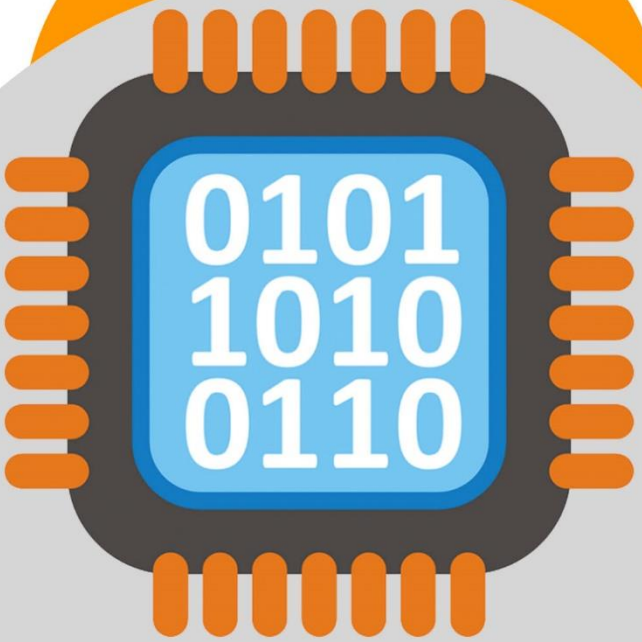


SHAPE DETECTOR

Program that takes points from the user and determine what's the shape of this points square, triangle or rectangle.

Team Name

Arch2 Team

Shape Detector
Arch2 Team

Procedures

CalculateDistance(): procedure that calculate the distance between two points.
- inputs: two points.
- outputs: length between two points.
- pre: having two points.

IsSquare(): procedure that determine if the four points make a square by check if the diagonals have the same distance and all ribs are equal.
- inputs: distance between each two point from four points.
- outputs: srting that comes out if the the shape is square or not
- pre: having four points and calculating the distances between each point with the first point.

IsRectangle(): procedure that determine if the four points make a rectangle by check if the diagonals are equal, the ribs of the length have the same distance and the ribs of the width have the same distance.
- inputs: distance between each two point from four points.
- outputs: srting that comes out if the shape is rectangle or not.
- pre: having four points and calculating the distances between each point with the first point.

IsTriangle(): procedure that determine if the three points make a triangle.
- inputs: distance between each two point from three points.
- outputs: srting that comes out if the shape is triangle or not.
- pre: having three points and calculating the distances between each point with the first point.

drawSquare(): procedure that draw a square with stars after the program comes out that the shape is square.
- inputs: if the shape is square.
- outputs: square shaped with stars.
- pre: IsSquare procedure comes out that the shape is square.

Shape Detector
Arch2 Team

Procedures

drawRectangle(): procedure that draw a rectangle with stars after the program comes out that the shape is rectangle.
- inputs: if the shape is rectangle.
- outputs: rectangle shaped with stars.
- pre: IsRectangle procedure comes out that the shape is rectangle.

drawTriangle(): procedure that draw a triangle with stars after the program comes out that the shape is triangle.
- inputs: if the shape is triangle.
- outputs: triangle shaped with stars.
- pre: IsTriangle procedure comes out that the shape is triangle.

menu (): procedure that ask the user how many numbers of points that he wants to enter.
- inputs: option that determine if there are three or four points from the user.
- outputs: none, it just calls get points procedure that depends on the option of the user.
- pre: set option to zero.

getThreePoints(): procedure that ask the user to enter the three points.
- inputs: three points from the user.
- outputs: three points.
- pre: the option variable in menu is equal 3.

getFourPoints(): procedure that ask the user to enter the three points.
- inputs: four points from the user.
- outputs: four points.
- pre: the option variable in menu() procedure is equal 4.

