

Game Go-Fish

Algorithm explanation

Random 0-X

ฟังก์ชันการสุ่มจะใช้ rand ของ C โดยนำค่าที่สุ่มมา mod ด้วย X+1 แต่ arm ไม่มี mod เลยลดจำนวนบิตจากค่าที่สุ่มได้(เพื่อความรวดเร็ว)แล้วลบด้วย X+1 จนกว่าจะน้อยกว่า X-1

```
82 :      .global srand
83 :      .global time
84 :      .global rand
---
90 :      @ Set up rand
91 :      mov r0, #0
92 :      bl  time
93 :      bl  srand
---
640 : @ == Utility Function ==
641 : @ Random Number (0-r1) <<<<<<<<<
642 : util_rand:
643 :     push {lr}
644 :
645 :     @ Spacial Case
646 :     cmp r1, #0
647 :     moveq r0, #0
648 :     beq u_ra_exit
649 :
650 :     @ r1 += 1
651 :     add r1, r1, #1
652 :
653 :     @ Rand
654 :     push {r1}
655 :     bl  rand
656 :     pop {r1}
657 :
658 :     @ Set Max Random Value For Mod Performance
659 :     mov r2, #255
660 :     cmp r1, #8
661 :     ldrhs r2, =#1023
662 :     cmp r1, #16
663 :     ldrhs r2, =#8191
664 :     cmp r1, #32
665 :     ldrhs r2, =#32767
666 :
667 :     and r0, r2
668 :
669 :     @ Mod by r1
670 : u_ra_loop:
671 :     cmp r0, r1
672 :     blo u_ra_exit
673 :     sub r0, r0, r1
674 :     b   u_ra_loop
675 : u_ra_exit:
676 :     pop {pc}
```

Deck Shuffle

จองข้อมูล .asciz ใน .data จำนวน 52 ตัวอักษร เซ็ตค่าทุกช่องเป็นเลข 14 (ให้ถือว่าไม่มีไพ่) เลื่อนหน้าไฟตั้งแต่ 2-A แล้วนำไปเก็บไว้ใน .asciz ใน .data เป็นตัวเลขตั้งแต่ 0-12 โดยที่ 0 คือ หน้าไฟ 2, 1 คือหน้าไฟ 3 ไปเรื่อย ๆ จนถึง 12 ซึ่งก็คือหน้าไฟ A โดยการนำไปเก็บจะทำการสลับตำแหน่งที่จะเก็บตั้งแต่ 0 ถึงจำนวนไฟทั้งหมดที่ยังไม่ได้เก็บ-1 และทำการเก็บโดยการเซ็ตตำแหน่งที่ 0 และเพิ่มไปเรื่อย ๆ หากช่องนั้นมีข้อมูลที่ไม่ใช่ 14 จะทำการลดค่าตัวเลขที่สลับมา 1 แต่ถ้าใช่จะทำการข้ามช่องนั้น หากตัวเลขที่สลับมาเป็น 0 และช่องนั้นไม่ใช่ 14 จะทำการเก็บข้อมูลหน้าไฟนั้นลงไป แล้วไปยังหน้าไฟถัดไปหากวงครบ 4 ใบแล้ว วนลูปจบครบทุกหน้า

```
62 :
d_card_deck:      .asciz  "22223333444455556666777788889999TTTTJJJJQQQQKKKKAAAA"
---
508 : @ Shuffle Deck <<<<<<<<<
509 : game_deck_suff:
510 :     push {lr}
511 :     push {r4-r8}
512 :
513 :     @ (r4 = *d_card_deck,r5 = d_card_deck offset,r6 = card face,r7 = card
left,r8 = face left)
514 :
515 :     @ Clear Deck (14 = No Card)
516 :     mov r0, #14
517 :     ldr r4, =d_card_deck
518 :     mov r5, #0
519 : g_ds_cd_loop:
520 :     strb r0, [r4,r5]
521 :     add r5, r5, #1
522 :     cmp r5, #52
523 :     blo g_ds_cd_loop
524 :
525 :     @ Shuffle Deck
526 :     mov r0, #0
527 :     mov r7, #51
528 :     mov r6, #0
529 :
530 : g_ds_sd_loop1: @ Select Face
531 :     mov r8, #0
532 :
533 : g_ds_sd_loop2: @ Place All 4 Card
534 :     mov r1, r7
535 :     bl  util_rand
536 :     mov r1, #-1
```

```
537 :
538 : g_ds_sd_loop3:  @ Find Pos
539 :     add r1, r1, #1
540 :     ldrb r2, [r4,r1]
541 :     cmp r2, #14
542 :     beq g_ds_sd_l3_else
543 :
544 :     add r0, r0, #1
545 :     b   g_ds_sd_loop3
546 :
547 : g_ds_sd_l3_else:
548 :     cmp r1,r0
549 :     blo g_ds_sd_loop3
550 :
551 :     strb r6, [r4,r0]
552 :     add r8, r8, #1
553 :     sub r7, r7, #1
554 :     cmp r8, #4
555 :     blo g_ds_sd_loop2
556 :
557 :     add r6, r6, #1
558 :     cmp r6, #13
559 :     blo g_ds_sd_loop1
560 :
561 :     pop {r4-r8}
562 :     pop {pc}
```

