

wootwoot

### Image Editor

(2 seconds, 512MB)

สร้างโปรแกรม image editor ที่สามารถเก็บรูปได้หลายรูปโดยแต่ละรูปจะมี imageld ของตัวเอง

#### Input

Line 0 : N - จำนวนของ command ที่ต้องการจะรับ

Line [1, N] : <command> <arguments> - command แต่ละ command พร้อม arguments

#### Commands

cim <imageld> <width> <height> - create image

สร้าง image ใหม่ที่มี id = imageld

หากมี image ที่มี imageld ซ้ำให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

imageld : int : [0, 10<sup>9</sup>] - imageld

width : int : [1, 64] - ความกว้างของ image

height : int : [1, 64] - ความสูงของ image

dim <imageld> - delete image

ลบ image ที่มี id = imageld

หากไม่เจอ image ที่มี id = imageld ให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

imageld : int : [0, 10<sup>9</sup>] - imageld

pim <imageld> - print image

แสดงผล image ที่มี id = imageld

หากไม่เจอ image ที่มี id = imageld ให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

imageld : int : [0, 10<sup>9</sup>] - imageld

fim <imageld> <horizontal> <vertical> - flip image

Flip image ตามแนวนอน(horizontal) หรือ แนวตั้ง(vertical) หรือทั้งสองแนว หรือไม่ชักแนว

หากไม่เจอ image ที่มี id = imageld ให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

imageld : int : [0, 10<sup>9</sup>] - imageld

horizontal : bool : 0/1 - flip image ในแนวนอน

vertical: bool : 0/1 - flip image ในแนวตั้ง

wootwoot

ich <imageld> <x> <y> <width> <height> <chunk> - insert chunk

ใส่ buffer คำสี rgb กับ image ที่มี id = imageld

หากไม่เจอ image ที่มี id = imageld ให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

หากอยู่นอกขอบเขตของ chunk หรือ image ให้ไม่ต้องวาดส่วนที่อยู่นอกขอบเขตนั้นๆ

imageld : int : [0, 10<sup>9</sup>] - imageld

x : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดที่ต้องการเริ่ม insert chunk จากมุมซ้ายบน

y : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดที่ต้องการเริ่ม insert chunk จากมุมซ้ายบน

width : int : [1, 64] - ความกว้างของ chunk

height : int : [1, 64] - ความสูงของ chunk

chunk : ข้อมูล chunk ใน format ของ [0, 255] [0, 255] [0, 255] ต่อ 1 pixel เช่น

107 57 70 105 53 70 มี 2 pixel โดย pixel แรกมีคำสี rgb คือ (107, 57, 70) และ pixel ที่สองมีคำสี (105, 53, 70) เป็นต้น

rch <imageld> <x> <y> <width> <height> <colorR> <colorG> <colorB> -  
remove chunk

แทนส่วนของ image ที่มี id = imageld ด้วยคำสี rgb คำเดียว

หากไม่เจอ image ที่มี id = imageld ให้แสดงผลว่า invalid และไปทำคำสั่งถัดไปจนจบ

หากอยู่นอกขอบเขตของ chunk ให้ไม่ต้องวาดส่วนที่อยู่นอกขอบเขตนั้นๆ

imageld : int : [0, 10<sup>9</sup>] - imageld

x : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดที่ต้องการเริ่ม remove chunk จากมุมซ้ายบน

y : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดที่ต้องการเริ่ม remove chunk จากมุมซ้ายบน

width : int : [1, 64] - ความกว้างของ chunk ที่ต้องการ remove

height : int : [1, 64] - ความสูงของ chunk ที่ต้องการ remove

colorR, colorG, colorB : คำสี rgb ใน format ของ [0, 255] [0, 255] [0, 255] ที่ต้องการจะ  
นำไปแทน chunk ที่นำออก

wootwoot

blt <srcImageld> <dstImageld> <srcX> <srcY> <srcWidth> <srcHeight> <dstX>  
<dstY> - blit

ทำการ bit blit หรือการ copy จาก image buffer ของ srcImage ไป dstImage

หากไม่เจอ srcImage ที่มี id = srcImageld ให้แสดงผลว่า srcInvalid และไปทำคำสั่งถัดไปจนจบ