Shape Detector
Arch2 Team

Procedures

printThreePoints(): procedure that print the value of the three points that the user has been entered on the screen.
- inputs: three points from the getThreePoints() procedure.
- outputs: print the three points values on the screen.
- pre: the user has been entered three points and the option variable in menu() procedure is equal 3.

printFourPoints(): procedure that print the value of the four points that the user has been entered on the screen.
- inputs: four points from the getFourPoints() procedure.
- outputs: print the four points values on the screen.
- pre: the user has been entered four points and the option variable in menu() procedure is equal 4.

doThreePoints(): procedure that print the value of the three points and print if the four points are making triangle shape or unknown and draw this shape.
all of it done by calling: 1- printThreePoints().
2- isTriangle().
3- drawTriangle().
- inputs: none.
- outputs: print the values of the three points and the shape of it.
- pre: the user has entered three points and the option variable in menu() procedure is equal 3.

doFourPoints(): procedure that print the value of the four points and print if the four points are making square, rectangle or unknown and draw the square or rectangle.
all of it done by calling: 1- printFourPoints().
2- isRectangle() or isSquare().
3- drawRectangle() or drawSquare().
-inputs: none.
-outputs: print the values of the four points and the shape of it.
-pre: the user has entered four points and the option variable in menu() procedure is equal 4.

Shape Detector

Arch2 Team

Pseudo Code

```
function calculateDistance(point1[], point2[])
    distance := ((point1[0] - point2[0]) x (point1[0] - point2[0])) + ((point1[1] -
    point2[1]) x (point1[1] - point2[1]))
EndFunction

function isSquare(point1[], point2[], point3[], point4[])
begin
    distance2 := calculateDistance(point1, point2)
    distance3 := calculateDistance(point1, point3)
    distance4 := calculateDistance(point1, point4)

    if (distance2 = 0 or distance3 = 0 or distance4 = 0)
        return false
    if ((distance2 = distance3) and (2 x distance2 = distance4)
      and (2 x calculateDistance(point2, point4) = calculateDistance(point2,
      point3)))
        return true
    if ((distance3 = distance4) and (2 x distance3 = distance2)
      and (2 x calculateDistance(point3, point2) = calculateDistance(point3,
      point4)))
        return true
    if ((distance2 = distance4) and (2 x distance2 = distance3)
      and (2 x calculateDistance(point2, point3) = calculateDistance(point2,
      point4)))
        return true
    return false
EndFunction
```

Shape Detector
Arch2 Team

Pseudo Code

```
function isRectangle(point1[], point2[], point3[], point4[])
begin
    distanceWidth1 := calculateDistance(point1, point2)
    distanceWidth2 := calculateDistance(point3, point4)

    distanceHeight1 := calculateDistance(point1, point4)
    distanceHeight2 := calculateDistance(point2, point3)

    distanceDiagonal1 := calculateDistance(point1, point3)
    distanceDiagonal2 := calculateDistance(point2, point4)

    if (distanceHeight1 = distanceHeight2
        and distanceWidth1 = distanceWidth2
        and distanceDiagonal1 = distanceDiagonal2
        and distanceHeight1 != distanceWidth1)
          return true
    else
          return false
EndFunction

function isTriangle(point1[], point2[], point3[])
    area := point1[0] x (point2[1] - point3[1]) +
        point2[0] x (point3[1]) - (point1[1]) +
        point3[0] x (point1[1]) - (point2[1])
   if (area = 0)
      return false
   else
     return true
EndFunction
```

Shape Detector
Arch2 Team

Pseudo Code

```
function drawSquare(point1[], point2[])
double size := round(sqrt(calculateDistance(point1, point2)))

    for i := 0 to size
    begin
      for j := 0 to size
        if (i = 0 or i = size - 1 or j = 0 or j = size - 1)
          print "* "
        else
          print "  "
        endIf
      endFor
    endFor
EndFunction

function drawRectangle(point1[], point2[], point3[], point4[])

    distanceWidth := round(sqrt(calculateDistance(point1, point2)))
    distanceHeight := round(sqrt(calculateDistance(point1, point4)))

    for i := 0 to distanceWidth
      begin

     for j := 0 to distanceHeight
        if (i = 0 or i = distanceWidth - 1 or j = 0 or j = distanceHeight - 1)
          print "* "
        else
          print "  "
        endIf
      endFor
      printLine
    endFor
EndFunction
```

