

Trabalho de Programação 1  
Processador RAMSES

## 1. Descrição Geral

A empresa AGRICART – Agrimensura & Cartografia S.A. – fornece serviço de medição de campos de agricultura, principalmente para fins de agricultura de precisão. Para isso, é fundamental que a empresa possa determinar a área de certo campo de agricultura, cujos limites foram obtidos através do uso de GPS – *Global Position System*.

Nesse trabalho, você deverá desenvolver, para a empresa, o algoritmo de cálculo da área de um campo delimitado pelos pontos fornecidos pelo GPS. Além disso, para demonstrar para a empresa que seu algoritmo funciona, este deverá ser implementado no processador RAMSES.

As informações do GPS sobre os limites da área a ser calculada serão fornecidas através de uma série de pontos que correspondem aos vértices de um “*polígono simples*” (<http://en.wikipedia.org/wiki/Polygon>) com até 32 lados (portanto, até 32 pontos, cada um deles com coordenadas “X” e “Y”).

As coordenadas dos vértices do polígono serão formadas por números inteiros entre “0” (zero) e “15” (quinze), de tal forma que o polígono estará totalmente contido no primeiro quadrante do sistema de coordenadas.

## 2. Especificação do Trabalho

Para calcular a área do polígono serão fornecidas as seguintes variáveis:

- **NumVertices:** número de vértices que formam o polígono. Representado por 1 byte sem sinal, no endereço HBD (189). Esse valor é um número maior ou igual que 3 e menor ou igual que 32;
- **VetPontos:** vetor com os pontos que formam o polígono, que tem um mínimo de 3 elementos e um máximo de 32 elementos. Cada vértice será representado por um par de valores, correspondentes às coordenadas “X” e “Y” (armazenados nessa ordem). Cada coordenada será representada por 1 byte sem sinal. Esse vetor inicia no endereço HBE (190) e vai até o endereço HFD (253).

O resultado do cálculo (a área do polígono) deverá ser colocado na seguinte variável:

- **Area:** valor da área, com 16 bits, sem sinal. O byte mais significativo dessa área deverá ser colocado no menor endereço e o menos significativo no maior endereço. Os endereços ocupados por essa variável são HFE (254) e HFF (255).

Na figura 1(a) está representada uma parte do sistema de coordenadas, onde aparecem 10 colunas (eixo das coordenadas “X”) e 9 linhas (eixo das coordenadas “Y”). Na figura 1(b) estão representados com linhas vermelhas os lados de um “*polígono simples*” (<http://en.wikipedia.org/wiki/Polygon>).

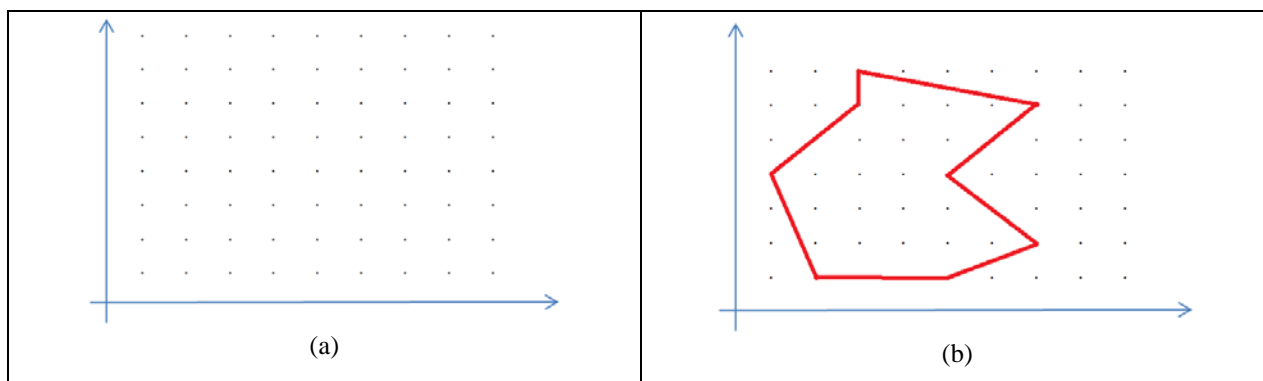


Figura 1 – Sistema de coordenadas

No exemplo que aparece na figura 1(b) os pontos que delimitam o polígono cuja área deve ser calculada são:

