

รายการอ้างอิง



ภาษาไทย

ชัชวาล ประภาวิทย์. ระบบบริหารการใช้โลหิตและส่วนประกอบของโลหิตของโรงพยาบาลรัฐและเอกชนในกรุงเทพมหานคร. วิทยานิพนธ์ปริญญาโทบริหารบัณฑิต ภาควิชาเวชศาสตร์ป้องกันและสังคม บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

ชัยเวช นุชประยูร. การจัดการโลหิตเพื่อใช้ภายในประเทศ. กรุงเทพมหานคร: ศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย, 2536. (อัดสำเนา)

ชัยเวช นุชประยูร. การจัดการโลหิตเพื่อใช้ภายในประเทศ. กรุงเทพมหานคร: ศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย, 2537. (อัดสำเนา)

ทัศนัยยานี จันทนยิ่งยง. เลือดและธนาคารเลือดในประเทศไทย. สารานุกรมไทยสำหรับเยาวชน 8 (2530): 207-225.

พิมล เชี่ยวศิลป์. การให้เลือด. กรุงเทพมหานคร: โรงพิมพ์พิมพ์มณเฑศ, 2526.

พิมล เชี่ยวศิลป์. ความจำเป็นในการพัฒนางานธนาคารเลือด. วารสารโลหิตวิทยาและเวชศาสตร์บริการโลหิต 2 (มกราคม-มีนาคม)2535): 73-77.

พิมล เชี่ยวศิลป์, สืบสันต์ มหาสันทนา และ รัชนี โอเจริญ. ถาม-ตอบปัญหาน่ารู้ในงานธนาคารโลหิต. คำบรรยายการประชุมทางวิชาการของศูนย์บริการโลหิตแห่งชาติ สภากาชาดไทย. หน้า 96-111. 25 ตุลาคม 2534 ณ ห้องประชุมศิริ สิริโยธิน ศูนย์บริการโลหิตแห่งชาติ.

มยุรี วดีนาคกร . วิสัยญ์วิทยา. พิมพ์ครั้งที่ 1. สงขลา: ภาควิชาวิสัยญ์วิทยา คณะแพทยศาสตร์ มหาวิทยาลัยสงขลานครินทร์, 2528.

มหาวิทยาลัยมหิดล. คณะแพทยศาสตร์ศิริราชพยาบาล. ภาควิชาอายุรศาสตร์. สาขาโลหิตวิทยา. คู่มือโลหิตวิทยา. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: โรงพิมพ์เรือนแก้วการพิมพ์, 2529.

โรงพยาบาลจุฬาลงกรณ์. ฝ่ายเวชระเบียนสถิติ. สถิติโรงพยาบาลจุฬาลงกรณ์ 2542. กรุงเทพมหานคร: โรงพยาบาลจุฬาลงกรณ์ ฝ่ายเวชระเบียนสถิติ, 2542.

ศรีสนิท ศรแดง. คู่มือวิชาธนาคารเลือด. กรุงเทพมหานคร: (ม.ป.ท.), 2522.

ศิริจันทร์ ทองประเสริฐ. การจำลองแบบปัญหา. กรุงเทพมหานคร: โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2542.

สาธารณสุข, กระทรวง. คู่มือการปฏิบัติงานธนาคารเลือด. กรุงเทพมหานคร: (ม.ป.ท.), 2537.

ภาษาอังกฤษ

- Cheng, G., et al. A novel system providing compatible blood to patients during surgery:"self-service" elective blood banking by nursing staff. Transfusion 4 (1996): 50-347.
- Forrester, J. W. Industrial Dynamics. Massachusetts: MIT Press,1961.
- Forrester, J. W. Information sources for modelling the national economy. Journal of the American Statistical Association 75 (1980): 555-567. Cited in Rasul, F. A. F. A system dynamics approach investigating managerial performance in the building construction industry : A case study of Pakistan. Master's Thesis, School of Civil Engineering, Asian Institute of Technology, 1998.
- Katsuhiko Ogata, and Cliffs E. System Dynamics. (n.p.): Prentice Hall,1992.
- Richardson,G.P., and Pugh III A.L. Introduction to system dynamics modelling with DYNAMO. Massachusetts: MIT Press, 1981.
- Rodrigues, A., and Williams, T. System dynamics in software project management: towards the development of a formal integrated framework. Management Science[Online]. 1996. Available from : <http://www.managementscience.org/research/ab9605.asp> [2001, July 16]
- Sahay, B. S.; Prem Vrat; and Jain, P. K. Long term fertiliser demand, production and imports in India-a system dynamics approach. System Dynamics : An Int. J. of Policy Modelling 8 (1996): 19-45.
- Shoukath Ali, K., and Ramaswamy, N. Setting inventory levels for a centrally located blood bank of a metropolis - a simulation approach. System Dynamics'91: Proceedings of the 1991 International System Dynamics Conference , pp. 1-8. (n.p.), 1991.
- Wantana Nanthana. An economic analysis of blood procurement system in Bangkok. Master's Thesis, Faculty of Economics, Graduate School, Thammasat University.

ภาคผนวก

ซอร์สโค้ดโปรแกรม : แบบจำลองการใช้โลหิตของโรงพยาบาลจุฬาลงกรณ์

actual fraction = อัตราส่วนการได้รับโลหิตตามจำนวนที่ขอเบิก เป็นค่าคงที่ในแต่ละช่วงเวลาที่ได้จากการเก็บข้อมูลในช่วงเดือนสิงหาคมถึงธันวาคม

actual use = จำนวนโลหิตที่มีการนำไปใช้จริงในแต่ละวัน ได้จากการเก็บข้อมูลจริง โดยเป็นข้อมูลนำเข้าจากไฟล์ excel

addition order = จำนวนโลหิตที่ขอเบิกจากศูนย์ฯเพิ่มจากการขอเบิกปกติ เนื่องจากมีโลหิตในคลังไม่เพียงพอ

age of blood = อายุของโลหิต เกิดจากการใช้ฟังก์ชัน random เพื่อสุ่มอายุโลหิต อายุของโลหิตแต่ละหมู่จะมีสมการคล้ายกัน แตกต่างกันที่ค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของการกระจายแบบปกติซึ่งได้แสดงไว้ในส่วนการวิเคราะห์ข้อมูลแล้ว

blood = จำนวนโลหิต (อายุมากกว่า 7 วัน) ที่มีการเบิกมาที่ธนาคารโลหิต

blood price per unit = ต้นทุนของโลหิตครบส่วน 1 หน่วย (330 บาท/หน่วย)

c_i = อัตราการนำโลหิตอายุ i วัน มาทำ crossmatch (i มีค่าตั้งแต่ 1-20 วัน)

CB_i = คลังโลหิตอายุ i วันที่ทำ crossmatch เสร็จแล้ว (ถูกจองใช้แล้ว)

$cb(i) \text{ to } (i-j)$ = จำนวนโลหิตอายุ i วันที่ทำ crossmatch แล้ว และจะถูกคืนในอีก j วัน โดยจะกลับเข้าสู่คลัง WB_{i-j} โดย i มีค่าตั้งแต่ 1-20 และ j มีค่าตั้งแต่ 0-3 ยกตัวอย่างเช่น $cb_{20} \text{ to } 18$ หมายถึง จำนวนโลหิตอายุ 20 วันที่ทำ crossmatch แล้ว และจะถูกคืนในอีก 2 วัน โดยจะเข้าสู่คลัง WB_{18}

$cb(i) \text{ to } 0 \text{ d}(j)$ = จำนวนโลหิตอายุ i วันที่ทำ crossmatch แล้ว และจะถูกคืนเข้าสู่คลังโลหิตหมดอายุในอีก j วัน โดย i และ j มีค่าตั้งแต่ 1-3 ยกตัวอย่างเช่น $cb_{2} \text{ to } 0 \text{ d}3$ หมายถึง จำนวนโลหิตอายุ 2 วันที่ทำ crossmatch แล้ว และจะถูกคืนในอีก 3 วัน โดยจะเข้าสู่คลัง Expire blood

crossmatch and use ratio = อัตราส่วนของจำนวนโลหิตที่เตรียมไว้กับจำนวนโลหิตที่ถูกนำไปใช้จริง

crossmatch cost per unit = ต้นทุนต่อหน่วยในการ crossmatch เท่ากับ 30 บาทต่อหน่วย

crossmatch rate = อัตราการทำ crossmatch รวมของโลหิตทุกอายุ

d_1 = ผลต่างระหว่างจำนวนโลหิตอายุมากกว่า 7 วันที่มีการเบิกเข้ามากับจำนวนโลหิตในคลัง

$d(i)$ = ผลต่างระหว่าง $d(i-1)$ กับจำนวนโลหิตในคลังโลหิตอายุ i วัน ค่า d เหล่านี้ใช้ในการคิดอัตราการทำ crossmatch ของโลหิตในคลังโลหิตอายุต่างๆกัน โดยคิดจากการเอาค่าความต้องการทั้งหมดหักลบจำนวนโลหิตที่มีอยู่ในคลังซึ่งมีอายุต่างกัน และจะหักจากโลหิตที่มีอายุเหลืออยู่น้อยกว่าก่อน ซึ่งจะแบ่งเป็น 2 ส่วนคือโลหิตสดกับโลหิตธรรมดา สำหรับโลหิตสดจะจ่ายโลหิตที่มีอายุ 15 วันออกไปก่อน แล้วไล่ไปจนถึงอายุ 20 วัน ส่วนโลหิตธรรมดา จะจ่ายโลหิตอายุ 1 วันออกไปก่อนแล้วไล่ไปจนถึงอายุ 14 วัน (ค่า i มีค่าตั้งแต่ 2-20)

desire inventory = จำนวนโลหิตที่ควรมีอยู่ในคลัง ธนาคารโลหิตได้กำหนดค่านี้สำหรับโลหิตแต่ละหมู่ไว้ดังนี้ หมู่ O,B อย่างละ 30 หน่วย หมู่ A 20 หน่วย และหมู่ AB 10 หน่วย

diff = ค่าที่น้อยที่สุดระหว่างจำนวนโลหิตทั้งหมดที่ยังไม่ได้ crossmatch กับจำนวนโลหิตที่แพทย์ขอเบิกไว้แต่ยังไม่ได้ crossmatch

expense of crossmatch but not used = ค่าใช้จ่ายในการทำ crossmatch แต่ไม่ได้นำโลหิตไปใช้จริง

Expire blood = จำนวนเลือดที่หมดอายุ

fill order = อัตราการทำ crossmatch ในแต่ละวัน

fraction for fresh blood = สัดส่วนของโลหิตสดที่มีความต้องการเข้ามายังธนาคารโลหิต

fresh blood = จำนวนโลหิตสด (อายุไม่เกิน 7 วัน) ที่มีการเบิกมาที่ธนาคารโลหิต

g_i = โลหิตอายุ i วัน ที่ได้รับจากศูนย์บริการโลหิต จำนวนนี้จะมีค่าเมื่อ age of blood เท่ากับ i และจะไหลเข้าไปในคลัง WB i โดยค่า i มีค่าตั้งแต่ 16-20 ยกตัวอย่างเช่น g_{17} หมายถึงโลหิตอายุ 17 วันที่ได้รับจากศูนย์บริการโลหิต

get from NBC = จำนวนโลหิตที่ได้รับจากศูนย์บริการโลหิตแห่งชาติ

group = หมู่โลหิต ได้แก่ หมู่โลหิต O,B,A และ AB โดยตัวแปรนี้เป็นตัวที่ใช้ระบุ Subscript สำหรับการเขียนโปรแกรม