Shape Detector

Arch2 Team

Pseudo Code

```
function drawTriangle(point1[], point2[], point3[])
  distanceWidth := round(sqrt(calculateDistance(point1, point2)))
  distancelHeight := round(sqrt(calculateDistance(point2, point3)))
  distancerHeight := round(sqrt(calculateDistance(point1, point3)))

  for i := 1 to distancelHeight
  begin
   for space := i to distancelHeight
    print "  "

   for j := 1 to(2 x distancelHeight - 1)
      if (i = distancelHeight or j = 1 or j = 2 x i - 1)
        print "*"
      else
        print " "
      endIf
   endFor
   printLine
  endFor
EndFunction

function printFourPoints(point1[], point2[], point3[], point4[])

  for i : = 0 to 1
    print "(" point1[i] ", "  point1[i + 1] ")"
    print "(" point2[i] ", "  point2[i + 1] ")"
    print "(" point3[i] ", "  point3[i + 1] ")"
    print "(" point4[i] ", "  point4[i + 1] ")"
  endFor
  printLine
EndFunction
```

Shape Detector
Arch2 Team

Pseudo Code

```
function printThreePoints(point1[], point2[], point3[])

   for i : = 0 to 1
     print "(" point1[i] ", " point1[i + 1] ")"
     print "(" point2[i] ", " point2[i + 1] ")"
     print "(" point3[i] ", " point3[i + 1] ")"
  endFor
     printLine
EndFunction

function getFourPoints(point1[], point2[], point3[], point4[])

   print "Enter the points: "
   input point1[0], point1[1]
   input point2[0], point2[1]
   input point3[0], point3[1]
   input point4[0], point4[1]
EndFunction

function getThreePoints(point1[], point2[], point3[])

   print "Enter the points: "
   input point1[0], point1[1]
   input point2[0], point2[1]
   input point3[0], point3[1]
EndFunction
```

Shape Detector

Arch2 Team

Pseudo Code

```
function doFourPoints(point1[], point2[], point3[], point4[])

  printLine
  call : printFourPoints(point1, point2, point3, point4)

  if (isSquare(point1, point2, point3, point4))
    print "Square! "
    call : drawSquare(point1, point2)

  else if (isRectangle(point1, point2, point3, point4))
      print "Rectangle! "
      call: drawRectangle(point1, point2, point3, point4)
  else
      print "UnKnown! "
EndFunction

function doThreePoints(point1[], point2[], point3[])
  printLine
  call : printThreePoints(point1, point2, point3)

  if (isTriangle(point1, point2, point3))
    print "Triangle! "
  call : drawTriangle(point1, point2, point3)
  else
    print "UnKnown! "
EndFunction
```

