

GRÀFICS PER COMPUTADOR I

PRÀCTICA 1

Operacions amb polígons 2D (retall)

INDEX

- 1.- Objectius
- 2.- Algorisme de Sutherland-Hodgman
- 3.- Càlcul d'interseccions
- 4.- Costats horitzontals i verticals del polígon
- 5.- Exemple
- 6.- Programació en C: Apuntadors
- 7.- Fitxers de la pràctica

OBJECTIUS

- Veure algunes operacions bàsiques dels gràfics en 2D
- Implementació en un llenguatge de programació
Ex.- OpenGL disposa de moltes primitives ja implementades → Com estan implementades?
- En gràfics hi ha dues restriccions bàsiques:
 - reduir cost de memòria
 - reduir temps de procés



MEMÒRIA DINÀMICA

ALGORISME DE SUTHERLAND-HODGMAN

V = Vèrtexs del polígon

Per a cada recta de la finestra

V' = buit

s = darrer vèrtex de V

Per tots els vèrtexs p de V

Mirem el cas en què ens trobem

Si estem al CAS 1 Afegim "p" a V'

Si estem al CAS 2 No afegim res a V'

Si estem al CAS 3 Afegim "i" a V'

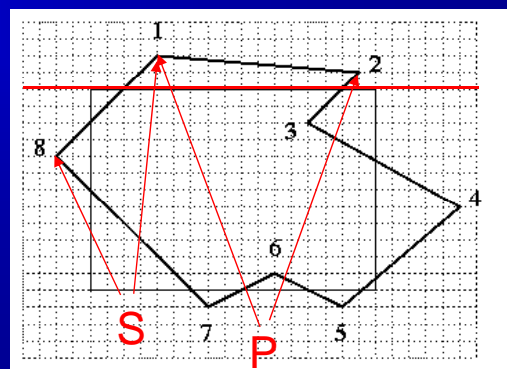
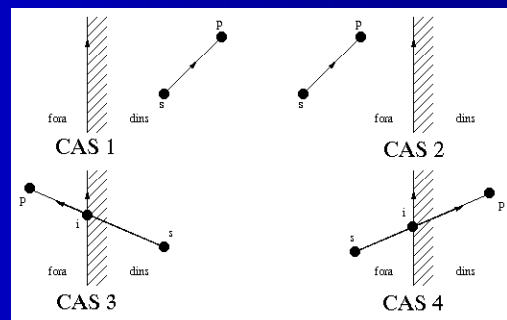
Si estem al CAS 4 Afegim "i" i "p" a V'

