46
43 to 47
42 to 48

--------------------------------
Process exited after 6.145 seconds with return value 9
Press any key to continue . . .
```

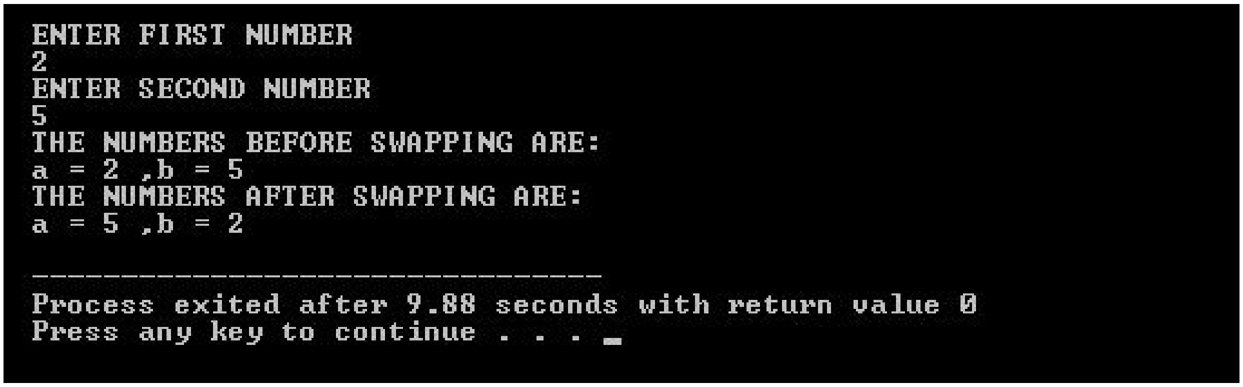
Discussions:Note
that in the above code I have used function to reduce the code redundancy of writing three print statements for the three ranges.
Note
that one can also do it without any functions

Problem 13 :WACP to swap two integer numbers using bitwise operators.

Source Code :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
int a,b;
printf("ENTER FIRST NUMBER\n");
scanf("%d",&a);
printf("ENTER SECOND NUMBER\n");
scanf("%d",&b);
printf("THE NUMBERS BEFORE SWAPPING ARE:\n");
printf("a = %d ,b = %d\n",a,b); //printing the numbers before swapping
a=a^b;
b=a^b;//stores a in b
a=a^b;//stores b in a
printf("THE NUMBERS AFTER SWAPPING ARE:\n");
printf("a = %d ,b = %d\n",a,b);
//printing the numbers after swapping
return 0;
}
```

Output :



```
ENTER FIRST NUMBER
2
ENTER SECOND NUMBER
5
THE NUMBERS BEFORE SWAPPING ARE:
a = 2 ,b = 5
THE NUMBERS AFTER SWAPPING ARE:
a = 5 ,b = 2

--------------------------------
Process exited after 9.88 seconds with return value 0
Press any key to continue . . .
```

Discussions:

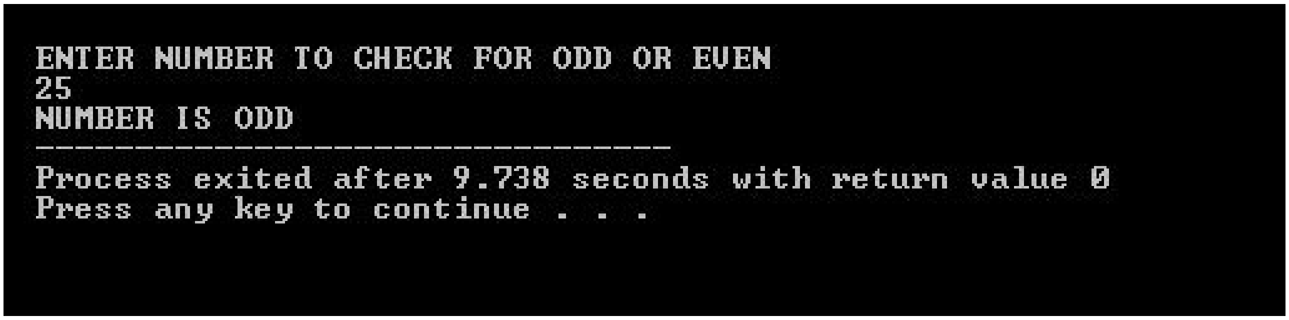
Here we have used the concept that the bitwise XOR operator is associative and commutative we know that a XOR 0 gives a and a^a gives 0 so,
a^b^b=a^(b^b)=a^0=a and
a^b^a=b^(a^a)=b^0=b

Problem 14 :WACP to check whether a number is odd or even. (using simple if statement)