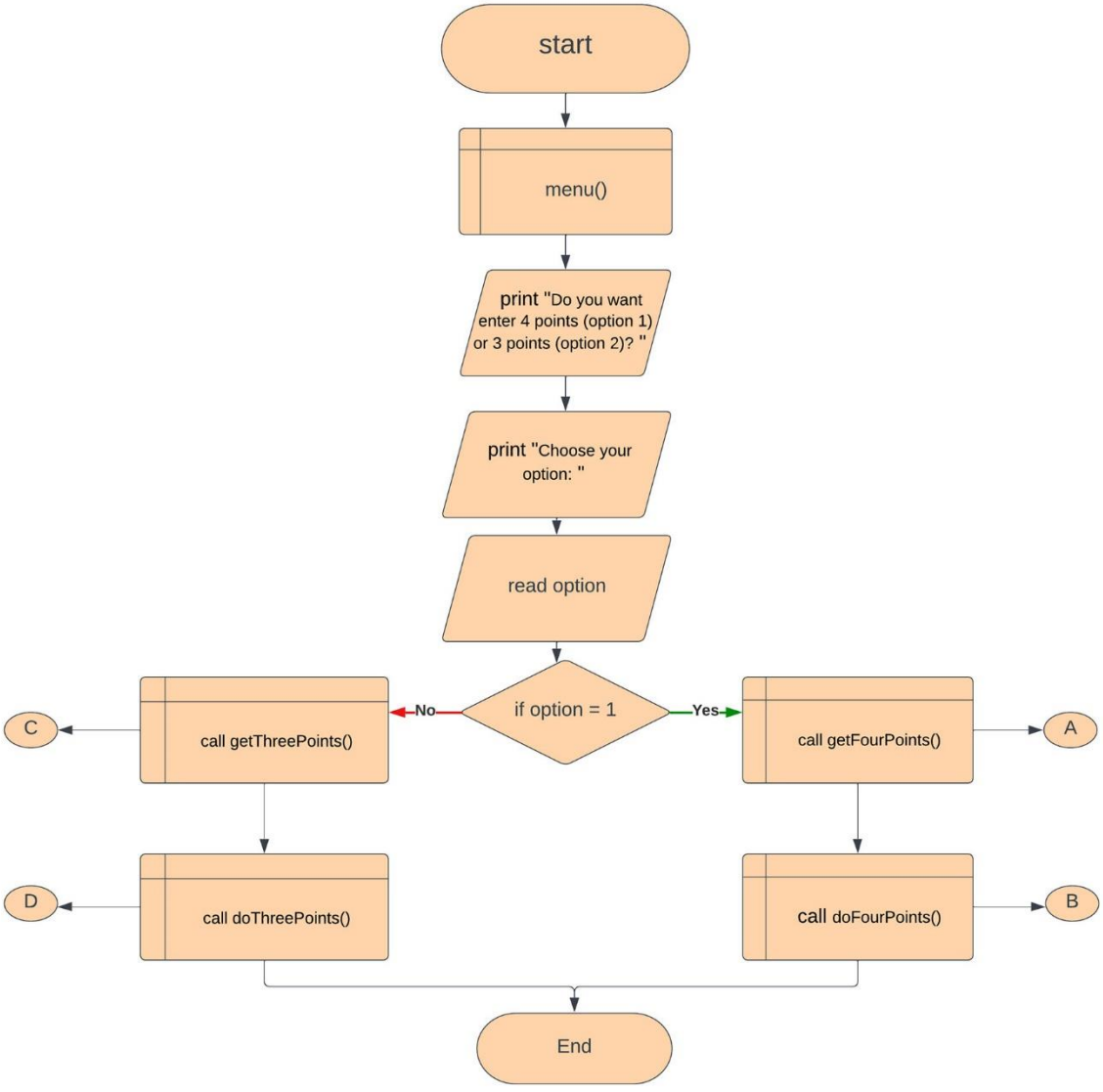
Shape Detector
Arch2 Team

Pseudo Code

```
function Menu(point1[], point2[], point3[], point4[])
  set option : = 0

  print "Do you want enter 4 points (option 1) or 3 points (option 2)? "
  print "Choose your option: "
  input option

  if (option = 1)
    call: getFourPoints(point1, point2, point3, point4)
    call : doFourPoints(point1, point2, point3, point4)

  else
    call: getThreePoints(point1, point2, point3)
    call : doThreePoints(point1, point2, point3)
  endIf
EndFunction

main

  call: Menu(point1, point2, point3, point4)

endMain
```

