



## -:: Sistemas de Coordenadas 2D ::-

O sistema de coordenada no espaço bidimensional (2D) são extremamente importantes para o desenvolvimento de games em *engines* como a Unity.

O sistema cartesiano é usado para localizar e representar elementos em um plano, o qual é formado por dois eixos perpendiculares: um horizontal e outro vertical que se cruzam na origem das coordenadas. O eixo horizontal é chamado de abscissa (x) e o vertical de ordenada (y). Os eixos são enumerados compreendendo o conjunto dos **números reais**. Para indicar a localização de um **ponto** no plano, primeiro identifica-se a sua coordenada no eixo x e, em seguida, no eixo y. Por padrão as coordenadas são representadas por pares ordenados (x,y). A Figura 1 traz uma representação de um plano cartesiano com vários pontos contidos nele.

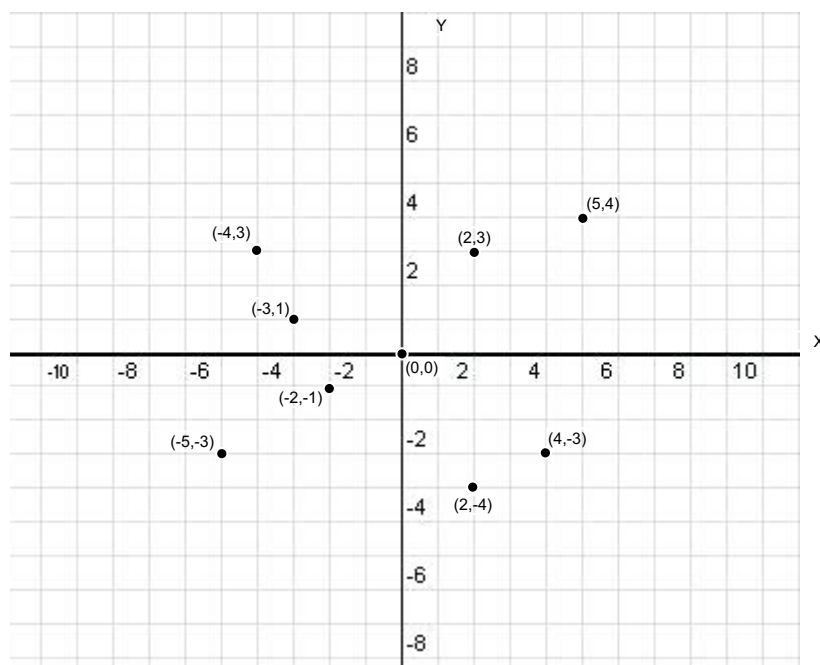


Figura 1 Plano cartesiano

O plano é dividido em 4 partes denominados quadrantes. No **1º quadrante**, a abscissa e a ordenada possui números positivos, portanto o ponto **(2,3)** está no **1º quadrante**. Então, tendo definido o primeiro quadrante, vamos seguindo o sentido anti-horário para definir os restantes. O ponto **(-3,1)** está no **2º quadrante**, ao qual nota-se que a abscissa tem valores negativos e a ordenada positivos. No **3º quadrante** temos o ponto **(-2,-1)** onde tanto a abscissa quanto a ordenada possuem valores negativos. Por fim, temos o ponto **(4,-3)** que está no **4º quadrante**, para o qual a abscissa é positiva e a ordenada é negativa. Observe ainda que a origem das coordenadas está no ponto **(0,0)**.



## Objetivos do trabalho

### Parte A

Em algumas *engines* é muito comum transladar a origem das coordenadas e que, por conseguinte, deslocam também os quadrantes. Um dos objetivos desse trabalho é escrever um programa que a partir da origem translada (X,Y) do sistema de coordenadas cartesianas e uma quantidade determinada de pontos, informa para cada ponto qual é o seu quadrante transladado e quantos pontos estão em cada quadrante.

A entrada de dados ao programa é realizada através do teclado, atendendo a seguinte configuração:

- No início do programa são lidos dois números inteiros, X e Y, representando as coordenadas da **origem transladada**.
- Depois é informado um inteiro **N** indicando a quantidade de pontos que deverão ser lidos e processados pelo programa.
- Em seguida, serão lidos os valores das coordenadas dos **N** pontos pelo programa.