$s = p$

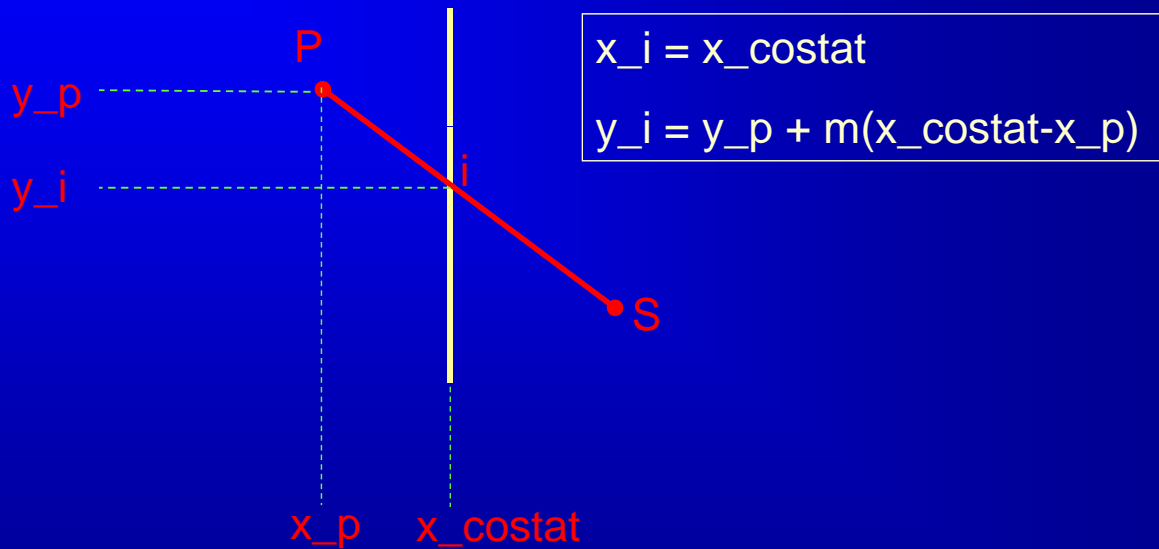
Fi Per

$V = V'$

Fi Per

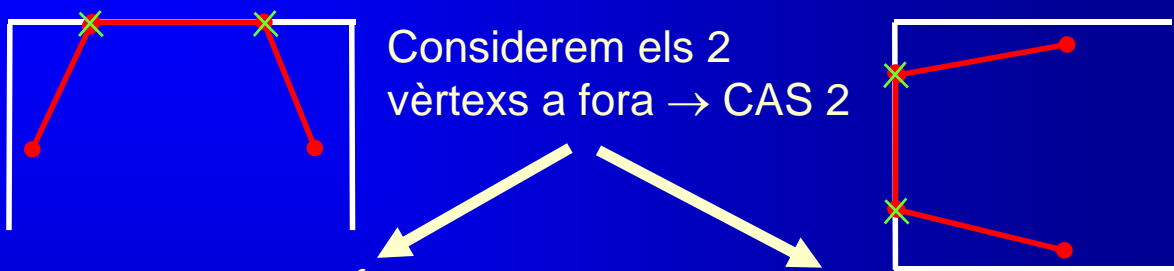


CÀLCUL DE LES INTERSECCIONS



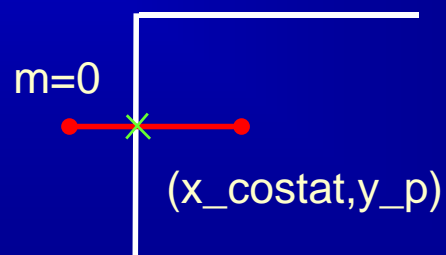
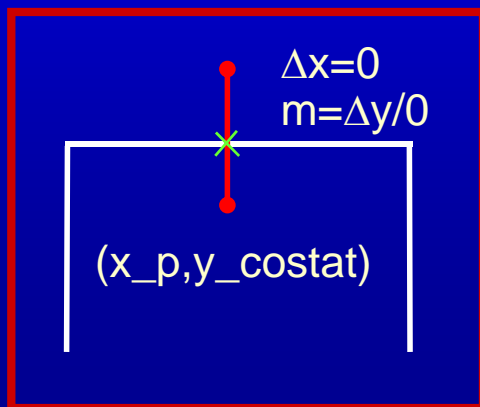
Problema: Què passa amb els costats horitzontals i verticals?

COSTATS HORIZONTALS I VERTICALS

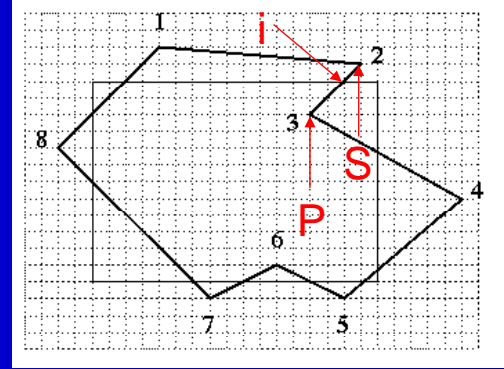
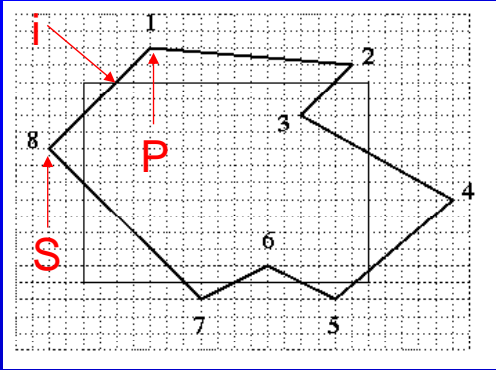


Si $y_p \geq y_{\text{costat}} \rightarrow$ fora
Si $y_p < y_{\text{costat}} \rightarrow$ dins

Si $x_p \leq x_{\text{costat}} \rightarrow$ fora
Si $x_p > x_{\text{costat}} \rightarrow$ dins