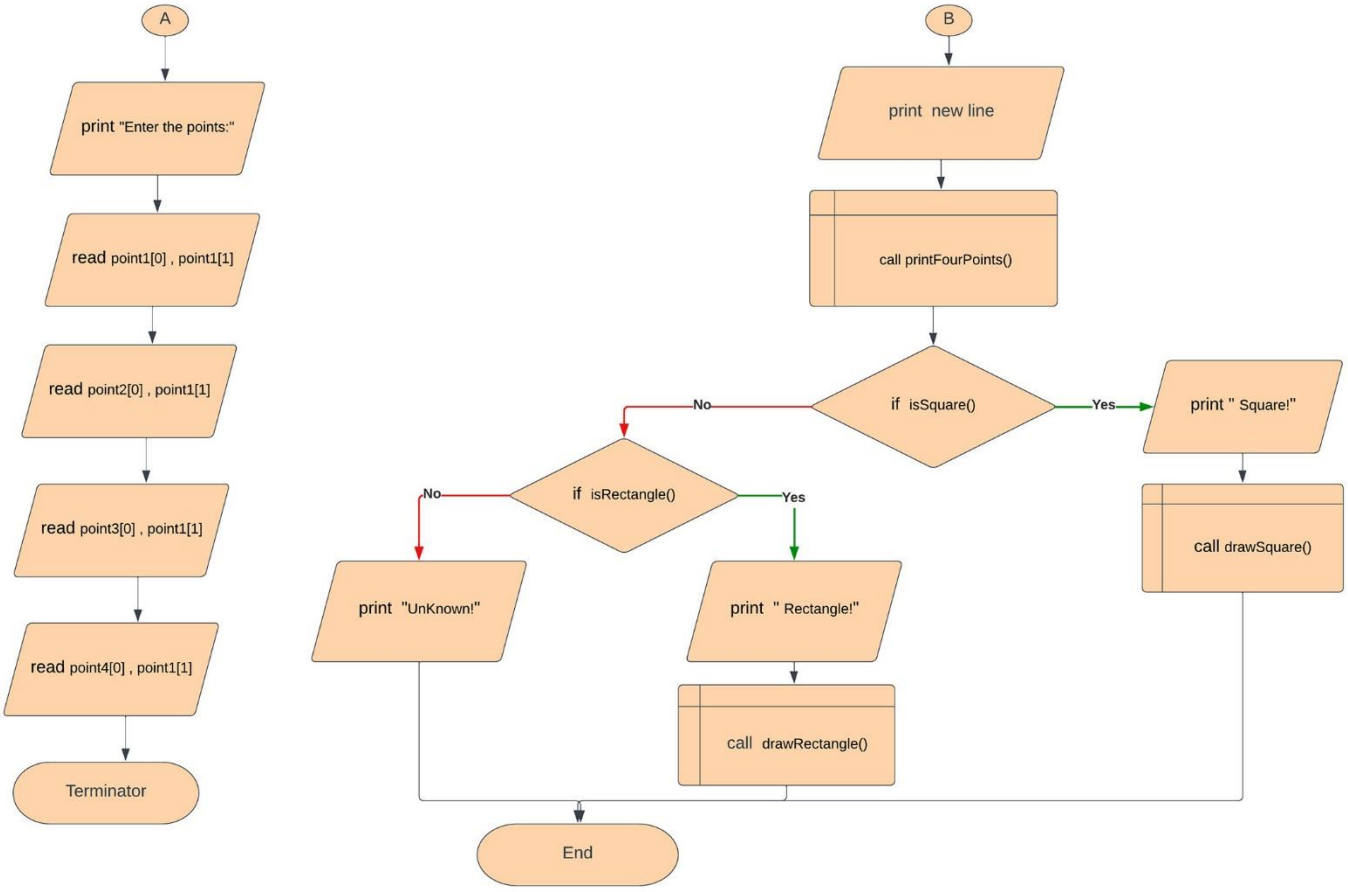
Shape Detector
Arch2 Team

Flowchart



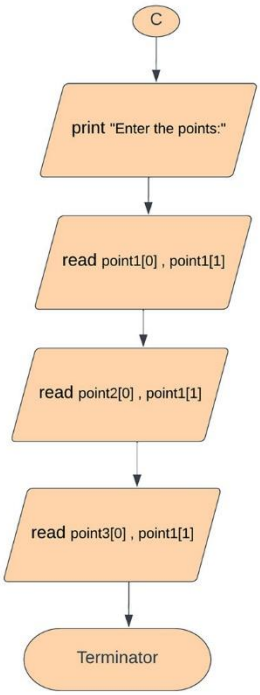
Shape Detector
Arch2 Team

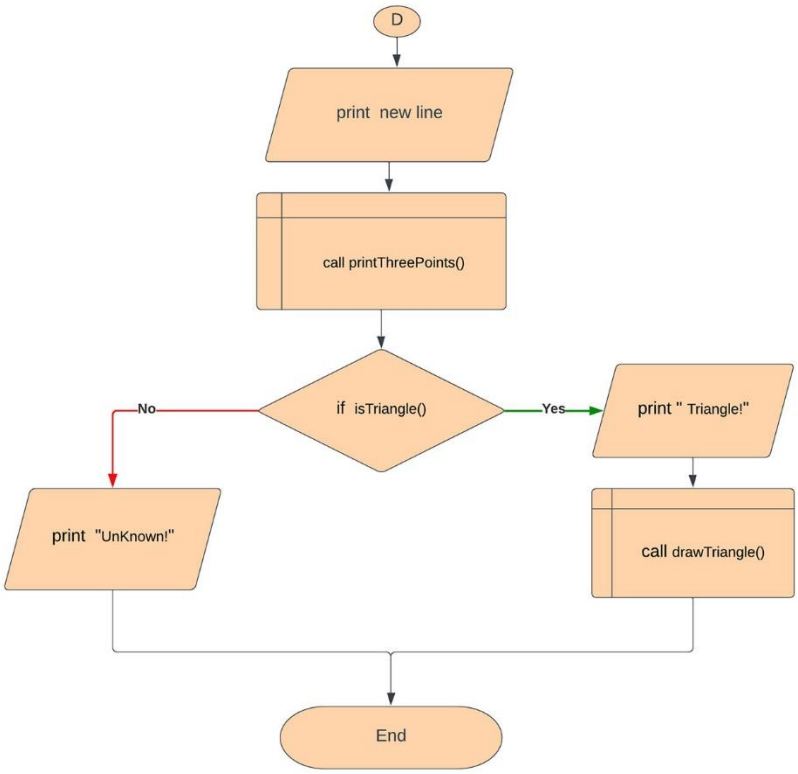
Flowchart



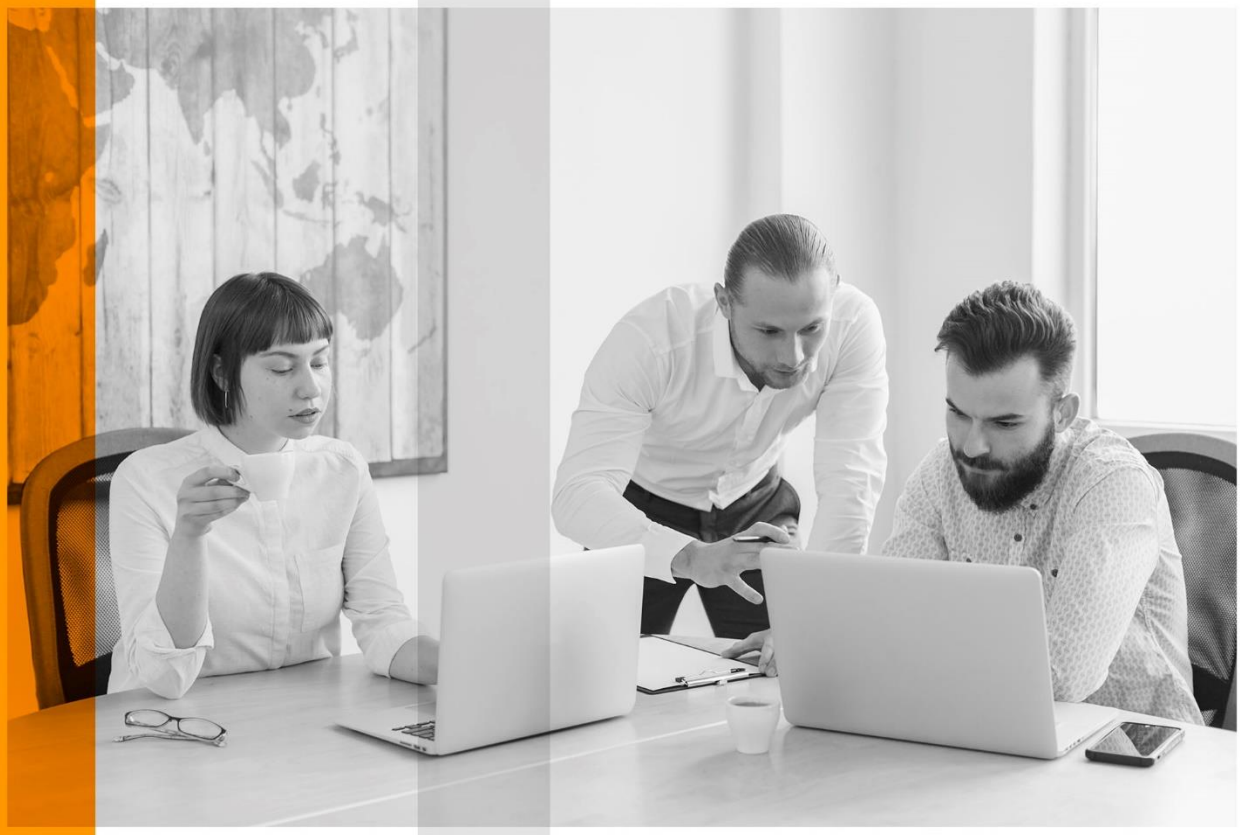
Shape Detector
Arch2 Team

Flowchart





OUR TEAM



Team Name

Arch2 Team

Supervision

Dr. Hewida Youssri
Eng. Nedaa Hussein
Eng. Mahmoud Mohammad

Our Team

Mohamed Shaban Ahmed
Amany Mohamed Sayed
Aya Hassan Kamal
Madonna Hany
Feby Hanna Dawod
Fatma Akram Hassan
Nourhan Khaled Gomaa