Para cada ponto lido o programa deve informar, imprimindo no console, a qual quadrante translado o ponto pertence. Caso o ponto esteja na abscissa transladada ou ordenada transladada o programa imprimirá “sobre o eixo de coordenadas”.

Ao final do processamento individual dos pontos, o programa também deve imprimir as seguintes informações:

- O **ponto de menor distância** em relação a origem translada e o **valor da distância euclidiana**.
- O **ponto de maior distância** em relação a origem translada e o **valor da distância euclidiana**.
- Quantos pontos existem em cada um dos quadrantes. Não considerar os pontos que estão na abscissa transladada ou ordenada transladada na contagem.

Para Figura 2 abaixo, se o a origem é transladada para **(1,-3)** teríamos o seguinte plano cartesiano:

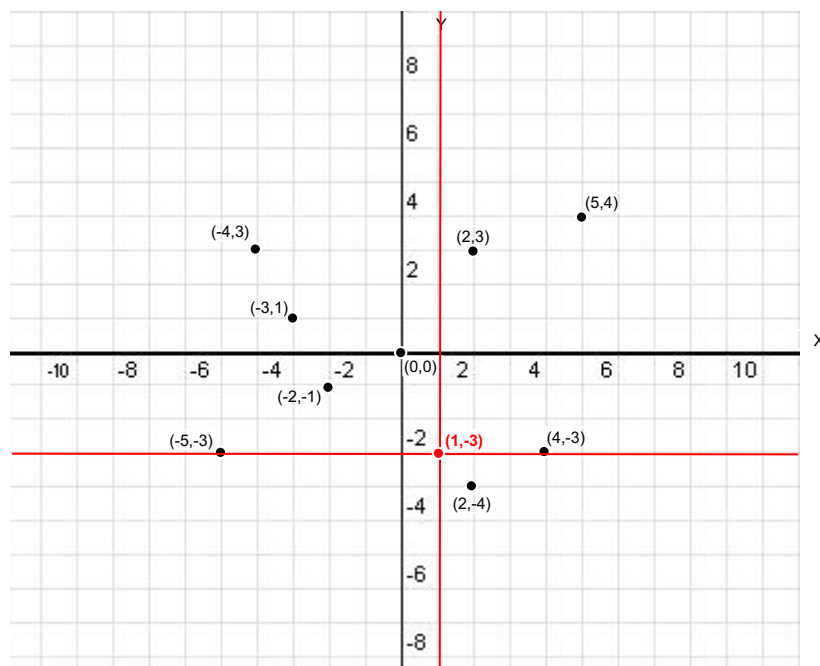


Figura 2 Plano cartesiano com origem translada



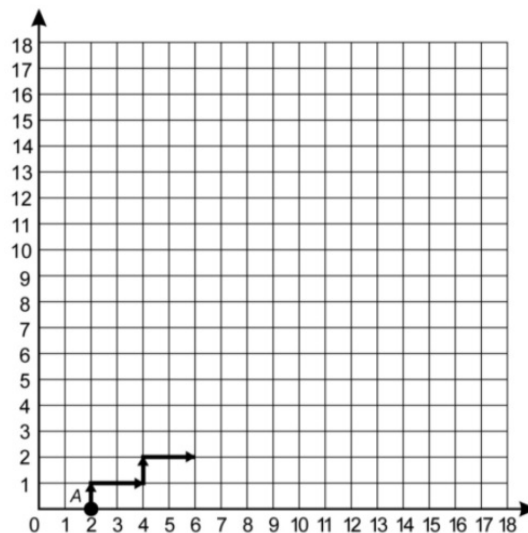
Se forem processados os 8 pontos acima com a origem transladada para (1,-3), teríamos a seguinte saída:

Ponto ( 5, 4) esta no 1o quadrante.  
Ponto ( 2, 3) esta no 1o quadrante.  
Ponto (-4, 3) esta no 2o quadrante.  
Ponto (-3, 1) esta no 2o quadrante.  
Ponto (-2,-1) esta no 2o quadrante.  
Ponto (-5,-3) esta sobre o eixo de coordenadas.  
Ponto ( 4,-3) esta sobre o eixo de coordenadas.  
Ponto ( 2,-4) esta no 4o quadrante.  
Ponto (2,-4) eh o mais proximo, distancia=1,41.  
Ponto (5,4) eh o mais distante, distancia=8,06.  
Existe(m) 2 ponto(s) no 1º quadrante; 3 no 2º quadrante; 0 no 4º quadrante e 1 no 5º quadrante.