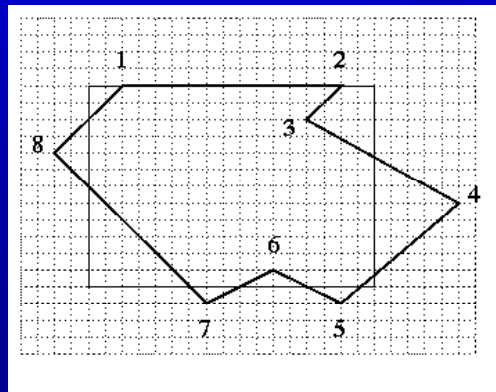


EXEMPLE



Punt Coordenades

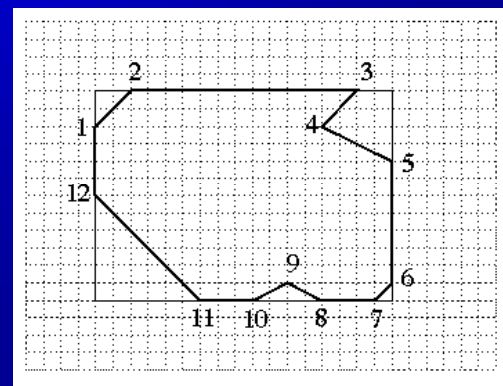
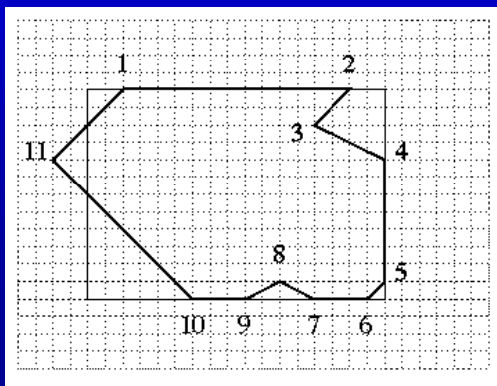
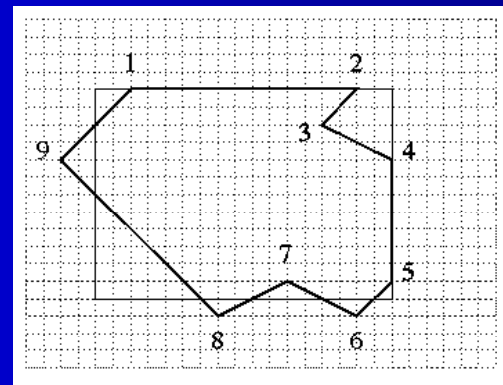
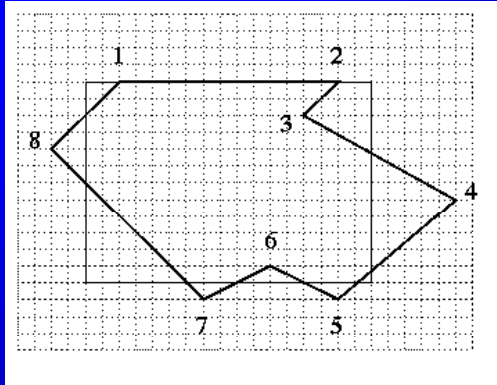
- 1 (80, 20)
- 2 (200, 30)
- 3 (170, 60)
- 4 (260, 110)
- 5 (190, 170)
- 6 (150, 150)
- 7 (110, 170)
- 8 (20, 80)



Punt Coordenades

- 1 (60, 40)
- 2 (190, 40)
- 3 (170, 60)
- 4 (260, 110)
- 5 (190, 170)
- 6 (150, 150)
- 7 (110, 170)
- 8 (20, 80)

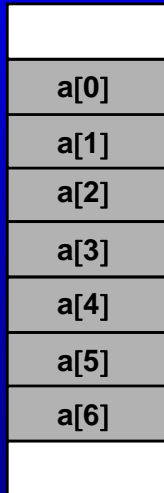
EXEMPLE



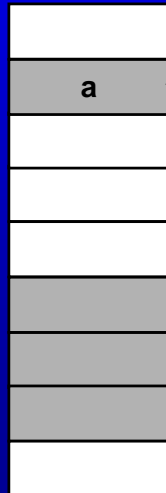
PROGRAMACIÓ EN C: APUNTADORS

APUNTADOR = Adreça de memòria

int a[7]



int *a



a=malloc(3*sizeof(int))

