VENN DIAGRAM

UNION:

*CODE*

#include<stdio.h>

#include<math.h>

void main(){

    int width =60; //dimension of rectangle

    int height =30;

    int cx1= width/2;

    int cy1=height/2;

    int radius=8;

    int cx2=width/3;

    int cy2=height/2;

    //drawing a rectangle

    for(int i=1;i<=height;++i){          //for row

        for(int j=1;j<=width;++j){       //for coloumn

            if( j==1 ||j==width){    //boundary of rectangle's side

                printf("|");

            }

            else if(i==1 || i== height ){ //boundary of rectangle’s top &bottom

                printf("-");

            }

            else if(fabs(pow(j-cx1,2)+pow(i-cy1,2)-pow(radius,2))<=6){ //circumference of circle1

                printf("+");

            }

            else if(fabs(pow(j-cx2,2)+pow(i-cy2,2)-pow(radius,2))<=6){  //circumference of circle2

                printf("\*");

            }

            else if((pow(j-cx1,2)+pow(i-cy1,2)<pow(radius,2)) ||(pow(j-cx2,2)+pow(i-cy2,2)<pow(radius,2))){  //union

                printf(".");

            }

            else{

                printf(" ");

            }

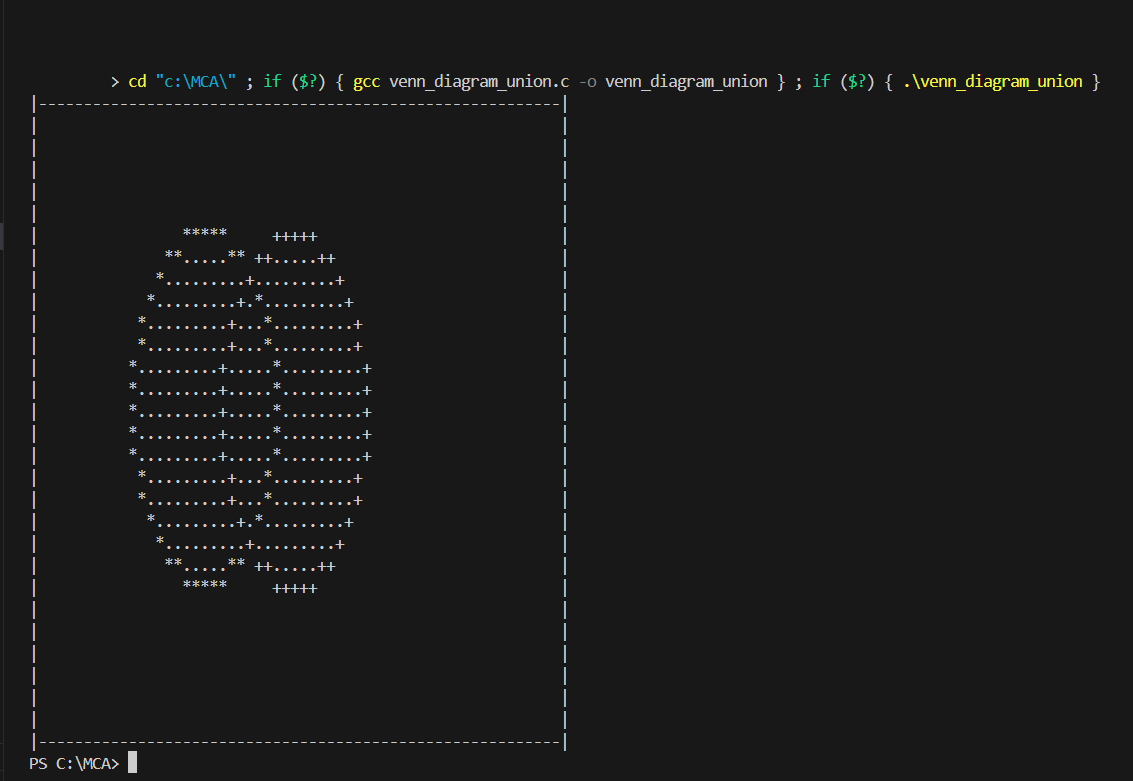
        }

        printf("\n");    //after completion of a row

    }

}

OUTPUT



INTERSECTION

CODE

#include<stdio.h>

#include<math.h>

void main(){

    int width =60; //dimension of rectangle

    int height =30;

    int cx1= width/2;

    int cy1=height/2;

    int radius=8;

    int cx2=width/3;

    int cy2=height/2;

    //drawing a rectangle

    for(int i=1;i<=height;++i){          //for row

        for(int j=1;j<=width;++j){       //for coloumn

            if( j==1 ||j==width){    //boundary of rectangle's side

                printf("|");

            }

            else if(i==1 || i== height ){     //boundary of rectangle top & bottom

                printf("-");

            }

            else if(fabs(pow(j-cx1,2)+pow(i-cy1,2)-pow(radius,2))<=6){ //circumference of circle1

                printf("+");

            }

            else if(fabs(pow(j-cx2,2)+pow(i-cy2,2)-pow(radius,2))<=6){  //circumference of circle2

                printf("\*");

            }

            else if((pow(j-cx1,2)+pow(i-cy1,2)<pow(radius,2)) && (pow(j-cx2,2)+pow(i-cy2,2)<pow(radius,2))){  //Intersection

                printf(".");

            }

            else{

                printf(" ");

            }

        }

        printf("\n");    //after completion of a row

    }

}

OUTPUT