Initial order to NBC = จำนวนโลหิตที่ธนาคารโลหิตขอเบิกจากศูนย์บริการโลหิต

int $wb(i)$ = จำนวนโลหิตที่มีอยู่ใน WB i ณ เวลาเริ่มต้น โดย i มีค่า 5-7,9-11 และ 15-20

money lose for expire blood = ค่าใช้จ่ายที่ต้องสูญเสียไปเนื่องจากโลหิตหมดอายุ

order = จำนวนโลหิตที่ขอเบิกจากศูนย์บริการโลหิต

order from doctor = จำนวนโลหิตที่แพทย์ขอเบิกมาที่ธนาคารโลหิต เป็นข้อมูลจริงที่เก็บรวบรวมมา

out from CB i = จำนวนโลหิตอายุ i วันที่ถูกจองใช้ไปแล้วแต่ไม่ได้ถูกนำไปใช้จริง ยกตัวอย่างเช่น out from CB20 หมายถึง จำนวนโลหิตอายุ 20 วันที่ถูกจองใช้ไปแล้วแต่ไม่ได้ถูกนำไปใช้จริง

percent of expire blood = ร้อยละของโลหิตที่หมดอายุต่อโลหิตที่จัดหามาได้

Qty on order = จำนวนโลหิตที่ขอเบิกจากศูนย์ฯไปแล้วแต่ยังไม่ได้รับ

rand = เลขสุ่มแบบเอกภาพที่ใช้เป็นตัวกำหนดการวนซ้ำเพื่อคำนวณหาอัตราการทำโลหิตอายุต่างๆออกไปใช้

rand $rt(i)$ = ตัวแปรสุ่มสำหรับจำลองเหตุการณ์ว่าโลหิตอายุ $20-i$ วัน (ในคลัง CB) จะถูกนำมาคืนในอีกกี่วัน ถ้า $rand rt(i)$ เท่ากับ 0 แสดงว่าถูกนำมาคืนภายในวันที่ทำ crossmatch ถ้าเท่ากับ 1 แสดงว่าถูกนำมาคืนหลังจากทำ crossmatch แล้ว 1 วัน หากเท่ากับ 2 แสดงว่าอีก 2 วันจึงได้โลหิตกลับคืนมาที่คลัง WB โดยค่าสถิติ

ของตัวแปรสุ่มนี้เป็นไปตามค่าสถิติที่ได้วิเคราะห์ไว้แล้วเบื้องต้น และค่า i มีค่าตั้งแต่ 0-19 rand rt_1 จะมีลักษณะเช่นเดียวกับ rand rt_0 ต่างกันแค่ค่า seed เพื่อให้เกิดการสุ่มค่าที่แตกต่างกันไป

return rate = อัตราการคืนโลหิตรวมทุกคลังอายุของโลหิต (หน่วย/วัน)

rt expire = โลหิตทั้งหมดที่คืนกลับเข้ายังคลัง Expire blood

rt in(i) = โลหิตในคลัง CB i ที่คืนกลับมายังคลัง WB i ภายใน i วัน โดย i มีค่าตั้งแต่ 1-20

rt time = เวลา delay สำหรับโลหิตที่คืนภายใน 1 วัน ในที่นี้กำหนดให้เท่ากับครึ่งวัน

rt (i) to ($i-j$) = โลหิตในคลัง CB i ที่คืนกลับมายังคลัง WB $i-j$ (คืนหลังจาก crossmatch j วัน) โดย i มีค่าตั้งแต่ 2-20, j มีค่า 1-3 และค่า $i-j$ ต้องมีค่ามากกว่า 0 ยกตัวอย่างเช่น $rt_{11}to_9$ หมายถึงโลหิตในคลัง CB11 ที่คืนกลับมายังคลัง WB9 (คืนหลังจาก crossmatch 2 วัน)

rt (i) d (j) = โลหิตที่คืนจากคลัง CB i เข้าสู่คลัง Expire blood โดยจะถูกคืนกลับมาหลัง crossmatch ภายใน j วัน ตัวอย่างเช่น $rt_{1d}2$ หมายถึงโลหิตที่คืนจากคลัง CB1 เข้าสู่คลัง Expire blood โดยจะถูกคืนกลับมาหลัง crossmatch ไปแล้ว 2 วัน (i และ j มีค่าตั้งแต่ 1-3)

rt_{1to exp} = โลหิตจากคลัง CB1 ทั้งหมดที่คืนกลับเข้ามายังคลัง Expire blood

rt_{2to exp} = โลหิตจากคลัง CB2 ทั้งหมดที่คืนกลับเข้ามายังคลัง Expire blood

rt_{3d3} = โลหิตที่คืนจากคลัง CB3 เข้าสู่คลัง Expire blood โดยจะถูกคืนกลับมาหลัง crossmatch ไปแล้ว 3 วัน

rt (i) = โลหิตทั้งหมดที่จะคืนกลับเข้าไปยังคลัง WB i

sub i = เท่ากับโลหิตช่วงอายุต่างๆที่ทำ crossmatch แล้ว โดยจะเป็นโลหิตที่มีอายุเท่าไรขึ้นกับค่าสุ่มที่ได้ในแต่ละช่วงเวลา เช่นถ้าค่าสุ่มเท่ากับ i sub i จะเท่ากับ CB i

to i = อัตราการไหลของโลหิตอายุ $i+1$ วันเข้าสู่คลังโลหิตอายุ i วัน โดย i มีค่าตั้งแต่ 1-19

Total blood get from NBC = จำนวนโลหิตทั้งหมดที่ได้รับจากศูนย์บริการโลหิต

Total cross blood = จำนวนโลหิตทั้งหมดที่ผ่านการทำ crossmatch

Total order = จำนวนโลหิตทั้งหมดที่ขอเบิกจากศูนย์บริการโลหิต

Total return = จำนวนโลหิตทั้งหมดคืนกลับมายัง คลัง WB

tune decimal = ผลต่างเนื่องจากการปรับค่าศนิยม ซึ่งเป็นผลจากส่วนอื่นของแบบจำลอง

Unfilled order = จำนวนโลหิตที่แพทย์ขอเบิกมา แต่ยังไม่ได้ทำ crossmatch

use rate = อัตราการนำโลหิตไปใช้รวม

use i = อัตราการนำจำนวนโลหิตอายุ i วัน ที่ทำ crossmatch ไว้แล้วไปใช้ โดย i มีค่าตั้งแต่ 1-20

useP_i = จำนวนโลหิตที่จะถูกนำไปใช้เป็นอันดับที่ i โดย i มีค่าตั้งแต่ 1-20 ตัวอย่างเช่น useP₂ หมายถึงจำนวนโลหิตที่จะถูกนำไปใช้เป็นอันดับที่สอง

waiting time = เวลาที่ต้องรอรับโลหิตจากศูนย์บริการโลหิต ปกติประมาณ 1 ชั่วโมง

WB_i = คลังโลหิตอายุ i วันที่ยังไม่ได้ทำ crossmatch ตัวอย่างเช่น WB₂₀ หมายถึง คลังโลหิตอายุ 20 วันที่ยังไม่ได้ทำ crossmatch ซึ่งเป็นผลรวมของจำนวนโลหิตอายุ 20 วันที่ได้รับจากศูนย์บริการโลหิต กับจำนวนโลหิตอายุ 20 วันที่เหลือกลับคืนมาเมื่อไม่ได้รับถ่ายโลหิตให้กับผู้ป่วย หักลบกับโลหิตที่ถูกนำไป crossmatch และโลหิตที่อายุลดลงเหลือ 19 วัน

$x(i)$ = ผลต่างของจำนวนโลหิตที่ใช้จริงกับ sub(i) ค่า x มีหลักการคิดเช่นเดียวกับค่า d ในแบบจำลองการตัดสินใจเลือกโลหิตนำมาทำ crossmatch

actual fraction[O]:=GET XLS DATA('blood.xls' , 'actual receive fraction' , '1' , 'B5')

actual fraction[B]:=GET XLS DATA('blood.xls' , 'actual receive fraction' , '1' , 'B3')

actual fraction[A]:=GET XLS DATA('blood.xls' , 'actual receive fraction' , '1' , 'B2')

actual fraction[AB]:=GET XLS DATA('blood.xls' , 'actual receive fraction' , '1' , 'B4') Units: 1/Day

actual use[O]:= GET XLS DATA('blood.xls' , 'actual use' , '1' , 'B5')

actual use[B]:= GET XLS DATA('blood.xls' , 'actual use' , '1' , 'B3')

actual use[A]:= GET XLS DATA('blood.xls' , 'actual use' , '1' , 'B2')

actual use[AB]:= GET XLS DATA('blood.xls' , 'actual use' , '1' , 'B4') Units: unit/Day

addition order[group]=IF THEN ELSE(d14[group]>0, d14[group], 0)+IF THEN ELSE(d20[group]>0, d20[group], 0)+IF THEN ELSE(tune decimal[group] >=1, INTEGER(tune decimal[group]))
Units: unit

age of blood[O]= IF THEN ELSE(Time<=31,INTEGER(RANDOM NORMAL(16, 21 , 18.439, 0.8719 , 0)), IF THEN ELSE(Time<=61, INTEGER(RANDOM NORMAL(16, 21 , 18.439, 0.8719 , 1)), IF THEN ELSE(Time<=92,INTEGER(RANDOM NORMAL(16, 21 , 18.439, 0.8719 , 2)), IF THEN ELSE(Time<=122,INTEGER(RANDOM NORMAL(16, 21 , 18.439, 0.8719 , 3)), INTEGER(RANDOM NORMAL(16, 21 , 18.439, 0.8719 , 4))))))

age of blood[B]= IF THEN ELSE(Time<=31,INTEGER(RANDOM NORMAL(16, 21 , 18.423, 0.885 , 0)), IF THEN ELSE(Time<=61, INTEGER(RANDOM NORMAL(16, 21 , 18.423, 0.885 , 1)), IF THEN ELSE(Time<=92,INTEGER(RANDOM NORMAL(16, 21 , 18.423, 0.885 , 2)), IF THEN ELSE(Time<=122,INTEGER(RANDOM NORMAL(16, 21 , 18.423, 0.885 , 3)), INTEGER(RANDOM NORMAL(16, 21 , 18.423, 0.885 , 4))))))

age of blood[A]= IF THEN ELSE(Time<=31,INTEGER(RANDOM NORMAL(17, 21 , 18.654, 0.7709 , 0)), IF THEN ELSE(Time<=61, INTEGER(RANDOM NORMAL(17, 21 , 18.654, 0.7709 , 1)), IF THEN ELSE(Time<=92,INTEGER(RANDOM NORMAL(17, 21 , 18.654, 0.7709 , 2)), IF THEN