Na implementação do seu programa você deve empregar os conceitos relacionados a disciplina atual e tópicos vistos até o momento (operadores, tipos, variáveis, estruturas condicionais e de repetição, etc) não podendo fazer uso de bibliotecas adicionais aos recursos da linguagem ou que ofereçam funcionalidades prontas para armazenamento e cálculos.

### Parte B

O gráfico abaixo ilustra o início do deslocamento de um robô que parte do ponto A (2, 0), movimentando-se para cima ou para a direita, com velocidade de uma unidade de comprimento por segundo, no plano cartesiano. Observe que é um movimento programado: uma unidade para cima, duas para direita, uma para cima, duas para direita e assim sucessivamente.



Este gráfico exemplifica a trajetória desse robô durante os primeiros 6 segundos.

Após seu programa imprimir as informações da Parte A, ele irá pedir ao usuário que informe a coordenada de origem do robô (ponto A) e por quanto tempo ele irá caminhar. Com esses dados, seu programa deverá imprimir qual será sua coordenada final no plano cartesiano. Considerar o plano cartesiano normal, com origem (0, 0).

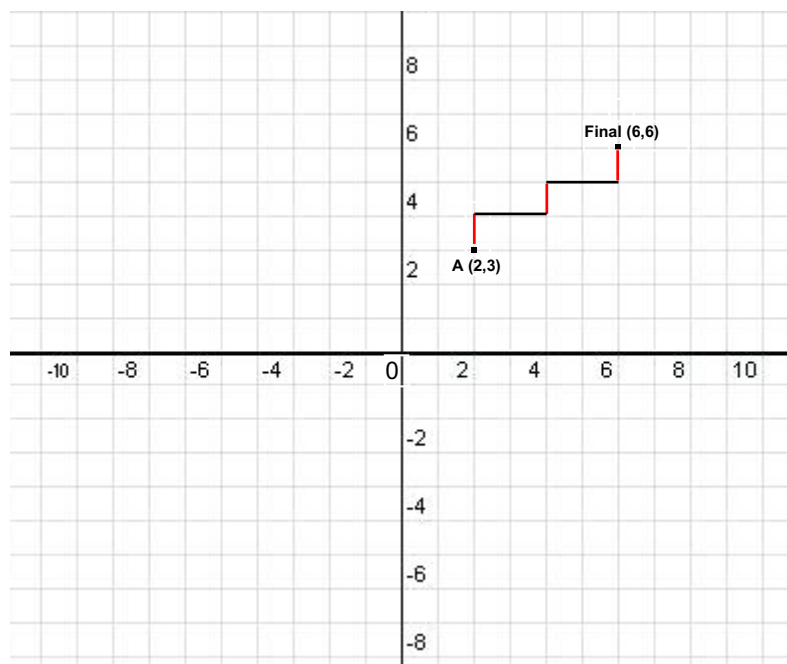


**Exemplo:**

Digite a coordenada X do ponto de origem A do robô: 2  
Digite a coordenada Y do ponto de origem A do robô: 3  
(Neste caso o robô estaria partindo da origem A (2,3))

Digite por quanto tempo o robô irá caminhar: 7

**Resposta:** ao final da caminhada o robô estará no ponto (6,6) do plano cartesiano.



**Observações importantes e considerações finais:**

O programa deve ser implementado na linguagem Python e estar bem documentado. A entrega do trabalho deve ser feita pelo **Moodle**, na entrada especificada e observando-se a data limite para entrega.