Source Code :

```
#include<stdio.h>
int main()
{
        int num;
        printf("ENTER NUMBER TO CHECK FOR ODD OR EVEN\n");
        scanf("%d",&num);
        if(num%2==0)
        printf("NUMBER IS EVEN");
        else
        printf("NUMBER IS ODD");
        return 0;

        }
```

Output :



```
ENTER NUMBER TO CHECK FOR ODD OR EVEN
25
NUMBER IS ODD
--------------------------------
Process exited after 9.738 seconds with return value 0
Press any key to continue . . .
```

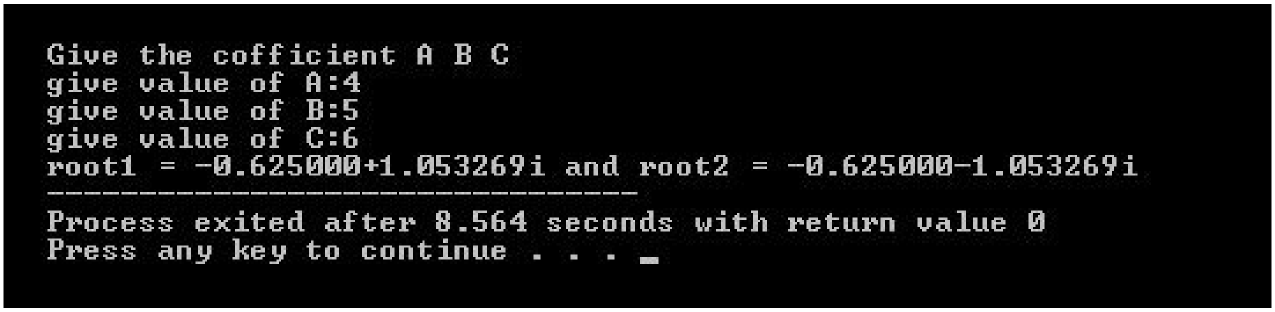
Problem 15 :WACP to find the roots of a Quadratic equation. (using if else statement)

Source Code :

```
#include<stdio.h>
#include<math.h>
int main()
{
        double a,b,c,dis,root1,root2,real,imaginary;
        printf("Give the cofficient A B C");
        printf("\ngive value of A:");
        scanf("%lf",&a);
        printf("give value of B:");
```

```
         scanf("%lf",&b);
         printf("give value of C:");
         scanf("%lf",&c);
         dis =b*b-4*a*c;
         if(dis>0)
         {
        root1 = (-b+sqrt(dis))/(2*a);
        root2 = (-b-sqrt(dis))/(2*a);
        printf("root1 = %lf and root2 = %lf",root1 , root2);
    }
    else if (dis == 0)
    {
        root1 = root2 = -b/(2*a);
        printf("root1 = root2 = %lf", root1);
    }
    else
    {
        real = -b/(2*a);
        imaginary = sqrt(-dis)/(2*a);
        printf("root1 = %lf+%lfi and root2 = %f-%fi", real, imaginary, real,
imaginary);
    }

    return 0;
}
```

Output :



```
Give the cofficient A B C
give value of A:4
give value of B:5
give value of C:6
root1 = -0.625000+1.053269i and root2 = -0.625000-1.053269i
--------------------------------
Process exited after 8.564 seconds with return value 0
Press any key to continue . . . _
```

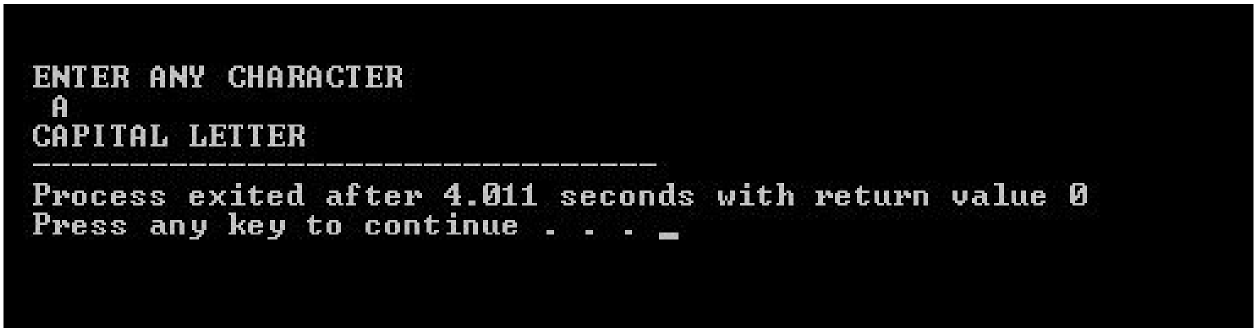
Discussion :The
code written above deals with all possible cases in a quadratic equation, including the situations when the roots are imaginary and the corresponding imaginary roots are also displayed