ELSE(Time<=122,INTEGER(RANDOM NORMAL(17, 21 , 18.654, 0.7709 , 3)) , INTEGER(RANDOM NORMAL(17, 21 , 18.654, 0.7709 , 4))))

age of blood[AB]=IF THEN ELSE(Time<=31,INTEGER(RANDOM NORMAL(17, 20 , 18.347, 0.6902 , 0)) , IF THEN ELSE(Time<=61 , INTEGER(RANDOM NORMAL(17, 20 , 18.347, 0.6902 , 1)) , IF THEN ELSE(Time<=92,INTEGER(RANDOM NORMAL(17, 20 , 18.347, 0.6902 , 2)) , IF THEN ELSE(Time<=122,INTEGER(RANDOM NORMAL(17, 20 , 18.347, 0.6902 , 3)) , INTEGER(RANDOM NORMAL(17, 20 , 18.347, 0.6902 , 4))))) Units: Dmnl

blood[group]=INTEGER(diff[group]*(1-fraction for fresh blood[group]))+IF THEN ELSE(MODULO (diff[group]*(1-fraction for fresh blood[group]),1)<>0 , 1 , 0) Units: unit

blood price per unit= 330 Units: bath/unit

c1[group]=IF THEN ELSE(blood[group]=0, 0 ,IF THEN ELSE(WB1[group]<=0, 0 , IF THEN ELSE (d1[group]>0, INTEGER(WB1[group]) ,INTEGER(blood[group])))) Units: unit/Day

c10[group]=IF THEN ELSE(WB10[group]<=0 :OR: d9[group]<=0, 0,IF THEN ELSE(d10[group]>0 , INTEGER(WB10[group]) ,INTEGER(d9[group]))) Units: unit/Day

c11[group]=IF THEN ELSE(WB11[group]<=0 :OR: d10[group]<=0 , 0 , IF THEN ELSE(d11[group]>0, INTEGER(WB11[group]), INTEGER(d10[group]))) Units: unit/Day

c12[group]= IF THEN ELSE(WB12[group]<=0 :OR:d11[group]<=0 , 0 , IF THEN ELSE(d12[group]>0, INTEGER(WB12[group]) ,INTEGER(d11[group]))) Units: unit/Day

c13[group]=IF THEN ELSE(WB13[group]<=0 :OR: d12[group]<=0 , 0 , IF THEN ELSE(d13[group]>0, INTEGER(WB13[group]) , INTEGER(d12[group]))) Units: unit/Day

c14[group]=IF THEN ELSE(WB14[group]<=0 :OR: d13[group]<=0 , 0 , IF THEN ELSE(d14[group]>0,INTEGER(WB14[group]) , INTEGER(d13[group]))) Units: unit/Day

c15[group]=IF THEN ELSE(WB15[group]<=0 :OR: fresh blood[group]<=0,0 , IF THEN ELSE(d15 [group]>0, INTEGER(WB15[group]) ,INTEGER(fresh blood[group]))) Units: unit/Day

c16[qroup]= IF THEN ELSE(WB16[group]<=0 :OR:d15[group]<=0 , 0,IF THEN ELSE(d16[group] >0, INTEGER(WB16[group]) ,INTEGER(d15[group]))) Units: unit/Day

c17[group]=IF THEN ELSE(WB17[group]<=0 :OR:d16[group]<=0 , 0 , IF THEN ELSE(d17[group]>0, INTEGER(WB17[group]) , INTEGER(d16[group]))) Units: unit/Day

c18[group]=IF THEN ELSE(WB18[group]<=0 :OR:d17[group]<=0 ,0 , IF THEN ELSE(d18[group] >0, INTEGER(WB18[group]) ,INTEGER(d17[group]))) Units: unit/Day

c19[group]=IF THEN ELSE(WB19[group]<=0 :OR:d18[group]<=0,0,IF THEN ELSE(d19[group]>0 , INTEGER(WB19[group]) , INTEGER(d18[group]))) Units: unit/Day

c2[group]=IF THEN ELSE(WB2[group]<=0 :OR: d1[group]<=0 , 0 , IF THEN ELSE(d2[group]>0 , INTEGER(WB2[group]) ,INTEGER(d1[group])) Units: unit/Day

c20[group]=IF THEN ELSE(WB20[group]<=0 :OR: d19[group]<=0,0 . IF THEN ELSE(d20[group]
 >0, INTEGER(WB20[group]) ,INTEGER(d19[group]))) Units: unit/Day
 c3[group]=IF THEN ELSE(WB3[group]<=0 :OR: d2[group]<=0, 0 ,IF THEN ELSE(d3[group]>0,
 INTEGER(WB3[group]) , INTEGER(d2[group]))) Units: unit/Day
 c4[group]=IF THEN ELSE(WB4[group]<=0 :OR: d3[group]<=0,0 ,IF THEN ELSE(d4[group]>0,
 INTEGER(WB4[group]) ,INTEGER(d3[group]))) Units: unit/Day
 c5[group]= IF THEN ELSE(WB5[group]<=0 :OR: d4[group]<=0,0 ,IF THEN ELSE(d5[group]>0,
 INTEGER(WB5[group]) , INTEGER(d4[group]))) Units: unit/Day
 c6[group]= IF THEN ELSE(WB6[group]<=0 :OR: d5[group]<=0,0 ,IF THEN ELSE(d6[group]>0,
 INTEGER(WB6[group]) , INTEGER(d5[group]))) Units: unit/Day
 c7[group]=IF THEN ELSE(WB7[group]<=0 :OR: d6[group]<=0, 0 ,IF THEN ELSE(d7[group]>0,
 INTEGER(WB7[group]) ,INTEGER(d6[group])) Units: unit/Day
 c8[group]=IF THEN ELSE(WB8[group]<=0 :OR: d7[group]<=0,0 ,IF THEN ELSE(d8[group]>
 0,INTEGER(WB8[group]) , INTEGER(d7[group]))) Units: unit/Day
 c9[group]=IF THEN ELSE(WB9[group]<=0 :OR: d8[group]<=0, 0 , IF THEN ELSE(d9[group]>0
 , INTEGER(WB9[group]) , INTEGER(d8[group]))) Units: unit/Day
 CB1[group]= INTEG (c1[group]-out from CB1[group]-use1[group],0) Units: unit
 CB10[group]= INTEG (c10[group]-out from CB10[group]-use10[group],0) Units: unit
 cb10to10[group]=INTEGER(IF THEN ELSE(rand rt10[group]=0,MAX(0,CB10[group]),0))
 cb10to7[group]= INTEGER(IF THEN ELSE(rand rt10[group]=3 ,MAX(0,CB10[group]) ,0))
 cb10to8[group]= INTEGER(IF THEN ELSE(rand rt10[group]=2 , MAX(0,CB10[group]) ,0))
 cb10to9[group]= INTEGER(IF THEN ELSE(rand rt10[group]=1 ,MAX(0,CB10[group]),0))
 Units: unit/Day
 CB11[group]= INTEG (c11[group]-out from CB11[group]-use11[group],0) Units: unit
 cb11to10[group]=INTEGER(IF THEN ELSE(rand rt9[group]=1 ,MAX(0,CB11[group]) ,0))
 cb11to11[group]=INTEGER(IF THEN ELSE(rand rt9[group]=0 ,MAX(0,CB11[group]) ,0))
 cb11to8[group]= INTEGER(IF THEN ELSE(rand rt9[group]=3 ,MAX(0,CB11[group]) ,0))
 cb11to9[group]= INTEGER(IF THEN ELSE(rand rt9[group]=2 ,MAX(0,CB11[group]) ,0))
 Units: unit/Day
 CB12[group]= INTEG (c12[group]-out from CB12[group]-use12[group],0) Units: unit
 cb12to10[group]=INTEGER(IF THEN ELSE(rand rt8[group]=2 ,MAX(0,CB12[group]) ,0))
 cb12to11[group]=INTEGER(IF THEN ELSE(rand rt8[group]=1 ,MAX(0,CB12[group]) ,0))
 cb12to12[group]=INTEGER(IF THEN ELSE(rand rt8[group]=0 ,MAX(0,CB12[group]) ,0))
 cb12to9[group]= INTEGER(IF THEN ELSE(rand rt8[group]=3 ,MAX(0,CB12[group]) ,0))
 Units: unit/Day
 CB13[group]= INTEG (c13[group]-out from CB13[group]-use13[group],0)Units: unit

cb13to10[group]=INTEGER(IF THEN ELSE(rand rt7[group]=3 ,MAX(0,CB13[group]) ,0))
 cb13to11[group]=INTEGER(IF THEN ELSE(rand rt7[group]=2 ,MAX(0,CB13[group]) ,0))
 cb13to12[group]=INTEGER(IF THEN ELSE(rand rt7[group]=1 ,MAX(0,CB13[group]) ,0))
 cb13to13[group]=INTEGER(IF THEN ELSE(rand rt7[group]=0 ,MAX(0,CB13[group]) ,0))

Units: unit/Day

CB14[group]= INTEG (c14[group]-out from CB14[group]-use14[group],0) Units: unit
 cb14to11[group]=IF THEN ELSE(rand rt6[group]=3 ,MAX(0,CB14[group]) ,0) Units: unit/Day
 cb14to12[group]=IF THEN ELSE(rand rt6[group]=2 ,MAX(0,CB14[group]) ,0) Units: unit/Day
 cb14to13[group]=IF THEN ELSE(rand rt6[group]=1 ,MAX(0,CB14[group]) ,0) Units: unit/Day
 cb14to14[group]=IF THEN ELSE(rand rt6[group]=0 ,MAX(0,CB14[group]) ,0) Units: unit/Day
 CB15[group]= INTEG (c15[group]-out from CB15[group]-use15[group],0) Units: unit
 cb15to12[group]=INTEGER(IF THEN ELSE(rand rt5[group]=3 ,MAX(0,CB15[group]) ,0))
 cb15to13[group]=INTEGER(IF THEN ELSE(rand rt5[group]=2 ,MAX(0,CB15[group]) ,0))
 cb15to14[group]=INTEGER(IF THEN ELSE(rand rt5[group]=1 ,MAX(0,CB15[group]) ,0))
 cb15to15[group]=INTEGER(IF THEN ELSE(rand rt5[group]=0 ,MAX(0,CB15[group]) ,0))

Units: unit/Day

CB16[group]= INTEG (c16[group]-out from CB16[group]-use16[group],0) Units: unit
 cb16to13[group]=INTEGER(IF THEN ELSE(rand rt4[group]=3 ,MAX(0,CB16[group]) ,0))
 cb16to14[group]=INTEGER(IF THEN ELSE(rand rt4[group]=2 ,MAX(0,CB16[group]) ,0))
 cb16to15[group]=INTEGER(IF THEN ELSE(rand rt4[group]=1 ,MAX(0,CB16[group]) ,0))
 cb16to16[group]=INTEGER(IF THEN ELSE(rand rt4[group]=0 ,MAX(0,CB16[group]) ,0))

Units: unit/Day

CB17[group]= INTEG (c17[group]-out from CB17[group]-use17[group],0) Units: unit
 cb17to14[group]=INTEGER(IF THEN ELSE(rand rt3[group]=3 ,MAX(0,CB17[group]) ,0))
 cb17to15[group]=INTEGER(IF THEN ELSE(rand rt3[group]=2 ,MAX(0,CB17[group]) ,0))
 cb17to16[group]=INTEGER(IF THEN ELSE(rand rt3[group]=1 ,MAX(0,CB17[group]) ,0))
 cb17to17[group]=INTEGER(IF THEN ELSE(rand rt3[group]=0 ,MAX(0,CB17[group]) ,0))

Units: unit/Day

CB18[group]= INTEG (c18[group]-out from CB18[group]-use18[group],0) Units: unit
 cb18to15[group]=INTEGER(IF THEN ELSE(rand rt2[group]=3 ,MAX(0,CB18[group]) ,0))
 cb18to16[group]=INTEGER(IF THEN ELSE(rand rt2[group]=2 ,MAX(0,CB18[group]) ,0))
 cb18to17[group]=INTEGER(IF THEN ELSE(rand rt2[group]=1 ,MAX(0,CB18[group]) ,0))
 cb18to18[group]=INTEGER(IF THEN ELSE(rand rt2[group]=0 ,MAX(0,CB18[group]) ,0))

Units: unit/Day

CB19[group]= INTEG (c19[group]-out from CB19[group]-use19[group],int cb19[group])

Units: unit

cb19to16[group]=INTEGER(IF THEN ELSE(rand rt1[group]=3,MAX(0,CB19[group]) ,0))
 cb19to17[group]=INTEGER(IF THEN ELSE(rand rt1[group]=2 ,MAX(0,CB19[group]) ,0))
 cb19to18[group]=INTEGER(IF THEN ELSE(rand rt1[group]=1,MAX(0,CB19[group]) ,0))
 cb19to19[group]=INTEGER(IF THEN ELSE(rand rt1[group]=0 ,MAX(0,CB19[group]),0))