O código entregue será avaliado de acordo com os seguintes critérios:

- Funcionamento do programa;
- O quão fiel é o programa quanto à descrição do enunciado;
- Indentação, comentários e legibilidade do código;
- Clareza na nomenclatura de variáveis;

Este trabalho deve ser desenvolvido individualmente ou em duplas, observando durante o processo seguir as orientações contidas no documento “**Orientações para Desenvolvimento de Trabalhos Práticos**”.



## Resolução:

```
print("PARTE A")
print()
print()

#ParteA
import math
cont = 0

#Definir origens
origemx = int(input("Ponto de origem x: "))
origemy = int(input("Ponto de origem y: "))

#Definir quantidade de coordenadas
ncoordenadas = int(input("Quantidade de coordenadas: "))

#Variáveis para definir o ponto mais perto e o mais distante da origem
distminima = math.inf
distmaxima = 0
pmaximo = pminimo = ""

#Determinando o início dos contadores de ponto em cada um dos quadrantes
quadrante1 = 0
quadrante2 = 0
quadrante3 = 0
quadrante4 = 0

#Coleta dos valores das coordenadas
for q in range(ncoordenadas):
    cont = cont + 1
    pontox = int(input(f"Valor do {cont}º ponto X: "))
    pontoy = int(input(f"Valor do {cont}º ponto Y: "))

#Definição de qual ponto estará em qual quadrante
    if pontox != origemx and pontoy != origemy:
        p = f"({pontox},{pontoy})"
        quadrante = ""

#Verificando o(s) quadrante(s) de cada coordenada
    if pontox > origemx and pontoy > origemy:
        quadrante1 = quadrante1 + 1
        quadrante = "1º"

    elif pontox < origemx and pontoy > origemy:
        quadrante2 = quadrante2 + 1
        quadrante = "2º"

    elif pontox < origemx and pontoy < origemy:
        quadrante3 = quadrante3 + 1
        quadrante = "3º"
```



```
elif pontox > origemx and pontoy < origemy:
    quadrante4 = quadrante4 + 1
    quadrante = "4°"

#Definindo qual é o ponto mais próximo e o mais distante da origem transladada
d = math.sqrt((pontox - origemx)**2 + (pontoy - origemy)**2)
if d > distmaxima:
    distmaxima = d
    pminimo = p

elif d < distminima:
    distminima = d
    pmaximo = p
print(f"O ponto {p} está no {quadrante} quadrante.")
else:
    if pontox == origemx and pontoy == origemy:
        print(f"O ponto ({pontox},{pontoy}) é congruente à origem.")
    else:
        if pontox == origemx:
            print(f"O ponto ({pontox},{pontoy}) está sobre o eixo de coordenadas y.")
        elif pontoy == origemy:
            print(f"O ponto ({pontox},{pontoy}) está sobre o eixo das coordenadas x.")

#Imprimindo as informações finais
print(f"O ponto {pmaximo} é o ponto mais distante da origem, distância = {distmaxima:.2f}")
print(f"O ponto {pminimo} é o ponto mais próximo da origem, distância = {distminima:.2f}")
print(f"Existe(m) {quadrante1} ponto(s) no 1° quadrante; {quadrante2} no 2° quadrante; {quadrante3} no 3° quadrante e {quadrante4} no 4° quadrante.")
print()
print()
print("PARTE B")
print()
print()

#ParteB

#Coordenada de origem
coordx = int(input("Digite a coordenada X do ponto A do robô: "))
coordy = int(input("Digite a coordenada Y do ponto A do robô: "))

#Tempo em que o robô será acionado
tempo= int(input("Digite por quanto tempo o robô irá caminhar:"))

#Cálculo de coordenadas
passos = 1
tempo2 = 1
y = 0
x = 0

while tempo2 <= tempo:
    if passos == 1:
```



```
y = y + 1
passos = passos + 1

elif passos == 2:
    x = x + 1
    passos = passos + 1

elif passos == 3:
    x = x + 1
    passos = passos - 2
    tempo2 = tempo2 + 1

#definição dos pontos finais
xfinal = coordx + x
yfinal = coordy + y

print()
print("Resultado final:")
print()

#Apresentação do resultado final
print (f"Ao final da caminhada o robô estará no ponto {xfinal, yfinal} do plano
cartesiano")
```