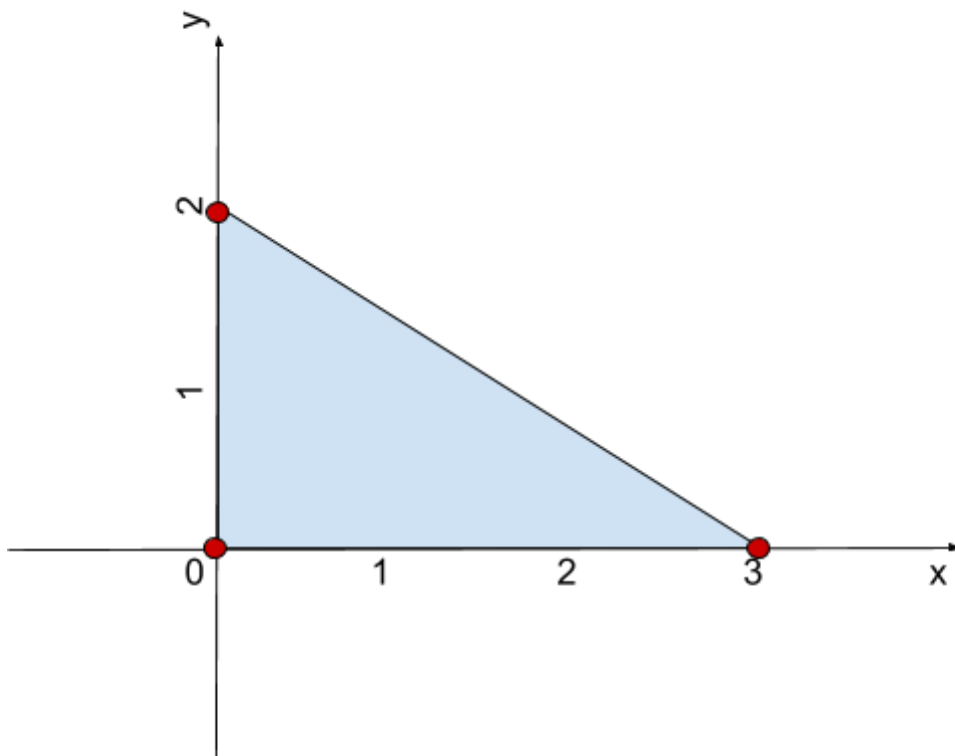
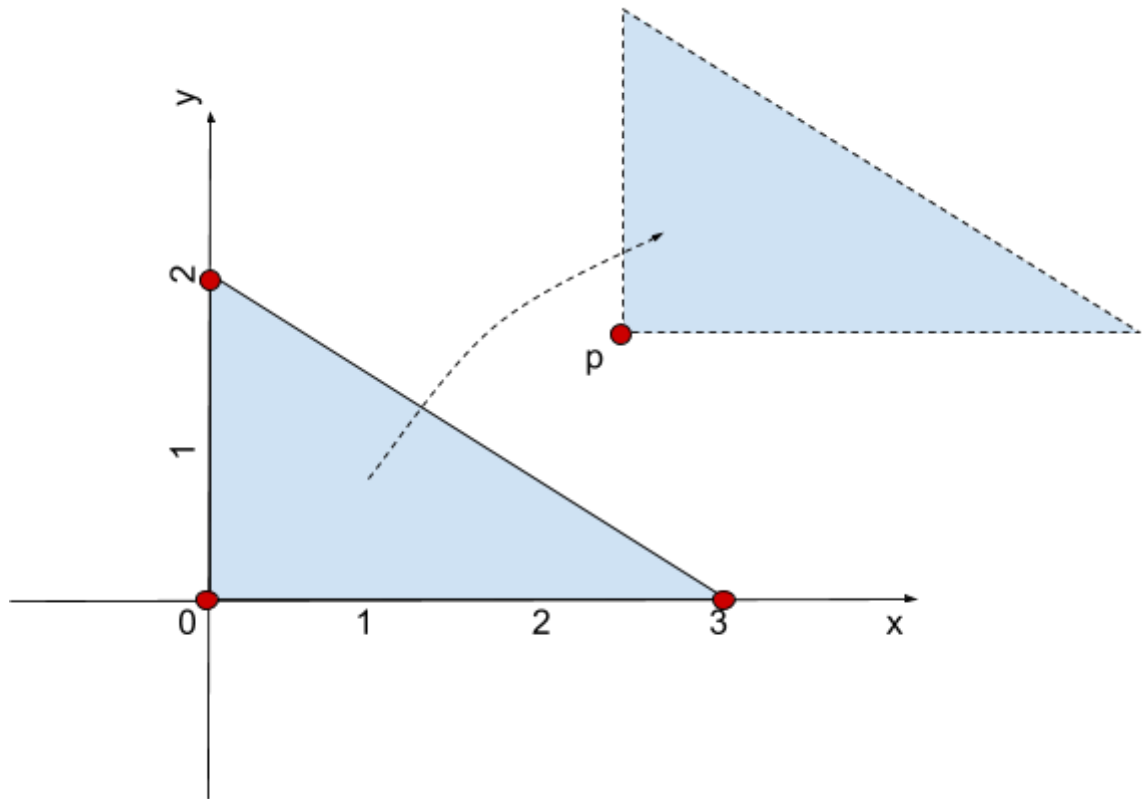


