

Digital Wallet

(1 sec, 512mb)

โจทย์ข้อนี้คือส่วนที่ 3 ของข้อสอบ นิสิตสามารถเลือกทำโจทย์ข้อนี้โดยการเขียนในกระดาษได้ โดยให้ปฏิบัติตามคำแนะนำด้านท้ายโจทย์

รัฐบาลแห่งประเทศไทยได้ออกนโยบายกระตุ้นเศรษฐกิจโดยการแจกเงินผ่าน Digital Wallet โดยประชากรจะต้องติดตั้ง app เพื่อใช้เงินที่ได้รับแจกนี้ รัฐบาลจะแจกเงินให้กับคนต่าง ๆ เป็นจำนวนเงินที่อาจจะแตกต่างกัน และแจกให้ ณ เวลาที่อาจจะแตกต่างกัน ด้วยจำนวนครั้งที่แตกต่างกันก็ได้ เงินที่ได้รับแจกเหล่านี้ จะมีระยะเวลาในการใช้งานอยู่ การใช้เงินดังกล่าวก็จะดำเนินการผ่าน app ด้วยเช่นกัน และเงินจำนวนดังกล่าวจะไม่สามารถใช้งานได้เมื่อพ้นระยะเวลาใช้งานไปแล้ว

จงเขียน class DigitalWallet เพื่อเป็นระบบกลางสำหรับให้ประชาชนและรัฐบาลได้ใช้งานในการดำเนินการนโยบายนี้ โดย DigitalWallet นี้จะต้องมีฟังก์ชันต่าง ๆ ให้เรียกใช้งาน การเรียกใช้งานฟังก์ชันการเงินต่าง ๆ นั้นจะมีการระบุเวลาที่เรียกใช้งานมาด้วย (ในตัวแปรชื่อ time) เพื่อความสะดวกเวลาในระบบนี้จะมีหน่วยเป็นวินาที โดยนับวินาทีที่ 0 คือเวลาแรกสุดที่ระบบนี้ใช้งาน

กำหนดให้คนแต่ละคนในประเทศนี้สามารถระบุได้ด้วย “รหัสประชาชน” ซึ่งเป็นข้อมูลประเภท string ความยาวไม่เกิน 10 ตัวอักษร

คลาสนี้จะต้องมีฟังก์ชันต่าง ๆ ต่อไปนี้สำหรับการใช้งาน

- void add_money(size_t time, string person_id, int amount, size_t duration) ณ เวลา time มีการเพิ่มเงินจำนวน amount ให้กับ wallet ของคนที่มี id คือ person_id โดยเงินดังกล่าวจะสามารถใช้ได้จนถึงเวลา time + duration เท่านั้น
- bool use_money(size_t time, string person_id, int amount) ณ เวลา time คนที่มี id คือ person_id ต้องการใช้จ่ายเงินจำนวน amount จาก wallet โดยเงินที่ใช้นี้จะถูกหักออกจากเงินที่อยู่ใน wallet ของคนดังกล่าว
 - ฟังก์ชันนี้จะคืนค่า true หากคนดังกล่าวมีเงินใน wallet ที่ยังใช้ได้เพียงพอต่อการใช้งาน และเงินใน wallet จะถูกหักออกไปตามจำนวน amount แต่ถ้าหากจำนวนเงินที่ยังใช้ได้ไม่พอเงินจะไม่ถูกหัก และฟังก์ชันนี้จะคืนค่า false
 - การหักเงินนั้น จะหักจากเงินที่พ้นระยะเวลาการใช้งานเร็วกว่าก่อนเสมอ (เช่น หากเราได้เงินมาสองก้อน คือ 100 กับ 200 โดยที่ก้อน 100 นั้นพ้นระยะเวลาการใช้งานก่อนก้อน 200 แล้วเราจะใช้เงิน 50 บาท เราจะหักเงินจากก้อน 100 ก่อน)
- int current_money(size_t time, string person_id) คืนจำนวนเงินที่ยังใช้ได้ของคนที่มี id เป็น person_id (ไม่รวมเงินที่พ้นระยะเวลาใช้งานไปแล้ว และไม่รวมเงินที่ถูกใช้ไปแล้ว)
- void status(size_t time, long long &total_give, long long &total_spent, long long &total_expired) เป็นฟังก์ชันที่คำนวณสถานะจำนวนเงินรวมของทั้งระบบ โดยเมื่อเรียกแล้วฟังก์ชันนี้จะต้องตั้งค่าให้กับตัวแปรต่าง ๆ ดังนี้
 - total_give คือจำนวนเงินรวมทั้งหมดที่รัฐบาลได้แจกมา
 - total_spent คือจำนวนเงินรวมทั้งหมดที่ถูกใช้ผ่านฟังก์ชัน use_money ที่คืนค่าเป็น true
 - total_expired คือจำนวนเงินรวมทั้งหมดที่หมดอายุไปก่อนโดยไม่ได้ใช้งาน