(1,4) - (3,6) - (3,7) - (7,6) - (5,4) - (7,2) - (5,1) - (2,1)

As variáveis de entrada, que representam esse polígono, seus endereços e dados, são os seguintes (nesse exemplo, o conteúdo dos endereços 206 até 253 não são significativos, pois o polígono é formado por 8 pontos):

Descrição	Endereço	Dados
NumVertices	HBD (189)	H08 (8)
VetPontos[0].x	HBE (190)	H01 (1)
VetPontos[0].y	HBF (191)	H04 (4)
VetPontos[1].x	HC0 (192)	H03 (3)
VetPontos[1].y	HC1 (193)	H06 (6)
VetPontos[2].x	HC2 (194)	H03 (3)
VetPontos[2].y	HC3 (195)	H07 (7)
VetPontos[3].x	HC4 (196)	H07 (7)
VetPontos[3].y	HC5 (197)	H06 (6)
VetPontos[4].x	HC6 (198)	H05 (5)
VetPontos[4].y	HC7 (199)	H04 (4)

Descrição	Endereço	Dados
VetPontos[5].x	HC8 (200)	H07 (7)
VetPontos[5].y	HC9 (201)	H02 (2)
VetPontos[6].x	HCA (202)	H05 (5)
VetPontos[6].y	HCB (203)	H01 (1)
VetPontos[7].x	HCC (204)	H02 (2)
VetPontos[7].y	HCD (205)	H01 (1)
...		
...		
...		
Area (MSByte)	HFE (254)	?
Area (LSByte)	HFF (255)	?

As coordenadas dos vértices do polígono serão fornecidas como uma sequência contígua de pontos, de tal forma que cada dois vértices contíguos formam um dos lados do polígono. Essa sequência de vértices pode ser fornecida no sentido horário ou anti-horário. Notar, também, que o último ponto da lista forma um lado do polígono com o primeiro ponto fornecido: no caso do exemplo este lado do polígono é formado pelo o último ponto (2,1) e o primeiro ponto (1,4).

### 3. Algoritmo: básico

Para calcular a área dentro do polígono poderá ser usado o seguinte procedimento:

- Os lados devem ser representados como vetores, de tal forma que a ponta de cada vetor estará ligado à base do vetor seguinte, conforme mostrado na figura 2;

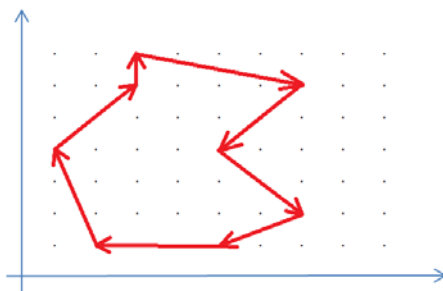
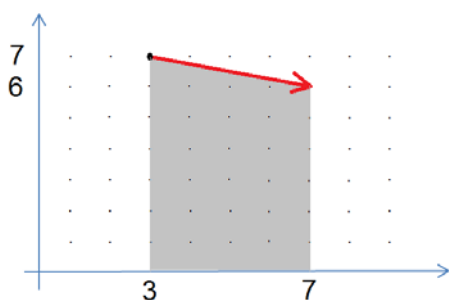


Figura 2 – Representação dos lados como vetores

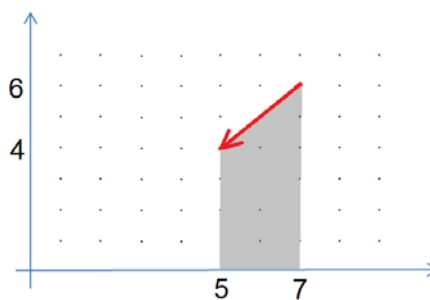
- Para cada vetor será calculada a área sob o mesmo. Para isso, considera-se que os pontos “base” e “ponta” do vetor são representados por (bx,by) e (px,py), respectivamente, e a expressão da área é dada por:

$$Area = (px - bx) \cdot \left( \frac{py + by}{2} \right)$$

- Notar que a expressão pode fornecer valores positivos ou negativos para a área, dependendo se o vetor está apontando na mesma direção do eixo das abcissas (eixo “X”) ou no sentido contrário. Na figura 3 estão representados dois vetores: um na mesma direção do eixo e outro em sentido contrário.