Atividades de revisão para a segunda avaliação

1. Declare um tipo estruturado chamado Ponto com 2 campos (x,y) do tipo double.
2. Declare um tipo estruturado Triangulo, formado por 3 pontos.
3. Implemente uma função que receba um triângulo e calcule seu perímetro.
`double perimetro (Triangulo t)`
4. Implemente uma função que retorne um triângulo retângulo com base e altura adjacentes aos eixos x e y e comprimento b e h recebidos por parâmetro
`Triangulo retangulo (double b, double h)`
Ex.: b=3, h=2.

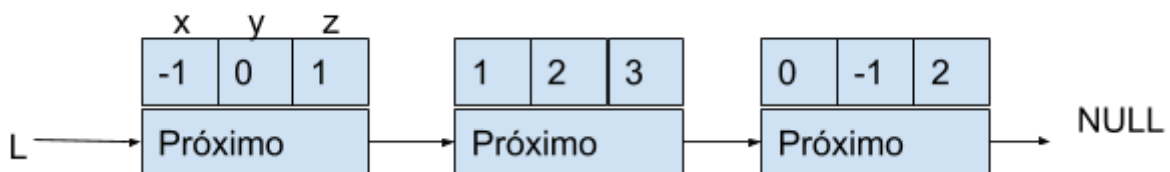


5. Implemente uma função que receba um triângulo e um ponto p e faça a translação do triângulo somando com o ponto p.
`void translacao(Triangulo *t, Ponto p)`
Ex:



- ✓ 6. Declare um tipo estruturado Polígono composto por um vetor de pontos e a quantidade de pontos.
7. Implemente uma função que leia do usuário dados de um polígono de n pontos.

```
Poligono* leia (int n)
```
8. Crie um tipo estruturado Ponto3D com os campos (x,y,z) do tipo double e o ponteiro para o próximo ponto (para ser usado em uma lista encadeada).
9. Implemente uma função que recebe uma lista encadeada de Ponto3D e remove os pontos cujo x,y,z sejam negativos.
10. Faça o teste de mesa do exercício 9. Considerando a lista encadeada abaixo:



11. Considerando as as funções de empilhar e desempilhar já estejam

implementadas e possuem a seguinte assinatura:

```
void empilha (int valor, int pilha[ ], int *quantidade)
```

```
int desempilha (int pilha[ ], int *quantidade)
```

Implemente uma função que faça a leitura de um número inteiro n, até que n seja igual a zero. Se n for negativo, empilhar n na pilha, se n for positivo, desempilhar n valores da pilha e mostrar na tela.

12. Realize o teste de mesa do exercício 11 considerando como entrada:

-4 -8 -5 -3 2 -9 1 -7 2 0

13. Considerando as as funções de enfileirar e desenfileirar já estejam

implementadas e possuem a seguinte assinatura:

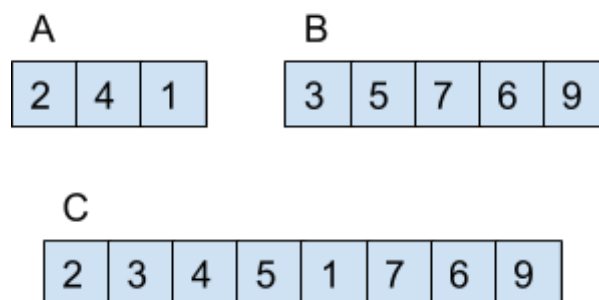
```
void enfileira(int fila[], int valor,
```

```
int *inicio, int* qtd)
```

```
int desenfileira(int fila[], int *inicio, int *qtd)
```

Implemente uma função que receba por parâmetro duas filas A e B e retorne uma fila C (alocada dinamicamente), que seja a união de A e B (intercalado).

Ex.:



14. Realize o teste de mesa do exercício 13, utilizando o exemplo como entrada.