1. Write a program containing a loop that iterates from 1 to 1000 using a variable I, which is incremented each time around the loop. The program should output the value of I every hundred iterations (i.e., the output should be 100, 200, etc.).

**create table temp**

**(**

**num int**

**);**

**---------------------------------**

**delimiter //**

**create procedure hundreads()**

**begin**

**declare x int default 1;**

**while x <= 1000 do**

**if x%100 = 0 then**

**insert into temp values(x);**

**set x = x+1;**

**end if;**

**set x = x+1;**

**end while;**

**end; //**

**delimiter ;**

**---------------------------------**

**call hundreads();**

**select \* from temp;**

1. Write a program that examines all the numbers from 1 to 999, displaying all those for which the sum of the cubes of the digits equal the number itself.

**delimiter //**

**create procedure check\_numbers()**

**begin**

**declare num int default 1;**

**declare sum\_digitcube int;**

**while num <1000 do**

**set sum\_digitcube =0;**

**set @x = num;**

**while @x > 0 do**

**set @y = @x % 10;**

**set sum\_digitcube = sum\_digitcube + power(@y,3);**

**set @x = floor(@x/10);**

**end while;**

**if sum\_digitcube = num then**

**insert into tempp values(num);**

**end if;**

**set num = num+1;**

**end while;**

**end; //**

**delimiter ;**

**drop table tempp;**

**create table tempp**

**(**

**num int);**

**call check\_numbers();**

**select \* from tempp;**

1. Write a program that Selects from any table a minimum and maximum value for a radius, along with an increment factor, and generates a series of radii by repeatedly adding the increment to the minimum until the maximum is reached. For each value of the radius, compute and display the circumference, area, and volume of the sphere. (Be sure to include both the maximum and the minimum values.).

**create table temp**

**(**

**radius float,**

**area float,**

**circumference float,**

**volume float**

**);**

**delimiter //**

**create procedure sphere\_cal(rmax float,rmin float, increment float)**

**begin**

**declare area float;**

**declare circumference float;**

**declare volume float;**

**while rmin <= rmax do**

**set area = 3.142\*(rmin)^2;**

**set circumference = 2\*3.142\*rmin;**

**set volume = 4/3 \* 3.142 \* (rmin)^3;**

**insert into temp values (rmin,area,circumference,volume);**

**set rmin = rmin \* increment;**

**end while;**

**end; //**

**delimiter ;**

**call sphere\_cal(100,10,1.5);**

**select \* from temp;**

1. A palindrome is a word that is spelled the same forward and backward, such as level, radar, etc. Write a program to Selects from any table a five letter word and determine whether it is a palindrome.

**create table temp**

**(**

**word char(5),**

**Pailndrome char(5)**

**);**

**delimiter //**

**create procedure is\_Palindrome(word char(5))**

**begin**

**declare reversed\_word char(5);**

**set reversed\_word = REVERSE(word);**

**if reversed\_word = word then**

**insert into temp values(word,'true');**

**else**

**insert into temp values(word,'false');**

**end if;**

**end; //**

**delimiter ;**

**call is\_Palindrome('table');**

**call is\_Palindrome('level');**

**select \* from temp;**

1. Modify the above program to Select from any table a variable length word. This requires determining how many characters are read in.

**delimiter //**

**create procedure is\_Palindrome(word varchar(255))**

**begin**

**declare reversed\_word varchar(255);**

**set reversed\_word = REVERSE(word);**

**if reversed\_word = word then**

**insert into temp values(word,'true');**

**else**

**insert into temp values(word,'false');**

**end if;**

**end; //**

**delimiter ;**

**create table temp**

**(**

**word varchar(255),**

**Pailndrome char(5)**

**);**

**call is\_Palindrome('sasalasas');**

**select \* from temp;**