รับประกันว่าในการใช้งานคลาสนี้ สถานการณ์ต่าง ๆ ต่อไปนี้จะเป็นจริงเสมอ

- การเรียกฟังก์ชันใด ๆ ที่มีค่า time นั้น รับประกันว่าการเรียกแต่ละครั้งค่า time จะไม่น้อยลงเลย
- สำหรับการเรียก add_money แต่ละครั้งนั้น รับประกันว่าเงินที่ add_money มาในการเรียกครั้งล่าสุดจะพ้นระยะเวลาการใช้งาน “ไม่เร็วกว่า” เงินจากการ add_money ที่ถูกเรียกมาก่อนหน้า

ข้อบังคับ

- โจทย์ข้อนี้จะมีไฟล์ตั้งต้นมาให้ ประกอบด้วยไฟล์ main.cpp และ student.h อยู่ให้นิสิตเขียน code เพิ่มเติมลงในไฟล์ student.h เท่านั้น และการส่งไฟล์เข้าสู่ระบบ grader ให้ส่งเฉพาะไฟล์ student.h เท่านั้น
 - ในไฟล์ student.h จะมีโครงของคลาส DigitalWallet อยู่ นิสิตสามารถแก้ไขเพิ่มเติมคลาสนี้เช่นใดก็ได้ แต่คลาสดังกล่าวยังคงต้องมีฟังก์ชันตามข้างต้นอยู่ และนิสิตสามารถแก้ไขฟังก์ชันดังกล่าวได้เช่นกัน
 - ไฟล์ student.h จะต้องไม่ทำการอ่านเขียนข้อมูลใด ๆ ไปยังหน้าจอหรือคีย์บอร์ดหรือไฟล์ใด ๆ

- หากใช้ VS Code ให้ทำการ compile ที่ไฟล์ main.cpp
**** main ที่ใช้จริงใน grader นั้นจะแตกต่างจาก main ที่ได้รับในไฟล์ตั้งต้นแต่จะทำการทดสอบในลักษณะเดียวกัน ****

คำอธิบายฟังก์ชัน main

main() จะสร้าง Object ของคลาส DigitalWallet ขึ้นมา และอ่านคำสั่งการใช้งานจากคีย์บอร์ด และเรียกใช้งานฟังก์ชันตามคำสั่งการใช้งานดังกล่าว โดยมีรูปแบบต่อไปนี้

- แต่ละบรรทัดเริ่มต้นด้วยอักขระ 1 ตัว ที่ระบุถึงฟังก์ชันที่ต้องการเรียก ตามด้วยค่าต่าง ๆ ที่ขึ้นอยู่กับการใช้งานฟังก์ชันนั้น ๆ โดยมีรายละเอียดดังนี้
 - ตัวอักษร a หมายถึงต้องการใช้ฟังก์ชัน add_money และบรรทัดนั้นจะตามด้วยข้อมูลสี่ตัวเป็นประเภท size_t string int size_t ตามลำดับ ซึ่งค่าดังกล่าวจะถูกใช้เป็น argument ของฟังก์ชัน add_money
 - ตัวอักษร u หมายถึงต้องการใช้ฟังก์ชัน use_money และบรรทัดนั้นจะตามด้วยข้อมูลสามตัวเป็นประเภท size_t string int ตามลำดับ ซึ่งค่าดังกล่าวจะถูกใช้เป็น argument ของฟังก์ชัน use_money เมื่อเรียกแล้ว จะแสดงผลที่ได้รับคืนมาจากฟังก์ชัน use_money ออกทางหน้าจอ
 - ตัวอักษร c หมายถึงต้องการใช้ฟังก์ชัน current_money และบรรทัดนั้นจะตามด้วยข้อมูลสองตัวเป็นประเภท size_t string ตามลำดับ ซึ่งค่าดังกล่าวจะถูกใช้เป็น argument ของฟังก์ชัน current_money เมื่อเรียกแล้ว จะแสดงผลที่ได้รับคืนมาจากฟังก์ชัน current_money ออกทางหน้าจอ
 - ตัวอักษร s หมายถึงต้องการใช้ฟังก์ชัน status และบรรทัดนั้นจะตามด้วยข้อมูลหนึ่งตัวเป็นประเภท size_t ซึ่งค่าดังกล่าวจะถูกใช้เป็น argument แรกของฟังก์ชัน status หลังจากนั้นจะทำการพิมพ์ค่าของตัวแปร total_give, total_spent, total_expired ออกทางหน้าจอ
 - ตัวอักษร q หมายถึงจบการทำงาน