PROGRAMACIÓ EN C: APUNTADORS

Creació d'un polígon = Declaració + reserva memòria

Declaració:

```
Poligon *V; /*figura 1*/
```

Reserva de memòria:

```
V=(Poligon *)malloc(sizeof(Poligon)); /*figura 2*/
```

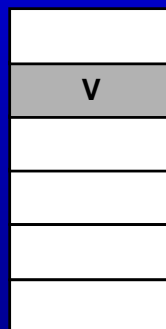


Figura 1

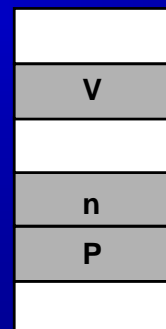


Figura 2

PROGRAMACIÓ EN C: APUNTADORS

També cal reservar memòria per la llista de punts:

```
V→P=(Punt *)malloc(sizeof(Punt)*V→n); /*figura 3*/
```

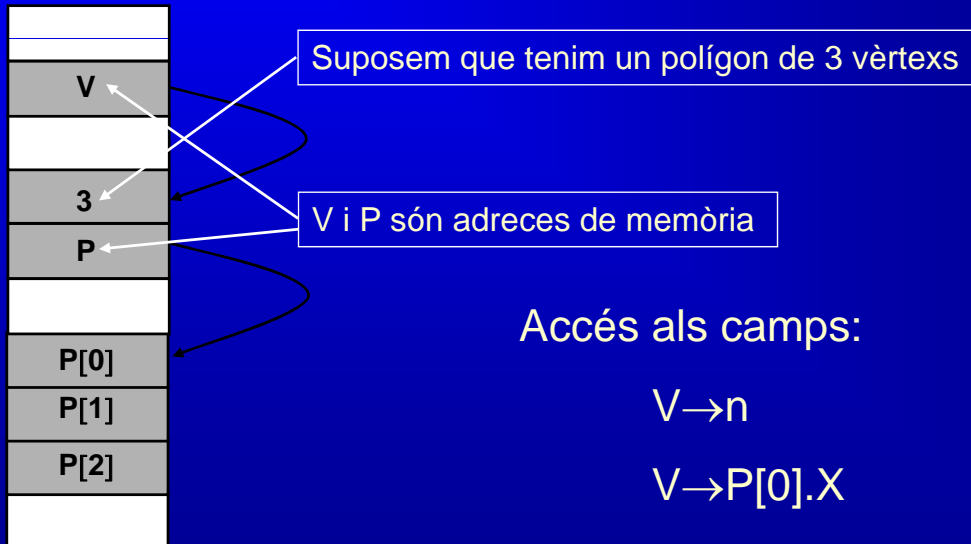
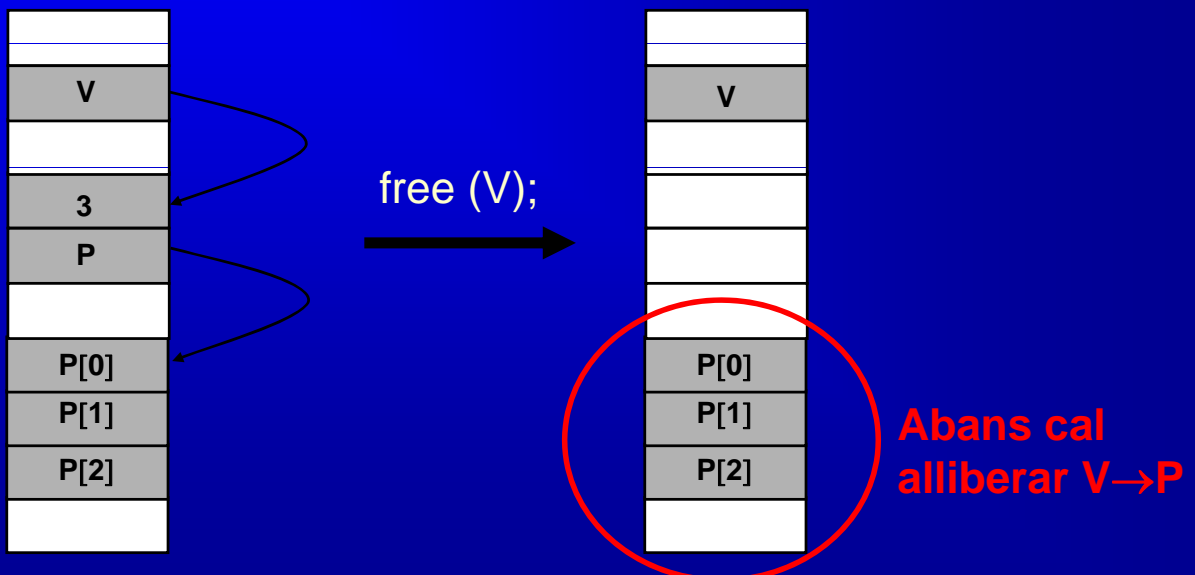