Units: unit/Day

cb1to0d1[group]=INTEGER(IF THEN ELSE(rand rt19[group]=1 ,MAX(0,CB1[group]) ,0))
 cb1to0d2[group]=INTEGER(IF THEN ELSE(rand rt19[group]=2 ,MAX(0,CB1[group]) ,0))
 cb1to0d3[group]=INTEGER(IF THEN ELSE(rand rt19[group]=3 ,MAX(0,CB1[group]) ,0))
 cb1to1[group]= INTEGER(IF THEN ELSE(rand rt19[group]=0 ,MAX(0,CB1[group]) ,0))

Units: unit/Day

CB2[group]= INTEG (c2[group]-out from CB2[group]-use2[group],0) Units: unit
 CB20[group]= INTEG (c20[group]-out from CB20[group]-use20[group],0) Units: unit
 cb20to17[group]=INTEGER(IF THEN ELSE(rand rt0[group]=3 ,MAX(0,CB20[group]) ,0))
 cb20to18[group]=INTEGER(IF THEN ELSE(rand rt0[group]=2 , MAX(0,CB20[group]),0))
 cb20to19[group]=INTEGER(IF THEN ELSE(rand rt0[group]=1 ,MAX(0,CB20[group]) ,0))
 cb20to20[group]=INTEGER(IF THEN ELSE(rand rt0[group]=0 ,MAX(0,CB20[group]) ,0))
 cb2to0d2[group]=INTEGER(IF THEN ELSE(rand rt18[group]=2 ,MAX(0,CB2[group]) ,0))
 cb2to0d3[group]=INTEGER(IF THEN ELSE(rand rt18[group]=3 ,MAX(0,CB2[group]) ,0))
 cb2to1[group]= INTEGER(IF THEN ELSE(rand rt18[group]=1 ,MAX(0,CB2[group]) ,0))
 cb2to2[group]= INTEGER(IF THEN ELSE(rand rt18[group]=0 ,MAX(0,CB2[group]) ,0))

Units: unit/Day

CB3[group]= INTEG (c3[group]-out from CB3[group]-use3[group],0) Units: unit
 cb3to0[group]=INTEGER(IF THEN ELSE(rand rt17[group]=3 ,MAX(0,CB3[group]) ,0))
 cb3to1[group]=INTEGER(IF THEN ELSE(rand rt17[group]=2 ,MAX(0,CB3[group]) ,0))
 cb3to2[group]= INTEGER(IF THEN ELSE(rand rt17[group]=1 ,MAX(0,CB3[group]) ,0))
 cb3to3[group]= INTEGER(IF THEN ELSE(rand rt17[group]=0 ,MAX(0,CB3[group]) ,0))

Units: unit/Day

CB4[group]= INTEG (c4[group]-out from CB4[group]-use4[group],0) Units: unit
 cb4to1[group]= INTEGER(IF THEN ELSE(rand rt16[group]=3 ,MAX(0,CB4[group]) ,0))
 cb4to2[group]= INTEGER(IF THEN ELSE(rand rt16[group]=2 ,MAX(0,CB4[group]) ,0))
 cb4to3[group]= INTEGER(IF THEN ELSE(rand rt16[group]=1 ,MAX(0,CB4[group]) ,0))
 cb4to4[group]= INTEGER(IF THEN ELSE(rand rt16[group]=0 ,MAX(0,CB4[group]) ,0))

Units: unit/Day

CB5[group]= INTEG (c5[group]-out from CB5[group]-use5[group],0) Units: unit
 cb5to2[group]=IF THEN ELSE(rand rt15[group]=3 ,MAX(0,CB5[group]) ,0) Units: unit/Day
 cb5to3[group]=IF THEN ELSE(rand rt15[group]=2 ,MAX(0,CB5[group]) ,0) Units: unit/Day

cb5to4[group]=IF THEN ELSE(rand rt15[group]=1 ,MAX(0,CB5[group]) ,0) Units: unit/Day

cb5to5[group]=IF THEN ELSE(rand rt15[group]=0 ,MAX(0,CB5[group]) ,0) Units: unit/Day

CB6[group]= INTEG (c6[group]-out from CB6[group]-use6[group],0) Units: unit

cb6to3[group]=INTEG(IF THEN ELSE(rand rt14[group]=3 ,MAX(0,CB6[group]) ,0))

cb6to4[group]=INTEG(IF THEN ELSE(rand rt14[group]=2 ,MAX(0,CB6[group]) ,0))

cb6to5[group]= INTEG(IF THEN ELSE(rand rt14[group]=1 ,MAX(0,CB6[group]) ,0))

cb6to6[group]= INTEG(IF THEN ELSE(rand rt14[group]=0 ,MAX(0,CB6[group]) ,0))

Units: unit/Day

CB7[group]= INTEG (c7[group]-out from CB7[group]-use7[group],0) Units: unit

cb7to4[group]=INTEG(IF THEN ELSE(rand rt13[group]=3 ,MAX(0,CB7[group]) ,0))

cb7to5[group]=INTEG(IF THEN ELSE(rand rt13[group]=2 ,MAX(0,CB7[group]) ,0))

cb7to6[group]= INTEG(IF THEN ELSE(rand rt13[group]=1 ,MAX(0,CB7[group]) ,0))

cb7to7[group]= INTEG(IF THEN ELSE(rand rt13[group]=0 ,MAX(0,CB7[group]) ,0))

Units: unit/Day

CB8[group]= INTEG (c8[group]-out from CB8[group]-use8[group],0) Units: unit

cb8to5[group]=INTEG(IF THEN ELSE(rand rt12[group]=3 ,MAX(0,CB8[group]) ,0))

cb8to6[group]= INTEG(IF THEN ELSE(rand rt12[group]=2 ,MAX(0,CB8[group]) ,0))

cb8to7[group]= INTEG(IF THEN ELSE(rand rt12[group]=1 ,MAX(0,CB8[group]) ,0))

cb8to8[group]= INTEG(IF THEN ELSE(rand rt12[group]=0 ,MAX(0,CB8[group]) ,0))

Units: unit/Day

CB9[group]= INTEG (c9[group]-out from CB9[group]-use9[group],0) Units: unit

cb9to6[group]=INTEG(IF THEN ELSE(rand rt11[group]=3 ,MAX(0,CB9[group]) ,0))

cb9to7[group]=INTEG(IF THEN ELSE(rand rt11[group]=2 ,MAX(0,CB9[group]) ,0))

cb9to8[group]= INTEG(IF THEN ELSE(rand rt11[group]=1 ,MAX(0,CB9[group]) ,0))

cb9to9[group]= INTEG(IF THEN ELSE(rand rt11[group]=0 ,MAX(0,CB9[group]) ,0))

Units: unit/Day

cross blood all group=SUM(Total cross blood[group!]) Units: unit

crossmatch and use ratio[group]= IF THEN ELSE(Used blood[group]=0, 0 , Total cross blood[group]/Used blood[group]) Units: Dmnl

crossmatch and use ratio all group=IF THEN ELSE(used blood all group=0, 0 , cross blood all group/used blood all group) Units: Dmnl

crossmatch cost per unit= 30 Units: bath/unit

crossmatch rate[group]= c1[group]+c2[group]+c3[group]+c4[group]+c5[group]+c6[group]+c7 [group]+c8[group]+c9[group]+c10[group]+c11[group]+c12[group]+c13[group]+c14[group]+c15 [group]+c16[group]+c17[group]+c18[group]+c19[group]+c20[group] Units: unit/Day

$d1[\text{group}] = \text{INTEGER}(\text{blood}[\text{group}]) - \text{INTEGER}(\text{WB1}[\text{group}])$ Units: unit
 $d10[\text{group}] = \text{INTEGER}(d9[\text{group}]) - \text{INTEGER}(\text{WB10}[\text{group}])$ Units: unit
 $d11[\text{group}] = \text{INTEGER}(d10[\text{group}]) - \text{INTEGER}(\text{WB11}[\text{group}])$ Units: unit
 $d12[\text{group}] = \text{INTEGER}(d11[\text{group}]) - \text{INTEGER}(\text{WB12}[\text{group}])$ Units: unit
 $d13[\text{group}] = \text{INTEGER}(d12[\text{group}]) - \text{INTEGER}(\text{WB13}[\text{group}])$ Units: unit
 $d14[\text{group}] = \text{INTEGER}(d13[\text{group}]) - \text{INTEGER}(\text{WB14}[\text{group}])$ Units: unit
 $d15[\text{group}] = \text{INTEGER}(\text{fresh blood}[\text{group}]) - \text{INTEGER}(\text{WB15}[\text{group}])$ Units: unit
 $d16[\text{group}] = \text{INTEGER}(d15[\text{group}]) - \text{INTEGER}(\text{WB16}[\text{group}])$ Units: unit
 $d17[\text{group}] = \text{INTEGER}(d16[\text{group}]) - \text{INTEGER}(\text{WB17}[\text{group}])$ Units: unit
 $d18[\text{group}] = \text{INTEGER}(d17[\text{group}]) - \text{INTEGER}(\text{WB18}[\text{group}])$ Units: unit
 $d19[\text{group}] = \text{INTEGER}(d18[\text{group}]) - \text{INTEGER}(\text{WB19}[\text{group}])$ Units: unit
 $d2[\text{group}] = \text{INTEGER}(d1[\text{group}]) - \text{INTEGER}(\text{WB2}[\text{group}])$ Units: unit
 $d20[\text{group}] = \text{INTEGER}(d19[\text{group}]) - \text{INTEGER}(\text{WB20}[\text{group}])$ Units: unit
 $d3[\text{group}] = \text{INTEGER}(d2[\text{group}]) - \text{INTEGER}(\text{WB3}[\text{group}])$ Units: unit
 $d4[\text{group}] = \text{INTEGER}(d3[\text{group}]) - \text{INTEGER}(\text{WB4}[\text{group}])$ Units: unit
 $d5[\text{group}] = \text{INTEGER}(d4[\text{group}]) - \text{INTEGER}(\text{WB5}[\text{group}])$ Units: unit
 $d6[\text{group}] = \text{INTEGER}(d5[\text{group}]) - \text{INTEGER}(\text{WB6}[\text{group}])$ Units: unit
 $d7[\text{group}] = \text{INTEGER}(d6[\text{group}]) - \text{INTEGER}(\text{WB7}[\text{group}])$ Units: unit
 $d8[\text{group}] = \text{INTEGER}(d7[\text{group}]) - \text{INTEGER}(\text{WB8}[\text{group}])$ Units: unit
 $d9[\text{group}] = \text{INTEGER}(d8[\text{group}]) - \text{INTEGER}(\text{WB9}[\text{group}])$ Units: unit
 $\text{desire inventory}[\text{group}] = 30, 30, 20, 10$ Units: unit
 $\text{diff}[\text{group}] = \text{MIN}(\text{Total WB}[\text{group}], \text{Unfilled order}[\text{group}])$ Units: unit

$\text{expense of crossmatch but not use all group} = \text{MAX}(0, (\text{crossmatch and use ratio all group} - 1) * \text{used blood all group} * \text{crossmatch cost per unit})$ Units: bath

$\text{expense of crossmatch but not used}[\text{group}] =$

$\text{MAX}(0, (\text{crossmatch and use ratio}[\text{group}] - 1) * \text{Used blood}[\text{group}] * \text{crossmatch cost per unit})$
 Units: bath

$\text{expire}[\text{group}] = \text{INTEGER}(\text{IF THEN ELSE}(\text{WB1}[\text{group}] \leq 0, 0, \text{WB1}[\text{group}]))$ Units: unit/Day

$\text{Expire blood}[\text{group}] = \text{INTEG}(\text{in expire}[\text{group}], 0)$ Units: unit

$\text{expire blood all group} = \text{SUM}(\text{Expire blood}[\text{group}!])$ Units: unit