Área (positiva) = (7-3)\*(7+6)/2 = 26



Área (negativa) = (5-7)\*(4+6)/2 = -10

Figura 3 – Exemplo de cálculo da área abaixo de um dos lados

4. Após calculadas as áreas sob todos os vetores, elas devem ser somadas (considerando-se o sinal das mesmas). O resultado dessa soma corresponderá à área interna do polígono.

#### 4. Exemplo de cálculo

No caso do polígono de exemplo, as áreas parciais serão:

$$A_{01} = (3-1)*(6+4)/2 = +10$$

$$A_{12} = (3-3)*(7+6)/2 = 0$$

$$A_{23} = (7-3)*(6+7)/2 = +26$$

$$A_{34} = (5-7)*(4+6)/2 = -10$$

$$A_{45} = (7-5)*(2+4)/2 = +6$$

$$A_{56} = (5-7)*(1+2)/2 = -3$$

$$A_{67} = (2-5)*(1+1)/2 = -3$$

$$A_{70} = (1-2)*(4+1)/2 = -2,5$$

Somando-se as áreas parciais, obtêm-se **Área = 23,5**.

Ainda, deve-se observar dois fatos:

1. Se os vetores que formam os lados estiverem no sentido horário, o resultado da soma será um valor positivo; se estiverem no sentido anti-horário, o resultado será um valor negativo. Portanto, como não se sabe o sentido que os pontos foram fornecidos e para determinar a área (que sempre é um valor positivo) deve-se completar o cálculo da área determinando o **valor absoluto da soma final**.
2. As áreas parciais podem ter valores fracionários. Entretanto, na fórmula de cálculo aparece uma divisão por 2: portanto esse valor fracionário só poderá ser “0,5” (por exemplo, verifique a área A70 acima). O resultado a ser fornecido **deve ser arredondado**, de tal forma que a área final será sempre inteira. No caso do exemplo, o valor informado deverá ser 24.

#### 5. Casos de Teste

Na tabela abaixo estão apresentados os casos de teste para os quais os programas devem estar funcionando corretamente, para serem considerados “minimamente operacionais”. Cada linha da tabela representa um caso de teste. Na coluna “NumVertices” está o valor do número de vértices; na coluna “VetPontos” estão listados os pontos que formam o polígono, na ordem que devem estar na memória e na coluna “Area” está o valor correspondente à área do polígono descrito. Notar que todos os valores apresentados na tabela estão em decimal.

Caso	NumVertices	VetPontos	Area
1	4	(5, 5) - (5, 10) - (10, 10) - (10, 5)	25
2	4	(5, 10) - (10, 15) - (15, 10) - (10, 5)	50
3	3	(5, 5) - (10, 5) - (10, 10)	13
4	5	(5, 5) - (15, 5) - (10, 10) - (15, 15) - (5, 15)	75

#### 6. Entregáveis: o que deve ser entregue?

Devem ser entregues dois arquivos:

1. Arquivo fonte em assembler do RAMSES (arquivo .RAD)
2. Arquivo fonte na linguagem “C” (arquivo .C)

A implementação entregue no arquivo .RAD deve seguir, rigorosamente, a especificação do trabalho no que diz respeito à localização na memória das variáveis de entrada e saída.

O programa escrito na linguagem “C” deve ter incorporado comentários das instruções em assembler do RAMSES correspondente a cada linha de codificação em “C”.

O trabalho só será considerado entregue se forem enviados os arquivos acima descritos e o programa estiver **minimamente operacional**, ou seja, deve fornecer os resultados corretos para os casos de teste descritos nessa especificação.

O trabalho deverá ser entregue até a **data prevista**. Não serão aceitos trabalhos entregues além dos prazos estabelecidos.

Os programas considerados funcionais serão corrigidos de forma automática pela aplicação de novos casos de teste e a nota final do trabalho dependerá do número de casos em que o programa produzir a resposta correta.

Dentre os programas que fornecerem os resultados corretos para todos os casos de teste, aqueles mais rápidos (que utilizarem o menor número de acessos a memória) concorrerão a um “bônus” de 10,0 pontos na nota (10%).

## **7. Observações**

---

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplina Discente e a tomada das medidas cabíveis para essa situação.

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.