Figura 3

PROGRAMACIÓ EN C: APUNTADORS

En acabar, cal alliberar la memòria reservada:

IMPORTANT: No n'hi ha prou amb alliberar V !!!

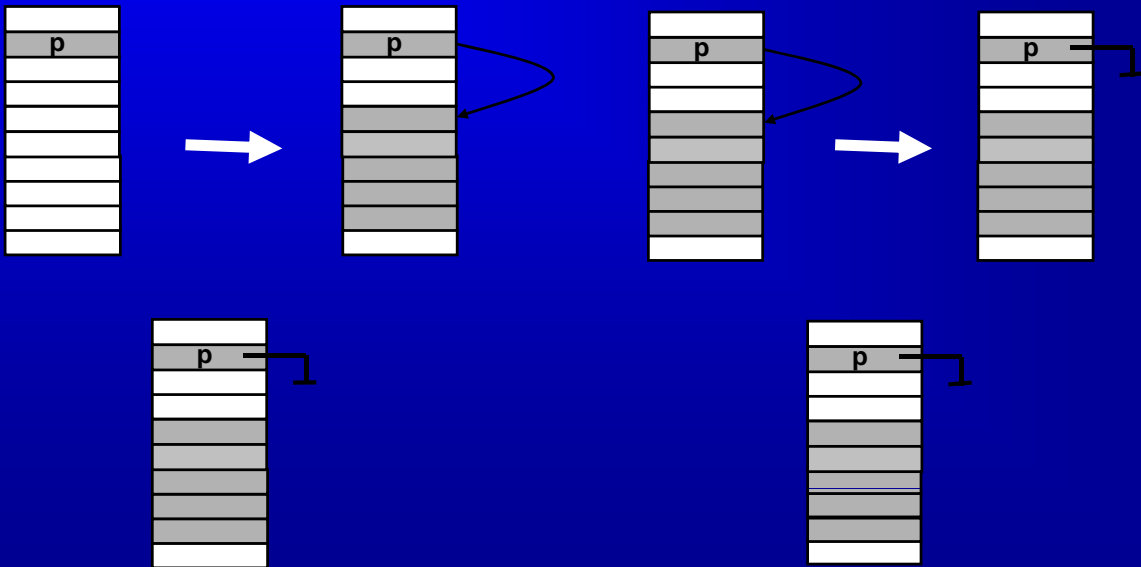


PROGRAMACIÓ EN C: APUNTADORS

Errors comuns:

~~p=malloc(...); /* reserva memoria */
p=null; /* inicializacio */~~

~~p=null,
free(p); /* alliberar memòria */~~



PROGRAMACIÓ EN C

Un altre problema comú → Càlcul del pendent (m)

Operador / → Si els dos operands són sencers, fa la divisió sencera

```
int a=1;  
int b=2;  
float c;  
c=a/b; → c=0!!!!
```

Solució: Conversió a float d'almenys un dels operands:

```
c=(float)a/(float)b; → c=0.5
```

graphics.h

putpixel(int x,int y, int color)

line(int x0, int y0, int x1, int y1)

rectangle(int left, int top, int right, int bottom)

cleardevice() → Esborrar la pantalla

setcolor(int color) → Assignar color

setbkcolor(int color) → Assignar color de fons

getmaxx(), **getmaxy()** → obtenir les coordenades màximes de la pantalla

initgraph(), **closegraph()** → mode gràfic (ja està fet a prac.c)

FITXERS DE LA PRÀCTICA

- Fitxers '.gr1' → Fitxers de comandes

- Fitxers '.pol' → Fitxers de dades (polígons)

- PRAC.C → Intèrpret de comandes

Es poden incloure noves funcions

- Fitxer 'Egavga.bgi' → Drivers de la vga

Important: Aquest fitxer l'heu de tenir al directori on executeu