$\text{fill order}[\text{group}] = \text{IF THEN ELSE}(\text{crossmatch rate}[\text{group}] < > 0, \text{IF THEN ELSE}(\text{Unfilled order}[\text{group}] > = \text{crossmatch rate}[\text{group}], \text{crossmatch rate}[\text{group}], \text{Unfilled order}[\text{group}]), 0)$ Units: unit/Day

$\text{FINAL TIME} = 153$ Units: Day

$\text{fraction for fresh blood}[\text{group}] = 0.4$ Units: Dmnl

fresh blood[group]=INTEGER(diff[group]*fraction for fresh blood[group])+IF THEN ELSE(MODULO(diff[group]*fraction for fresh blood[group], 1)<>0, 1, 0) Units: unit

g16[group]=INTEGER(IF THEN ELSE(age of blood[group]=16,get from NBC[group],0))

g17[group]=INTEGER(IF THEN ELSE(age of blood[group]=17,get from NBC[group], 0))

g18[group]=INTEGER(IF THEN ELSE(age of blood[group]=18,get from NBC[group] , 0))

g19[group]=INTEGER(IF THEN ELSE(age of blood[group]=19, get from NBC[group] ,0))

g20[group]=INTEGER(IF THEN ELSE(age of blood[group]=20,get from NBC[group] , 0))

Units: unit/Day

get blood[group]=get from NBC[group] Units: unit/Day

get blood all group=SUM(Total blood get from NBC[group!]) Units: unit

get from NBC[group]= DELAY FIXED(((actual fraction[group]*order to NBC[group])),waiting time [group] , 0) Units: unit/Day

group: O,B,A,AB

in expire[O]=IF THEN ELSE(Time<=31, expire[O]+rt expire[O], IF THEN ELSE(MODULO (Time,1)<0.08, expire[O]+rt expire[O], 0))

in expire[B]=IF THEN ELSE(Time<=31, expire[B]+rt expire[B], IF THEN ELSE(MODULO (Time,1)<0.08, expire[B]+rt expire[B], 0))

in expire[A]=IF THEN ELSE(Time<=31, expire[A]+rt expire[A], IF THEN ELSE(Time<=61, IF THEN ELSE(MODULO(Time,1)<0.6, expire[A]+rt expire[A], 0), IF THEN ELSE(MODULO(Time,1)<0.095, expire[A]+rt expire[A], 0)))

in expire[AB]=IF THEN ELSE(Time<=31, expire[AB]+rt expire[AB], IF THEN ELSE(MODULO (Time,1)<0.5, expire[AB]+rt expire[AB],0)) Units: unit/Day

in tune[group]= MODULO(tune decimal[group],1) Units: unit/Day

initial order to NBC[group]=IF THEN ELSE(MODULO(Time,1)<=0.8, IF THEN ELSE((Unfilled order [group]+addition order[group])<(Total WB[group]+Qty on order[group]) ,IF THEN ELSE(desire inventory[group] >(Total WB[group]+Qty on order[group]-Unfilled order[group]-addition order[group]) ,desire inventory[group]-(Total WB[group]+Qty on order[group]-Unfilled order[group]-addition order[group]),0), desire inventory[group]+Unfilled order[group]+addition order[group]-Total WB[group]-Qty on order[group]) , 0) Units: unit

INITIAL TIME = 0 Units: Day

int cb19[group]= 3,0,0,0 Units: unit

int wb10[group]= 2,0,0,0 Units: unit

int wb11[group]= 5,0,5,2 Units: unit

int wb15[group]= 3,6,0,0 Units: unit

int wb16[group]= 4,0,2,0 Units: unit
 int wb17[group]= 4,8,3,0 Units: **undefined**
 int wb18[group]= 5,2,3,0 Units: unit
 int wb19[group]= 0,5,3,0 Units: unit
 int wb20[group]= 0,3,0,0 Units: unit
 int wb5[group]= 0,0,0,8 Units: **undefined**
 int wb6[group]= 5,0,0,0 Units: unit
 int wb7[group]= 4,3,3,4 Units: unit
 int wb9[group]= 2,3,3,0 Units: unit

money lose for expire blood[group]=blood price per unit*Expire blood[group] Units: bath
 money lose for expire blood all group=blcod price per unit*expire blood all group Units: bath

order[group]=order to NBC[group]*actual fraction[group] Units: unit/Day
 order from doctor[O]:=GET XLS DATA('blood.xls' , 'order from doctor' , '1' , 'B5')
 order from doctor[B]:=GET XLS DATA('blood.xls' , 'order from doctor' , '1' , 'B3')
 order from doctor[A]:=GET XLS DATA('blood.xls' , 'order from doctor' , '1' , 'B2')
 order from doctor[AB]:=GET XLS DATA('blood.xls' , 'order from doctor' , '1' , 'B4')
 Units: unit/Day

order to NBC[group]=initial order to NBC[group] Units: unit

order to NBC all group= SUM(Total order[group!]) nits: unit

out from CB1[group]=MAX(0,IF THEN ELSE((cb1to0d1[group]+cb1to0d2[group]+cb1to0d3
 [group]+cb1to1[group])>CB1[group],INTEGER(CB1[group]),cb1to0d1[group]+cb1to0d2[group]+
 cb1to0d3[group]+cb1to1[group])) Units: unit/Day

out from CB10[group]=MAX(0, IF THEN ELSE((cb10to7[group]+cb10to8[group]+cb10to9[group])>
 CB10[group],INTEGER(CB10[group]),cb10to10[group]+cb10to7[group]+cb10to8[group]+cb10to
 9[group])) Units: unit/Day

out from CB11[group]=MAX(0,IF THEN ELSE((cb11to10[group]+cb11to11[group]+cb11to8
 [group]+cb11to9[group]) >CB11[group], INTEGER(CB11[group]) ,cb11to10
 [group]+cb11to11[group]+cb11to8[group]+cb11to9[group])) Units: unit/Day

out from CB12[group]=MAX(0,IF THEN ELSE((cb12to10[group]+cb12to11[group]+cb12to12
 [group]+cb12to9[group]) >CB12[group],INTEGER(CB12[group]) , cb12to10
 [group]+cb12to11[group]+cb12to12[group]+cb12to9[group])) Units: unit/Day

out from CB13[group]=MAX(0,IF THEN ELSE((cb13to10[group]+cb13to11[group]+cb13to12
 [group]+cb13to13[group])>CB13[group],INTEGER(CB13[group])
 ,cb13to10[group]+cb13to11[group]+cb13to12[group]+cb13to13[group])) Units: unit/Day

out from CB14[group]=MAX(0,IF THEN ELSE((cb14to11[group]+cb14to12[group]+cb14to13
 [group]+cb14to14[group])>CB14[group],INTEGER(CB14[group])
 ,cb14to11[group]+cb14to12[group]+cb14to13[group]+cb14to14[group])) Units: unit/Day

out from CB15[group]= MAX(0,IF THEN ELSE((cb15to13[group]+cb15to14[group]+cb15to15
 [group])>CB15[group],
 INTEGER(CB15[group]),cb15to12[group]+cb15to13[group]+cb15to14[group]+cb15to15[group]))
 Units: unit/Day

out from CB16[group]= MAX(0,IF THEN ELSE((cb16to13[group]+cb16to14[group]+cb16to15
 [group]+cb16to16[group]) >CB16[group] ,INTEGER(CB16[group]) ,cb16to13
 [group]+cb16to14[group]+cb16to15[group]+cb16to16[group])) Units: unit/Day

out from CB17[group]=MAX(0,IF THEN ELSE((cb17to14[group]+cb17to15[group]+cb17to16
 [group]+cb17to17[group])>CB17[group],INTEGER(CB17[group])
 ,cb17to14[group]+cb17to15[group]+cb17to16[group]+cb17to17[group])) Units: unit/Day

out from CB18[group]=MAX(0,IF THEN ELSE((cb18to15[group]+cb18to16[group]+cb18to17
 [group]+cb18to18[group]) >CB18[group] , INTEGER(CB18[group]) ,cb18to15
 [group]+cb18to16[group]+cb18to17[group]+cb18to18[group])) Units: unit/Day

out from CB19[group]=MAX(0, IF THEN ELSE((cb19to16[group]+cb19to17[group]+cb19to18
 [group]+cb19to19[group])>CB19[group],
 INTEGER(CB19[group]),cb19to16[group]+cb19to17[group]+cb19to18[group]+cb19to19[group]))
 Units: unit/Day

out from CB20[group]=MAX(0,IF THEN ELSE((cb2to0d2[group]+cb2to0d3[group]+cb2to1
 [group]+cb2to2[group]) >CB20[group] , INTEGER(CB20[group]), cb2to0d2[group]+cb2to0d3
 [group]+cb2to1[group]+cb2to2[group])) Units: unit/Day

out from CB20[group]=MAX(0, IF THEN ELSE((cb20to17[group]+cb20to18[group]+cb20to19
 [group]+cb20to20[group]) >CB20[group] , INTEGER(CB20[group]), cb20to17
 [group]+cb20to18[group]+cb20to19[group]+cb20to20[group])) Units: unit/Day

out from CB3[group]=MAX(0,IF THEN ELSE((cb3to0[group]+cb3to1[group]+cb3to2[group]+cb3to3
 [group]) >CB3[group] , INTEGER(CB3[group]),cb3to0[group]+cb3to1[group]+cb3to2
 [group]+cb3to3[group])) Units: unit/Day

out from CB4[group]=MAX(0,IF THEN ELSE((cb4to1[group]+cb4to2[group]+cb4to3[group]+cb4to4
 [group]) >CB4[group] , INTEGER(CB4[group]),cb4to1[group]+cb4to2[group]+cb4to3
 [group]+cb4to4[group])) Units: unit/Day

out from CB5[group]=MAX(0,IF THEN ELSE((cb5to2[group]+cb5to3[group]+cb5to4[group]+cb5to5
 [group]) >CB5[group] , INTEGER(CB5[group]), cb5to2[group]+cb5to3[group]+cb5to4
 [group]+cb5to5[group])) Units: unit/Day

out from CB6[group]=MAX(0,IF THEN ELSE((cb6to3[group]+cb6to4[group]+cb6to5[group]+cb6to6

$[group] > CB6[group], \text{INTEGER}(CB6[group])$
 $, cb6to3[group] + cb6to4[group] + cb6to5[group] + cb6to6[group])$ Units: unit/Day
 out from $CB7[group] = \text{IF THEN ELSE}(CB7[group] \leq 0, \text{INTEGER}(CB7[group]) , cb7to4[group] + cb7to5$
 $[group] + cb7to6[group] + cb7to7[group])$ Units: unit/Day
 out from $CB8[group] = \text{MAX}(0, \text{IF THEN ELSE}((cb8to5[group] + cb8to6[group] + cb8to7[group] + cb8to8$
 $[group]) > CB8[group], \text{INTEGER}(CB8[group])$
 $, cb8to5[group] + cb8to6[group] + cb8to7[group] + cb8to8[group])$ Units: unit/Day
 out from $CB9[group] = \text{MAX}(0, \text{IF THEN ELSE}((cb9to6[group] + cb9to7[group] + cb9to8[group] + cb9to9$
 $[group]) > CB9[group], \text{INTEGER}(CB9[group]) , cb9to6[group] + cb9to7[group] + cb9to8$
 $[group] + cb9to9[group]))$ Units: unit/Day

percent of expire blood $[group] = \text{IF THEN ELSE}(\text{Total blood get from NBC}[group] = 0, 0 , (\text{Expire blood}$
 $[group] / \text{Total blood get from NBC}[group]) * 100)$ Units: Dmnl

percent of expire blood all group = $\text{IF THEN ELSE}(\text{get blood all group} = 0, 0 , (\text{expire blood all}$
 $group / \text{get blood all group}) * 100)$ Units: Dmnl