Problem 16 :WACP to determine whether the character entered is a capital, small case letter, a digit or a special symbol. (using else if ladder)

Source Code :

```
#include <stdio.h>
int main()
{
   char ch;
   printf("ENTER ANY CHARACTER\n ");
   scanf("%c", &ch);
   if(ch >= 'a' && ch <= 'z')
   {
      printf("SMALL LETTER", ch);
   }
   else if(ch >= 'A' && ch <= 'Z')
        {
      printf("CAPITAL LETTER", ch);
   }
   else if(ch >= '0' && ch <= '9')
   {
      printf("NUMBER", ch);
   }
   else
   {
      printf("SPECIAL CHARACTER", ch);
   }
   return 0;
}
```

Output :



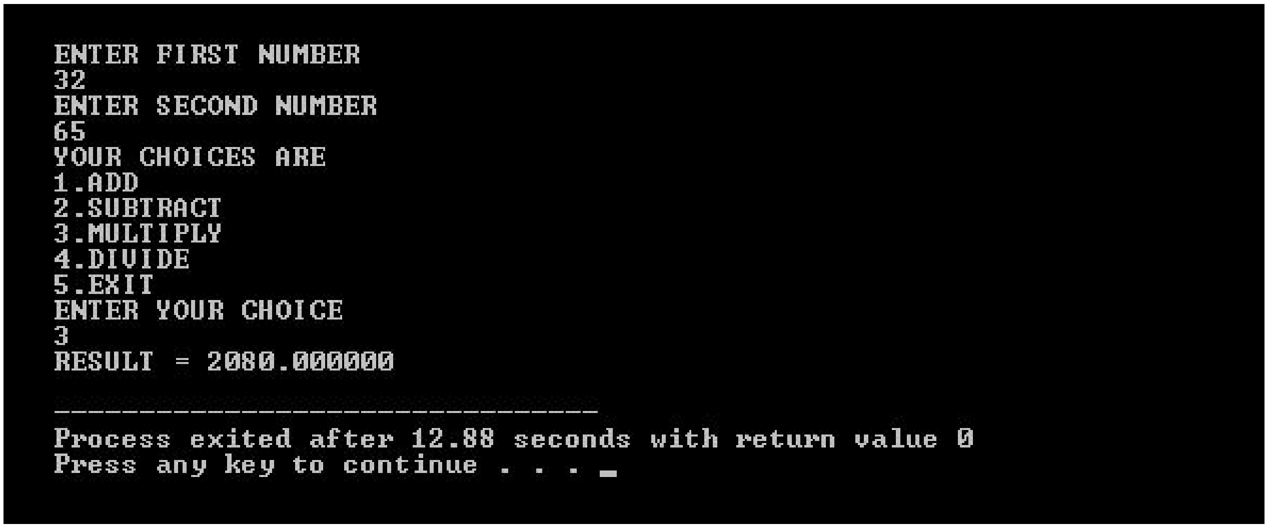
```
ENTER ANY CHARACTER
 A
CAPITAL LETTER
--------------------------------
Process exited after 4.011 seconds with return value 0
Press any key to continue . . . _
```

Problem 17 :WACP to add, subtract, multiply and divide two numbers using switch case.

Source Code :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
double a,b,result=0;
printf("ENTER FIRST NUMBER\n");
scanf("%lf",&a);//input of a
printf("ENTER SECOND NUMBER\n");
scanf("%lf",&b);//input of b
printf("YOUR CHOICES ARE \n");
printf("1.ADD\n2.SUBTRACT\n3.MULTIPLY\n4.DIVIDE\n5.EXIT\n");
printf("ENTER YOUR CHOICE\n");
short int ch;//to store the choice
scanf("%hd",&ch);//input of ch
switch(ch)
{
case 1: result=a+b;
break;
case 2: result=ab;
22
break;
case 3: result=a*b;
break;
case 4: if(!b)//checking the possibility of b being zero
{
printf("DIVISION BY ZERO NOT POSSIBLE\n");
return 0;
}
result=a/b;
break;
case 5: return 0;
default:printf("WRONG INPUT OF CHOICE\n");
return 0;
}//end of switch
printf("RESULT = %lf\n",result);
return 0;
}
```

Output:



```
ENTER FIRST NUMBER
32
ENTER SECOND NUMBER
65
YOUR CHOICES ARE
1.ADD
2.SUBTRACT
3.MULTIPLY
4.DIVIDE
5.EXIT
ENTER YOUR CHOICE
3
RESULT = 2080.000000