สมมติฐานเกี่ยวกับการใช้งาน

ฟังก์ชันทั้ง 4 นั้นสามารถถูกเรียกใช้งานเป็นจำนวนแตกต่างกันอย่างไรก็ได้ ไม่จำเป็นต้องมีการเรียกฟังก์ชันหนึ่งมากกว่าอีกฟังก์ชันหนึ่ง และความซับซ้อนเชิงเวลาที่คาดหวังไว้ของคลาสนี้คือ โดยเฉลี่ยแล้ว หากมีการเรียกฟังก์ชันทั้งหมด k ครั้ง เวลาที่ใช้ควรจะเป็น $O(k)$

ชุดข้อมูลทดสอบ

สำหรับชุดทดสอบใด ๆ จะมี จำนวนคำสั่ง $\leq 1,000,000$ คำสั่ง และมีจำนวนคนในระบบ $\leq 50,000$ คน และค่าเวลาที่เรียกใช้งานฟังก์ชันจะไม่เกิน 1,000,000,000

- 10% จำนวนคำสั่ง ≤ 100 มีคนในระบบเพียงคนเดียว และมีแต่คำสั่ง add_money กับ current_money
- 10% จำนวนคำสั่ง ≤ 100 และมีแต่คำสั่ง add_money กับ current_money
- 15% มีคนในระบบเพียงคนเดียว และมีแต่คำสั่ง add_money กับ current_money
- 20% มีคนในระบบเพียงคนเดียว
- 45% ไม่มีเงื่อนไขอื่น ๆ

คำแนะนำในการทำโจทย์ในกระดาษ

- ให้นิสิตเขียนคำตอบลงในกระดาษเป็นสามข้อย่อยดังนี้
 - อธิบายตัวแปร data member ทั้งหมดที่คลาสนี้ใช้ โดยให้ระบุว่าตัวแปรแต่ละตัว (รวมถึงองค์ประกอบภายใน) ใช้เก็บค่าอะไร
 - อธิบาย “แนวคิด” ในการใช้งาน data member ดังกล่าวเพื่อทำงานตามฟังก์ชันต่าง ๆ ของคลาสนี้
 - วิเคราะห์ Big O ของแต่ละฟังก์ชันด้วย

(ตัวอย่างอยู่ในหน้าถัดไป)

ตัวอย่าง

ข้อมูลนำเข้า	คำอธิบาย
a 1 somchai 20 100	somchai ได้เงิน 20 บาท โดยต้องใช้ภายในเวลา 101
a 1 nattee 20 100	nattee ได้เงิน 20 บาท โดยต้องใช้ภายในเวลา 101
a 1 vishnu 30 100	vishnu ได้เงิน 30 บาท โดยต้องใช้ภายในเวลา 101
a 1 somchai 50 100	somchai ได้เงิน 50 บาท โดยต้องใช้ภายในเวลา 101 (รวมเป็นมี 70 บาท)
a 5 somchai 60 100	somchai ได้เงินอีก 60 บาท โดยต้องใช้ภายในเวลา 105 (รวมเป็นมี 130 บาท)
u 10 nattee 10	nattee ใช้เงิน 10 บาท (เหลือเงิน 10 บาท) คืนค่า true
u 101 nattee 5	nattee ใช้เงินอีก 5 บาท (เหลือเงิน 5 บาท) คืนค่า true
c 101 nattee	คืนค่า 5
u 103 nattee 2	nattee ใช้เงินอีก 2 บาท แต่เนื่องจากพ้นระยะเวลาที่กำหนดไว้ จึงใช้ไม่ได้ คืนค่า false
u 105 nuttapong 40	nuttapong ใช้เงิน 40 บาท แต่ไม่เคยได้เงินมาก่อน จึงคืนค่า false
s 105	total_give = 180, total_spent = 15, total_expired = 105
q	