Qty on order $[group] = \text{INTEG}(+order[group] - \text{get from NBC}[group], 0)$ Units: unit

$\text{rand}[O] = \text{INTEGER}(\text{RANDOM UNIFORM}(1, 21, 0))$

$\text{rand}[B] = \text{INTEGER}(\text{RANDOM UNIFORM}(1, 21, 1))$

$\text{rand}[A] = \text{INTEGER}(\text{RANDOM UNIFORM}(1, 21, 2))$

$\text{rand}[AB] = \text{INTEGER}(\text{RANDOM UNIFORM}(1, 21, 3))$ Units: Dmnl

$\text{rand rt0}[O] = \text{IF THEN ELSE}(\text{Time} \leq 31, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.3851, 0.9074, 0)) ,$
 $\text{IF THEN ELSE}(\text{Time} \leq 61, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.255, 0.8422, 0)) ,$
 $\text{IF THEN ELSE}(\text{Time} \leq 92, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.4545, 0.7856, 0)) ,$
 $\text{IF THEN ELSE}(\text{Time} \leq 122, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.4958, 0.8379, 0)) ,$
 $\text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.3819, 0.8278, 0)))$

$\text{rand rt0}[B] = \text{IF THEN ELSE}(\text{Time} \leq 31, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.44262, 0.7164, 0)) , \text{IF THEN}$
 $\text{ELSE}(\text{Time} \leq 61, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.2823, 0.6815, 0)) , \text{IF THEN ELSE}(\text{Time} \leq$
 $92, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.7698, 1.1183, 0)) , \text{IF THEN ELSE}(\text{Time} \leq 122,$
 $\text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.9658, 1.1059, 0)) , \text{INTEGER}(\text{RANDOM NORMAL}(0, 4,$
 $1.5044, 0.8977, 0)))$

$\text{rand rt0}[A] = \text{IF THEN ELSE}(\text{Time} \leq 31, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.6667, 0.9684, 0)) , \text{IF THEN}$
 $\text{ELSE}(\text{Time} \leq 61, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.6917, 0.8773, 0)) , \text{IF THEN ELSE}(\text{Time} \leq$
 $92, \text{INTEGER}(\text{RANDOM NORMAL}(0, 4, 1.1013, 0.6717, 0)) , \text{IF THEN ELSE}(\text{Time} \leq 122,$

```

INTEGER(RANDOM NORMAL(0, 4, 1.5556, 0.9132, 0)),INTEGER(RANDOM NORMAL(0, 4, 2.1194,
1.1351, 0) ))))
rand rt0[AB]=IF THEN ELSE(Time<= 31,INTEGER(RANDOM NORMAL(0, 4, 1.3214 , 0.6118 , 0 )) , IF THEN
ELSE(Time<= 61, INTEGER(RANDOM NORMAL(0, 4, 1.5484 , 1.3125 ,0)), , IF THEN ELSE(Time<=
92, INTEGER(RANDOM NORMAL(0, 4, 1.6 , 1.1832,0 )), IF THEN ELSE( Time<=122,
INTEGER(RANDOM NORMAL(0, 4, 1.0345, 0.6258, 0)), INTEGER(RANDOM NORMAL(0, 4, 1.2,
0.8165, 0) )))) Units: Dmnl

```

ค่า rand rt1 ถึง rand rt19 ของโลหิตแต่ละหมู่มีสมการเหมือน rand rt0 ของแต่ละหมู่ ต่างกันเพียงค่า seed ของฟังก์ชัน RANDOM NORMAL {RANDOM NORMAL(min,max,mean,stdv,seed)} โดย seed ของ rand rt (n) = n โดยค่า n คือค่าที่อยู่หลัง rand rt

```

return rate[group]=rt expire[group]+rt1[group]+rt10[group]+rt11[group]+rt12[group]+rt13
[group]+rt14[group]+rt15[group]+rt16[group]+rt17[group]+rt18[group]+rt19[group]+rt2
[group]+rt20[group]+rt3[group]+rt4[group]+rt5[group]+rt6[group]+rt7[group]+rt8
[group]+rt9[group] Units: **undefined**

```

```

rt expire[group]= rt1to exp[group]+rt2to exp[group]+rt3d3[group] Units: unit/Day
rt in1[group]= DELAY FIXED (MAX(0,cb1to1[group]) ,rt time ,0) Units: unit/Day
rt in10[group]= DELAY FIXED (MAX(0,cb10to10[group]) ,rt time ,0) Units: unit/Day
rt in11[group]= DELAY FIXED (MAX(0,cb11to11[group]) ,rt time ,0) Units: unit/Day
rt in12[group]= DELAY FIXED (MAX(0,cb12to12[group]) ,rt time ,0) Units: unit/Day
rt in13[group]= DELAY FIXED (MAX(0,cb13to13[group]),rt time ,0) Units: unit/Day
rt in14[group]= DELAY FIXED (MAX(0,cb14to14[group]) ,rt time ,0) Units: unit/Day
rt in15[group]= DELAY FIXED (MAX(0,cb15to15[group]) ,rt time ,0) Units: unit/Day
rt in16[group]= DELAY FIXED (MAX(0,cb16to16[group]),rt time ,0) Units: unit/Day
rt in17[group]= DELAY FIXED (MAX(0,cb17to17[group]) ,rt time ,0) Units: unit/Day
rt in18[group]= DELAY FIXED (MAX(0,cb18to18[group]),rt time ,0) Units: unit/Day
rt in19[group]= DELAY FIXED (MAX(0,cb19to19[group]),rt time,0) Units: unit/Day
rt in2[group]= DELAY FIXED (MAX(0,cb2to2[group]) ,rt time ,0) Units: unit/Day
rt in3[group]= DELAY FIXED (MAX(0,cb3to3[group]) ,rt time ,0) Units: unit/Day
rt in4[group]= DELAY FIXED (MAX(0,cb4to4[group]) ,rt time ,0) Units: unit/Day
rt in5[group]= DELAY FIXED (MAX(0,cb5to5[group]) ,rt time ,0) Units: unit/Day
rt in6[group]= DELAY FIXED (MAX(0,cb6to6[group]) ,rt time ,0) Units: unit/Day
rt in7[group]= DELAY FIXED (MAX(0,cb7to7[group]),rt time ,0) Units: unit/Day
rt in8[group]= DELAY FIXED (MAX(0,cb8to8[group]) ,rt time ,0) Units: unit/Day
rt in9[group]= DELAY FIXED (MAX(0,cb9to9[group]) ,rt time ,0) Units: unit/Day

```

rt time= 0.5 Units: Day
 rt1[group]=rt in1[group]+rt2to1[group]+rt3to1[group]+rt4to1[group] Units: unit/Day
 rt10[group]=rt in10[group]+rt11to10[group]+rt12to10[group]+rt13to10[group] Units: unit/Day
 rt10to7[group]= DELAY FIXED (MAX(0,cb10to7[group]),3,0) Units: unit/Day
 rt10to8[group]= DELAY FIXED (MAX(0,cb10to8[group]),2,0) Units: unit/Day
 rt10to9[group]= DELAY FIXED (MAX(0,cb10to9[group]),1,0) Units: unit/Day
 rt11[group]=rt in11[group]+rt12to11[group]+rt13to11[group]+rt14to11[group] Units: unit/Day
 rt11to10[group]= DELAY FIXED (MAX(0,cb11to10[group]),1,0) Units: unit/Day
 rt11to8[group]= DELAY FIXED (MAX(0,cb11to8[group]),3,0) Units: unit/Day
 rt11to9[group]= DELAY FIXED (MAX(0,cb11to9[group]),2,0) Units: unit/Day
 rt12[group]=rt in12[group]+rt13to12[group]+rt14to12[group]+rt15to12[group] Units: unit/Day
 rt12to10[group]= DELAY FIXED (MAX(0,cb12to10[group]),2,0) Units: unit/Day
 rt12to11[group]= DELAY FIXED (MAX(0,cb12to11[group]),1,0) Units: unit/Day
 rt12to9[group]= DELAY FIXED (MAX(0,cb12to9[group]),3,0) Units: unit/Day
 rt13[group]=rt in13[group]+rt14to13[group]+rt15to13[group]+rt16to13[group] Units: unit/Day
 rt13to10[group]= DELAY FIXED (MAX(0,cb13to10[group]),3,0) Units: unit/Day
 rt13to11[group]= DELAY FIXED (MAX(0,cb13to11[group]),2,0) Units: unit/Day
 rt13to12[group]= DELAY FIXED (MAX(0,cb13to12[group]),1,0) Units: unit/Day
 rt14[group]=rt in14[group]+rt15to14[group]+rt16to14[group]+rt17to14[group] Units: unit/Day
 rt14to11[group]= DELAY FIXED (MAX(0,cb14to11[group]),3,0) Units: unit/Day
 rt14to12[group]= DELAY FIXED (MAX(0,cb14to12[group]),2,0) Units: unit/Day
 rt14to13[group]= DELAY FIXED (MAX(0,cb14to13[group]),1,0) Units: unit/Day
 rt15[group]=rt in15[group]+rt16to15[group]+rt17to15[group]+rt18to15[group] Units: unit/Day
 rt15to12[group]= DELAY FIXED (MAX(0,cb15to12[group]),3,0) Units: unit/Day
 rt15to13[group]= DELAY FIXED (MAX(0,cb15to13[group]),2,0) Units: unit/Day
 rt15to14[group]= DELAY FIXED (MAX(0,cb15to14[group]),1,0) Units: unit/Day
 rt16[group]=rt in16[group]+rt17to16[group]+rt18to16[group]+rt19to16[group] Units: unit/Day
 rt16to13[group]= DELAY FIXED (MAX(0,cb16to13[group]),3,0) Units: unit/Day
 rt16to14[group]= DELAY FIXED (MAX(0,cb16to14[group]),2,0) Units: unit/Day
 rt16to15[group]= DELAY FIXED (MAX(0,cb16to15[group]),1,0) Units: unit/Day
 rt17[group]=rt in17[group]+rt18to17[group]+rt19to17[group]+rt20to17[group] Units: unit/Day
 rt17to14[group]= DELAY FIXED (MAX(0,cb17to14[group]),3,0) Units: unit/Day
 rt17to15[group]= DELAY FIXED (MAX(0,cb17to15[group]),2,0) Units: unit/Day
 rt17to16[group]= DELAY FIXED (MAX(0,cb17to16[group]),1,0) Units: unit/Day
 rt18[group]=rt in18[group]+rt19to18[group]+rt20to18[group] Units: unit/Day
 rt18to15[group]= DELAY FIXED (MAX(0,cb18to15[group]),3,0) Units: unit/Day