--------------------------------
Process exited after 12.88 seconds with return value 0
Press any key to continue . . . _
```

Discussions:Note

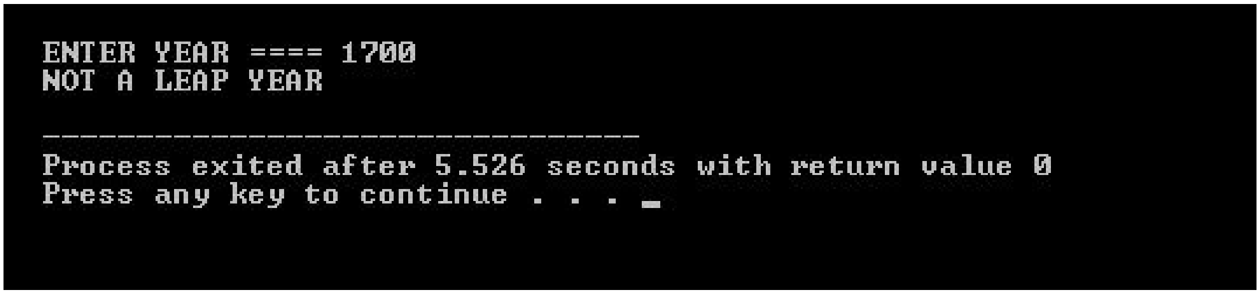
that we have checked the case of division of zero error. One could also do the same code using simple if else if statements.

Problem 18:Any year is entered through keyboard. WACP to determine whether the year is a leap year or not.
(using conditional operators).

Source Code :

```
#include <stdio.h>
int main()
{
int yr;
printf("ENTER A YEAR\n");
scanf("%d",&yr);
(!(yr%400)||(yr%100!=0 &&yr%4==0))?printf("LEAP
YEAR\n"):printf("NOT A LEAP YEAR\n");
return 0;
}
```

Output:



```
ENTER YEAR ==== 1700
NOT A LEAP YEAR

--------------------------------
Process exited after 5.526 seconds with return value 0
Press any key to continue . . . _
```

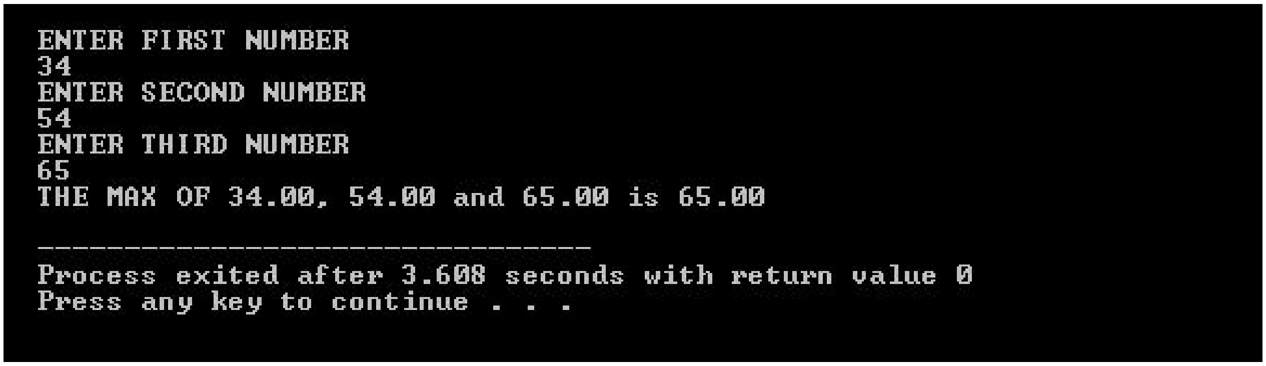
Program 19:Find greatest of three numbers.(using ternary operator).

Source Code :

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

```
int main()
{
double a,b,c;//to store the numbers
printf("ENTER FIRST NUMBER\n");
scanf("%lf",&a);//input of first number
printf("ENTER SECOND NUMBER\n");
scanf("%lf",&b);//input of second number
printf("ENTER THIRD NUMBER\n");
scanf("%lf",&c);//input of third number
double max = a>b?(a>c?a:c):(b>c?b:c);//getting the max
printf("THE MAX OF %.2lf, %.2lf and %.2lf is %.2lf\n",a,b,c,max);
//display of the max
return 0;
}
```

Output:



```
ENTER FIRST NUMBER
34
ENTER SECOND NUMBER
54
ENTER THIRD NUMBER
65
THE MAX OF 34.00, 54.00 and 65.00 is 65.00

--------------------------------
Process exited after 3.608 seconds with return value 0
Press any key to continue . . .
```

Discussion:The use of the parenthesis plays a very important role in finding the max of the three numbers here. Also note that the ternery operator becomes handy only when the conditions are simple , in multilevel nesting it might get complex,and moreover
unlike if elseif else
ladder a block containing many statements cannot be executed if a condition is true!!