หากไม่เจอ dstImage ที่มี id = dstImageld ให้แสดงผลว่า dstInvalid และไปทำคำสั่งถัดไปจนจบ

หากไม่เจอทั้งคู่ให้แสดงผลว่า bthInvalid และไปทำคำสั่งถัดไปจนจบ

หากอยู่นอกขอบเขตของ srcImage หรือ dstImage ให้ไม่ต้องวาดส่วนที่อยู่นอกขอบเขตนั้นๆ

srcImageld : int : [0, 10<sup>9</sup>] - imageld ของ source

dstImageld : int : [0, 10<sup>9</sup>] - imageld ของ destination

srcX : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดจาก source ที่ต้องการ blit ไปยัง destination จากมุมซ้ายบน

srcY : int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดจาก source ที่ต้องการ blit ไปยัง destination จากมุมซ้ายบน

srcWidth : int : [1, 64] - ความกว้างของส่วนตัดที่ต้องการ blit ไปยัง destination

srcheight : int : [1, 64] - ความสูงของส่วนตัดที่ต้องการ blit ไปยัง destination

dstX: int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดของ destination จากมุมซ้ายบนที่ต้องการเริ่มวางส่วนตัดจาก source

dstY: int : [-10<sup>9</sup>, 10<sup>9</sup>] - จุดของ destination จากมุมซ้ายบนที่ต้องการเริ่มวางส่วนตัดจาก source

การแสดงผล pixel ที่มีสีภายใน console/terminal

สามารถทำได้โดยการ output ตัวอักษรดังนี้

"\033[48;2;<r>;<g>;<b>m \033[0m"

โดยแทน <r> <g> <b> ด้วยค่าสี rgb แต่ละค่ามี range ตั้งแต่ [0, 255] อย่างเช่น

"\033[48;2;255;0;0m \033[0m"

จะเป็นการแสดงผล pixel สีแดง 1 pixel

wootwoot

### ตัวอย่าง input output

Input:

11

cim 0 34 43

cim 1 44 24

ich 0 0 0 34 43

<imageData0>

ich 1 0 0 44 24

<imageData1>

fm 1 1 1

blt 1 0 25 0 20 100 20 10

rch 0 0 0 5 5 0 255 255

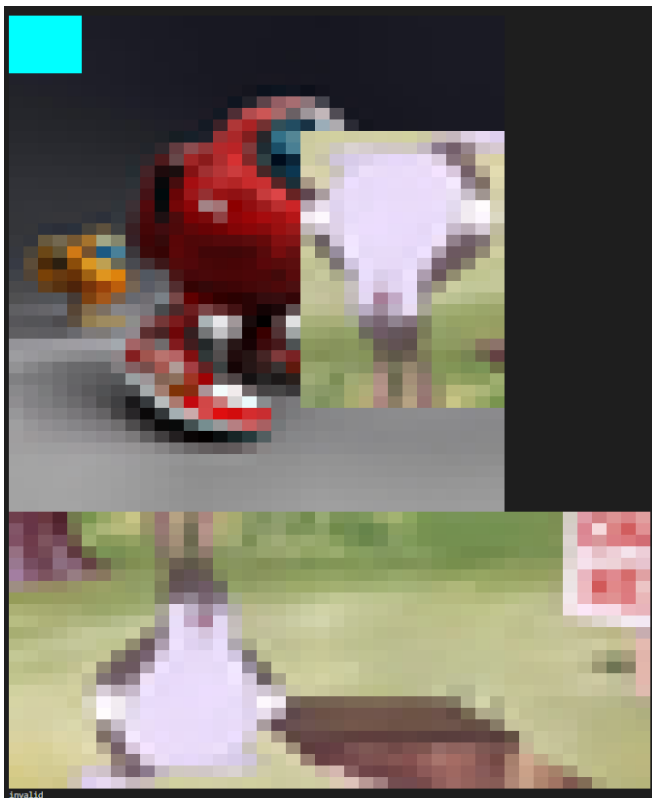
fm 1 1 1

pim 0

pim 1

dim 42

Output:



ในส่วนของ Input ตัวอย่าง ให้ใช้ไฟล์ input ที่แนบกับโจทย์แทน เพราะจะมีข้อมูลของ <imageData0> และ <imageData1> ด้วย