$rt_{18to16}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{18to16}[group]), 2, 0)$ Units: unit/Day
 $rt_{18to17}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{18to17}[group]), 1, 0)$ Units: unit/Day
 $rt_{19}[group] = rt_{20to19}[group] + rt_{in19}[group]$ Units: unit/Day
 $rt_{19to16}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{19to16}[group]), 3, 0)$ Units: unit/Day
 $rt_{19to17}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{19to17}[group]), 2, 0)$ Units: unit/Day
 $rt_{19to18}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{19to18}[group]), 1, 0)$ Units: unit/Day
 $rt_{1d1}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{1to0d1}[group]), rt_{time}, 0)$ Units: unit/Day
 $rt_{1d2}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{1to0d2}[group]), 2, 0)$ Units: unit/Day
 $rt_{1d3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{1to0d3}[group]), 3, 0)$ Units: unit/Day
 $rt_{1to\ exp}[group] = rt_{1d1}[group] + rt_{1d2}[group] + rt_{1d3}[group]$ Units: unit/Day
 $rt_{2}[group] = rt_{in2}[group] + rt_{3to2}[group] + rt_{4to2}[group] + rt_{5to2}[group]$ Units: unit/Day
 $rt_{20}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{20to20}[group]), rt_{time}, 0)$ Units: unit/Day
 $rt_{20to17}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{20to17}[group]), 3, 0)$ Units: unit/Day
 $rt_{20to18}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{20to18}[group]), 2, 0)$ Units: unit/Day
 $rt_{20to19}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{20to19}[group]), 1, 0)$ Units: unit/Day
 $rt_{2d2}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{2to0d2}[group]), 2, 0)$ Units: unit/Day
 $rt_{2d3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{2to0d3}[group]), 3, 0)$ Units: unit/Day
 $rt_{2to\ exp}[group] = rt_{2d2}[group] + rt_{2d3}[group]$ Units: unit/Day
 $rt_{2to1}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{2to1}[group]), 1, 0)$ Units: unit/Day
 $rt_{3}[group] = rt_{in3}[group] + rt_{4to3}[group] + rt_{5to3}[group] + rt_{6to3}[group]$ Units: unit/Day
 $rt_{3d3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{3to0}[group]), 3, 0)$ Units: unit/Day
 $rt_{3to1}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{3to1}[group]), 2, 0)$ Units: unit/Day
 $rt_{3to2}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{3to2}[group]), 1, 0)$ Units: unit/Day
 $rt_{4}[group] = rt_{in4}[group] + rt_{5to4}[group] + rt_{6to4}[group] + rt_{7to4}[group]$ Units: unit/Day
 $rt_{4to1}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{4to1}[group]), 3, 0)$ Units: unit/Day
 $rt_{4to2}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{4to2}[group]), 2, 0)$ Units: unit/Day
 $rt_{4to3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{4to3}[group]), 1, 0)$ Units: unit/Day
 $rt_{5}[group] = rt_{in5}[group] + rt_{6to5}[group] + rt_{7to5}[group] + rt_{8to5}[group]$ Units: unit/Day
 $rt_{5to2}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{5to2}[group]), 3, 0)$ Units: unit/Day
 $rt_{5to3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{5to3}[group]), 2, 0)$ Units: unit/Day
 $rt_{5to4}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{5to4}[group]), 1, 0)$ Units: unit/Day
 $rt_{6}[group] = rt_{in6}[group] + rt_{7to6}[group] + rt_{8to6}[group] + rt_{9to6}[group]$ Units: unit/Day
 $rt_{6to3}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{6to3}[group]), 3, 0)$ Units: unit/Day
 $rt_{6to4}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{6to4}[group]), 2, 0)$ Units: unit/Day
 $rt_{6to5}[group] = \text{DELAY FIXED} (\text{MAX}(0, cb_{6to5}[group]), 1, 0)$ Units: unit/Day
 $rt_{7}[group] = rt_{in7}[group] + rt_{8to7}[group] + rt_{9to7}[group] + rt_{10to7}[group]$ Units: unit/Day

```

rt7to4[group]= DELAY FIXED (MAX(0,cb7to4[group]) ,3,0) Units: unit/Day
rt7to5[group]= DELAY FIXED (MAX(0,cb7to5[group]) ,2,0) Units: unit/Day
rt7to6[group]= DELAY FIXED (MAX(0,cb7to6[group]) , 1,0) Units: unit/Day
rt8[group]=rt in8[group]+rt10to8[group]+rt11to8[group]+rt9to8[group] Units: unit/Day
rt8to5[group]= DELAY FIXED (MAX(0,cb8to5[group]) ,3,0) Units: unit/Day
rt8to6[group]= DELAY FIXED (MAX(0,cb8to6[group]) ,2,0) Units: unit/Day
rt8to7[group]= DELAY FIXED (MAX(0,cb8to7[group]),1,0) Units: unit/Day
rt9[group]=rt in9[group]+rt10to9[group]+rt11to9[group]+rt12to9[group] Units: unit/Day
rt9to6[group]= DELAY FIXED (MAX(0,cb9to6[group]),3,0) Units: unit/Day
rt9to7[group]= DELAY FIXED (MAX(0,cb9to7[group]) , 2,0) Units: unit/Day
rt9to8[group]= DELAY FIXED (MAX(0,cb9to8[group]),1,0) Units: unit/Day

```

SAVEPER = TIME STEP Units: Day [0,?]

```

sub1[group]=IF THEN ELSE(rand[group]=1,CB1[group] ,IF THEN ELSE(rand[group]=2,CB2[group]
,IF THEN ELSE(rand[group]=3,CB3[group] ,IF THEN ELSE(rand[group]=4, CB4[group] , IF
THEN ELSE(rand[group]=5,CB5[group] ,IF THEN ELSE(rand[group]=6,CB6[group],IF THEN
ELSE(rand[group]=7, CB7[group], IF THEN ELSE(rand[group]=8,CB8[group],IF THEN ELSE
(rand[group]=9,CB9[group] ,IF THEN ELSE(rand[group]=10,CB10[group] ,IF THEN
ELSE(rand[group]=11,CB11[group],IF THEN ELSE(rand[group]=12,CB12[group] ,IF THEN
ELSE(rand[group]=13,CB13[group] ,IF THEN ELSE(rand[group]=14,CB14[group] ,IF THEN
ELSE(rand[group]=15,CB15[group] ,IF THEN ELSE(rand[group]=16,CB16[group] ,IF THEN
ELSE(rand[group]=17,CB17[group] ,IF THEN ELSE(rand[group]=18,CB18[group] , IF THEN
ELSE(rand[group]=19,CB19[group] ,CB20[group]))))))) Units: unit/Day

```

สมการของ sub2[group] ถึง sub20[group]มีลักษณะเช่นเดียวกับสมการของ sub1[group] โดยค่าของ sub n[group] ถูกกำหนดโดยค่า rand[group] คือ ถ้า rand = 1 sub1 จะเท่ากับ CB1 sub2 จะเท่ากับ CB2 ไล่ไปเรื่อยๆ จนถึง 20 และวนกลับมา 1 อีกครั้ง ตัวอย่างเช่น rand =10 sub1จะเท่ากับ CB10 sub2 จะเท่ากับ CB11 และ sub20 จะเท่ากับ CB9

TIME STEP = 0.03125 Units: Day [0,?]

```

to1[group]=INTEGER(IF THEN ELSE(WB2[group]<=0, 0 , WB2[group])) Units: unit/Day
to10[group]=INTEGER( IF THEN ELSE(WB11[group]<=0,0 , WB11[group] )) Units: unit/Day
to11[group]=INTEGER(IF THEN ELSE(WB12[group]<=0,0 , WB12[group] )) Units: unit/Day
to12[group]=INTEGER( IF THEN ELSE(WB13[group]<=0, 0,WB13[group])) Units: unit/Day
to13[group]=INTEGER(IF THEN ELSE(WB14[group]<=0, 0 , WB14[group] )) Units: unit/Day
to14[group]=INTEGER(IF THEN ELSE(WB15[group]<=0, 0 ,WB15[group] )) Units: unit/Day

```

$to15[group]=INTEGER(IF THEN ELSE(WB16[group]<=0,0 ,WB16[group]))$ Units: unit/Day
 $to16[group]=INTEGER(IF THEN ELSE(WB17[group]<=0, 0 ,WB17[group]))$ Units: unit/Day
 $to17[group]=INTEGER(IF THEN ELSE(WB18[group]<=0,0 , WB18[group]))$ Units: unit/Day
 $to18[group]=INTEGER(IF THEN ELSE(WB19[group]<=0,0 , WB19[group]))$ Units: unit/Day
 $to19[group]=INTEGER(IF THEN ELSE(WB20[group]<=0, 0, WB20[group]))$ Units: unit/Day
 $to2[group]=INTEGER(IF THEN ELSE(WB3[group]<=0,0 , WB3[group]))$ Units: unit/Day
 $to3[group]=INTEGER(IF THEN ELSE(WB4[group]<=0, 0 , WB4[group]))$ Units: unit/Day
 $to4[group]=INTEGER(IF THEN ELSE(WB5[group]<=0 ,0 ,WB5[group]))$ Units: unit/Day
 $to5[group]=INTEGER(IF THEN ELSE(WB6[group]<=0, 0 , WB6[group]))$ Units: unit/Day
 $to6[group]=INTEGER(IF THEN ELSE(WB7[group]<=0,0 ,WB7[group]))$ Units: unit/Day
 $to7[group]=INTEGER(IF THEN ELSE(WB8[group]<=0,0 , WB8[group]))$ Units: unit/Day
 $to8[group]=INTEGER(IF THEN ELSE(WB9[group]<=0,0 ,WB9[group]))$ Units: unit/Day
 $to9[group]=INTEGER(IF THEN ELSE(WB10[group]<=0, 0, WB10[group]))$ Units: unit/Day
 $Total\ blood\ get\ from\ NBC[group]=\ INTEG\ (\ get\ blood[group],0)$ Units: unit
 $Total\ cross\ blood[group]=\ INTEG\ (\ fill\ order[group],\ 0)$ Units: unit
 $Total\ crossblood[group]=\ CB1[group]+CB10[group]+CB11[group]+CB12[group]+CB13$
 $\ [group]+CB14[group]+CB15[group]+CB16[group]+CB17[group]+CB18[group]+CB19[group]+C$
 $B2[group]+CB20[group]+CB3[group]+CB4[group]+CB5[group]+CB6[group]+CB7[group]+CB8[g$
 $roup]+CB9[group]$ Units: unit
 $Total\ order[group]=\ INTEG\ (order\ to\ NBC[group],0)$ Units: unit
 $Total\ return[group]=\ INTEG\ (return\ rate[group],0)$ Units: **undefined**
 $Total\ WB[group]=WB1[group]+WB2[group]+WB3[group]+WB4[group]+WB5[group]+WB6$
 $\ [group]+WB7[group]+WB8[group]+WB9[group]+WB10[group]+WB11[group]+WB12$
 $\ [group]+WB13[group]+WB14[group]+WB15[group]+WB16[group]+WB17[group]+WB18$
 $\ [group]+WB19[group]+WB20[group]-tune[group]$ Units: unit
 $tune[group]=\ INTEG\ (in\ tune[group],0)$ Units: unit
 $tune\ decimal[group]=actual\ use[group]-use\ rate[group]$ Units: unit/Day
 $tuner[group]=MODULO(WB1[group],1)+MODULO(WB10[group],1)+MODULO(WB11$
 $\ [group],1)+MODULO(WB12[group],1)+MODULO(WB13[group],1)+MODULO(WB14[group],1)+$
 $MODULO(WB15[group],1)+MODULO(WB16[group],1)+MODULO(WB17[group],1)+MODULO(WB$
 $18[group],1)+MODULO(WB19[group],1)+MODULO(WB20[group],1)+MODULO(WB2[group],1)+$
 $MODULO(WB3[group],1)+MODULO(WB4[group],1)+MODULO(WB5[group],1)+MODULO(WB6[gr$
 $oup],1)+MODULO(WB7[group],1)+MODULO(WB8[group],1)+MODULO(WB9[group],1)$ Units: unit
 $Unfilled\ order[group]=\ INTEG\ (+order\ from\ doctor[group]-fill\ order[group],0)$ Units: unit
 $use\ rate[group]=\ use1[group]+use10[group]+use11[group]+use12[group]+use13[group]+use14$

[group]+use15[group]+use16[group]+use17[group]+use18[group]+use19[group]+use2[group]+
use20[group]+use3[group]+use4[group]+use5[group]+use6[group]+use7[group]+use8[group]+u
se9[group] Units: unit/Day

use1[group]=IF THEN ELSE(rand[group]=1,useP1[group] ,IF THEN ELSE(rand[group]=2,useP20
[group] ,IF THEN ELSE(rand[group]=3,useP19[group] ,IF THEN ELSE(rand[group]=4,
useP18[group],IF THEN ELSE(rand[group]=5,useP17[group] ,IF THEN
ELSE(rand[group]=6,useP16[group],IF THEN ELSE(rand[group]=7, useP15[group], IF THEN
ELSE(rand[group]=8,useP14[group] ,IF THEN ELSE(rand[group]=9,useP13[group] , IF THEN
ELSE(rand[group]=10,useP12[group] ,IF THEN ELSE(rand[group]=11,useP11[group],IF THEN
ELSE(rand[group]=12,useP10[group] ,IF THEN ELSE(rand[group]=13,useP9[group] ,IF THEN
ELSE(rand[group]=14,useP8[group] ,IF THEN ELSE(rand[group]=15,useP7[group],IF THEN
ELSE(rand[group]=16,useP6[group] ,IF THEN ELSE(rand[group]=17,useP5[group] ,IF THEN
ELSE(rand[group]=18,useP4[group] ,IF THEN ELSE(rand[group]=19,useP3[group] ,useP2[group])
))))))))) Units: unit/Day

สมการของ use 2 ถึง use20 มีลักษณะเช่นเดียวกับสมการของ use1 โดยค่า use n[group] จะถูกกำหนด
จากค่า rand[group] คือ ถ้า rand =1 use1 จะเท่ากับ useP1 เนื่องจาก sub1=CB1 (use1คือโลหิตที่ออกจาก CB1)
นั่นก็คือให้ priority เป็นอันดับแรก ดังนั้นถ้า rand=10 use5 จะเท่ากับ useP16

Used blood[group]= INTEG (actual use[group],0) Units: unit
used blood all group=SUM(Used blood[group!]) Units: unit
useP1[group]=INTEG(MAX(0,IF THEN ELSE(actual use[group]=0 :OR: sub1[group]<=0, 0, IF
THEN ELSE(x1[group] <=0, actual use[group] ,sub1[group])))) Units: unit/Day
useP10[group]= INTEG(MAX(0,IF THEN ELSE(x9[group]<=0 :OR: sub10[group]<=0, 0 ,IF THEN
ELSE(x10[group]<=0, x9[group] , sub10[group])))) Units: unit/Day
useP11[group]= INTEG(MAX(0,IF THEN ELSE(x10[group]<=0 :OR: sub11[group]<=0, 0, IF THEN
ELSE(x11[group]<=0, x10[group] ,sub11[group])))) Units: unit/Day
useP12[group]=INTEG(MAX(0,IF THEN ELSE(x11[group]<=0 :OR: sub12[group]<=0, 0, IF THEN
ELSE(x12[group]<=0, x11[group] ,sub12[group])))) Units: unit/Day
useP13[group]= INTEG(MAX(0,IF THEN ELSE(x12[group]<=0 :OR: sub13[group]<=0, 0 ,IF
THEN ELSE(x13[group]<=0, x12[group] ,sub13[group])))) Units: unit/Day
useP14[group]= INTEG(MAX(0,IF THEN ELSE(x13[group]<=0 :OR: sub14[group]<=0, 0 ,IF
THEN ELSE(x14[group]<=0, x13[group] ,sub14[group])))) Units: unit/Day
useP15[group]= INTEG(MAX(0,IF THEN ELSE(x14[group]<=0 :OR:sub15[group]<=0,0 ,IF THEN
ELSE(x15[group]<=0, x14[group] , sub15[group])))) Units: unit/Day
useP16[group]= INTEG(MAX(0,IF THEN ELSE(x15[group]<=0 :OR: sub16[group]<=0, 0 ,IF THEN

ELSE(x16[group]<=0, x15[group], sub16[group]))) Units: unit/Day
 useP17[group]= INTEGER(MAX(0, IF THEN ELSE(x16[group]<=0 :OR: sub17[group]<=0, 0 ,IF THEN
 ELSE(x17[group]<=0, x16[group], sub17[group])))) Units: unit/Day
 useP18[group]= INTEGER(MAX(0,IF THEN ELSE(x17[group]<=0 :OR: sub18[group]<=0, 0, IF THEN
 ELSE(x18[group]<=0,x17[group] , sub18[group]))) Units: unit/Day
 useP19[group]= INTEGER(MAX(0,IF THEN ELSE(x18[group]<=0 :OR: sub19[group]<=0, 0 , IF
 THEN ELSE(x19[group]<=0, x18[group] , sub19[group])))) Units: unit/Day
 useP2[group]=INTEGER(MAX(0,IF THEN ELSE(x1[group]<=0 :OR: sub2[group]<=0, 0,IF THEN
 ELSE(x2[group]<= 0, x1[group] , sub2[group])))) Units: unit/Day
 useP20[group]= INTEGER(MAX(0,IF THEN ELSE(x19[group]<=0 :OR: sub20[group]<=0, 0, IF THEN
 ELSE(x20[group]<=0, x19[group], sub20[group])))) Units: unit/Day
 useP3[group]=INTEGER(MAX(0,IF THEN ELSE(x2[group] <=0 :OR: sub3[group]<=0,0, IF THEN
 ELSE(x3[group] <=0,x2[group], sub3[group])))) Units: unit/Day
 useP4[group]=INTEGER(MAX(0,IF THEN ELSE(x3[group]<= 0 :OR: sub4[group] <=0, 0, IF THEN
 ELSE(x4[group]<= 0 , x3[group] ,sub4[group])))) Units: unit/Day
 useP5[group]=INTEGER(MAX(0, IF THEN ELSE(x4[group]<=0 :OR: sub5[group]<=0, 0 , IF THEN
 ELSE(x5[group]<=0, x4[group], sub5[group])))) Units: unit/Day
 useP6[group]=INTEGER(MAX(0,IF THEN ELSE(x5[group]<=0 :OR: sub6[group]<=0, 0 , IF THEN
 ELSE(x6[group]<=0, x5[group] , sub6[group])))) Units: unit/Day
 useP7[group]=INTEGER(MAX(0,IF THEN ELSE(x6[group]<=0 :OR: sub7[group]<=0, 0 , IF THEN
 ELSE(x7[group]<=0, x6[group] , sub7[group])))) Units: unit/Day
 useP8[group]=INTEGER(MAX(0,IF THEN ELSE(x7[group]<=0 :OR: sub8[group]<=0, 0 , IF THEN
 ELSE(x8[group]<=0, x7[group], sub8[group])))) Units: unit/Day
 useP9[group]=INTEGER(MAX(0,IF THEN ELSE(x8[group]<=0 :OR: sub9[group]<=0, 0, IF THEN
 ELSE(x9[group]<=0, x8[group], sub9[group])))) Units: unit/Day
 waiting time[group]=0.04167 Units: Day
 WB Inv[group]= MAX(0,Total WB[group]-tuner[group]) Units: unit
 WB1[group]= INTEG (rt1[group]+to1[group]-c1[group]-expire[group], 0) Units: unit
 WB10[group]= INTEG (rt10[group]+to10[group]-c10[group]-to9[group],int wb10[group]) Units: unit
 WB11[group]= INTEG (rt11[group]+to11[group]-c11[group]-to10[group],int wb11[group]) Units: unit
 WB12[group]= INTEG (rt12[group]+to12[group]-c12[group]-to11[group],0) Units: unit
 WB13[group]= INTEG (rt13[group]+to13[group]-c13[group]-to12[group],0) Units: unit
 WB14[group]= INTEG (rt14[group]+to14[group]-c14[group]-to13[group],0) Units: unit
 WB15[group]= INTEG (rt15[group]+to15[group]-c15[group]-to14[group],int wb15[group]) Units: unit
 WB16[group]= INTEG (g16[group]+rt16[group]+to16[group]-c16[group]-to15[group],int wb16[group])
 Units: unit

WB17[group]= INTEG (g17[group]+rt17[group]+to17[group]-c17[group]-to16[group],int wb17
[group]) Units: unit

WB18[group]= INTEG (g18[group]+rt18[group]+to18[group]-c18[group]-to17[group],int wb18
[group]) Units: unit

WB19[group]= INTEG (g19[group]+rt19[group]+to19[group]-c19[group]-to18[group],int wb19
[group]) Units: unit

WB2[group]= INTEG (rt2[group]+to2[group]-c2[group]-to1[group],0) Units: unit

WB20[group]= INTEG (g20[group]+rt20[group]-c20[group]-to19[group],int wb20[group]) Units: unit

WB3[group]= INTEG (rt3[group]+to3[group]-c3[group]-to2[group],0) Units: unit

WB4[group]= INTEG (rt4[group]+to4[group]-c4[group]-to3[group],0) Units: unit

WB5[group]= INTEG (rt5[group]+to5[group]-c5[group]-to4[group],int wb5[group]) Units: unit

WB6[group]= INTEG (rt6[group]+to6[group]-c6[group]-to5[group],int wb6[group]) Units: unit

WB7[group]= INTEG (rt7[group]+to7[group]-c7[group]-to6[group],int wb7[group]) Units: unit

WB8[group]= INTEG (rt8[group]+to8[group]-c8[group]-to7[group],0) Units: unit

WB9[group]= INTEG (rt9[group]+to9[group]-c9[group]-to8[group],int wb9[group]) Units: unit

x1[group]=INTEG(actual use[group]) - INTEG(sub1[group]) Units: unit/Day

x10[group]=INTEG(x9[group])-INTEG(sub10[group]) Units: unit/Day

x11[group]=!INTEG(x10[group])-INTEG(sub11[group]) Units: unit/Day

x12[group]=INTEG(x11[group])-INTEG(sub12[group]) Units: unit/Day

x13[group]=INTEG(x12[group])-INTEG(sub13[group]) Units: unit/Day

x14[group]=INTEG(x13[group])-INTEG(sub14[group]) Units: unit/Day

x15[group]=INTEG(x14[group])-INTEG(sub15[group]) Units: unit/Day

x16[group]=INTEG(x15[group])-INTEG(sub16[group]) Units: unit/Day

x17[group]=INTEG(x16[group])-INTEG(sub17[group]) Units: unit/Day

x18[group]=INTEG(x17[group])-INTEG(sub18[group]) Units: unit/Day

x19[group]=INTEG(x18[group])-INTEG(sub19[group]) Units: unit/Day

x2[group]=INTEG(x1[group])-INTEG(sub2[group]) Units: unit/Day

x20[group]=INTEG(x19[group])-INTEG(sub20[group]) Units: unit/Day

x3[group]=INTEG(x2[group])-INTEG(sub3[group]) Units: unit/Day

x4[group]=INTEG(x3[group])-INTEG(sub4[group]) Units: unit/Day

x5[group]=INTEG(x4[group])-INTEG(sub5[group]) Units: unit/Day

x6[group]=INTEG(x5[group])-INTEG(sub6[group]) Units: unit/Day

x7[group]=INTEG(x6[group])-INTEG(sub7[group]) Units: unit/Day

x8[group]=INTEG(x7[group])-INTEG(sub8[group]) Units: unit/Day

x9[group]=INTEG(x8[group])-INTEG(sub9[group]) Units: unit/Day



ประวัติผู้เขียนวิทยานิพนธ์

นางสาวพัชรภา ตปนียพันธ์ เกิดเมื่อวันที่ 11 ตุลาคม 2522 เข้ารับการศึกษา
ระดับประถมศึกษาที่โรงเรียนปัญจทรัพย์ ปีการศึกษา 2528 ถึง 2533 ระดับมัธยมศึกษาที่โรงเรียน
สาธิตแห่งมหาวิทยาลัยรามคำแหง ปีการศึกษา 2534 ถึง 2538 สำเร็จการศึกษาระดับปริญญาตรี
วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหการ จากมหาวิทยาลัยเกษตรศาสตร์ เมื่อปี
การศึกษา 2542 และได้เข้าศึกษาต่อในระดับปริญญาโทในปีการศึกษา 2543 ที่คณะ
วิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมอุตสาหการ จุฬาลงกรณ์มหาวิทยาลัย