Deal Card

การเก็บข้อมูลไฟของผู้เล่นแต่ละคนจะเก็บโดยใช้ .asciz ขนาด 13 ตัว แทนจำนวนไฟ โดยช่องที่ 1 คือ จำนวนไฟหน้าเลข 2 ไปจนถึงช่องที่ 13 คือ จำนวนไฟหน้า A

ทำการแจกไฟ โดยดูไฟที่อยู่ท้ายสุดของกอง แล้วทำการเพิ่มจำนวนไฟของผู้เล่นตามหน้าไฟ (หน้าไฟที่เก็บในกองตรงกับตำแหน่งที่เก็บจำนวนไฟของผู้เล่นพอดี) ลดจำนวนไฟในกองลง 1 แล้ววนลูปจนครบจำนวนไฟตามจำนวนผู้เล่น และวนลูปจนครบจำนวนผู้เล่น

เมื่อแจกไฟเสร็จ จะทำการหาไฟที่ครบชุด ก่อนทำการเริ่มเกม

```
58 : @ Game Data
59 : d_player_count:      .word   4
---
63 : d_card_deck_count:   .word   51
---
65 : d_player_hand:       .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0\0"
66 :                       .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0\0"
67 :                       .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0\0"
68 :                       .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0\0"
---
564 : @ Deal cards <<<<<<<<<
565 : game_deal_card:
566 :     push {lr}
567 :     push {r4-r10}
568 :
569 :     @ (r4 = *d_player_hand, r5 = d_player_count, r6 = Card count)
570 :
571 :     @ Clear hand and book
572 :     mov r0, #0
573 :     ldr r4, =d_player_hand
574 :     mov r1, r4
575 :     add r1, r1, #112
576 : g_dc_ch_loop:
577 :     str r0, [r4], #4
578 :     cmp r4, r1
579 :     blo g_dc_ch_loop
580 :
581 :     @ Player count
582 :     ldr r5, =d_player_count
583 :     ldr r5, [r5]
584 :
```

```

585 :      @ 7 Cards for 2 Players, 6 for 3P, 5 for 4P
586 :      mov r6, #7
587 :      mov r2, #37
588 :      cmp r5, #3
589 :      moveq r6, #6
590 :      moveq r2, #33
591 :      cmp r5, #4
592 :      moveq r6, #5
593 :      moveq r2, #31
594 :
595 :      @ Store card left
596 :      ldr r1, =d_card_deck_count
597 :      str r2, [r1]
598 :
599 :      @ Init loop
600 :      ldr r4, =d_player_hand
601 :      ldr r0, =d_card_deck
602 :      add r0, r0, #51
603 :
604 :      @ Loop stop point
605 :      mov r1, #14
606 :      mul r5, r5, r1
607 :      add r5, r5, r4
608 :
609 :      g_dc_gc_loop1: @ Select Player
610 :          mov r3, r6      @ Card Dealt
611 :
612 :      g_dc_gc_loop2: @ Deal Cards
613 :          ldrb r1, [r0] , #-1
614 :          ldrb r2, [r4,r1]
615 :          add r2, r2, #1
616 :          strb r2, [r4,r1]
617 :          sub r3, r3, #1
618 :          cmp r3, #0
619 :          bne g_dc_gc_loop2
620 :
621 :          add r4, r4, #14
622 :          cmp r4, r5
623 :          bne g_dc_gc_loop1
624 :

```

```
625 :      @ Check for book
626 :      mov r1, #0
627 :      bl  util_check_book
628 :      mov r1, #1
629 :      bl  util_check_book
630 :      mov r1, #2
631 :      cmp r5, #3
632 :      blhs util_check_book
633 :      mov r1, #3
634 :      cmp r5, #4
635 :      blhs util_check_book
636 :
637 :      pop {r4-r10}
638 :      pop {pc}
```

Check Book

ฟังก์ชันหาไฟที่ครบหน้า จะทำการวนลูปเช็คจำนวนไฟของผู้เล่นนั้น หากมีเท่ากับ 4 ก็แสดงว่าครบ และทำการเซตค่านั้นเป็น 0 และเพิ่มค่าจำนวนไฟที่เก็บครบชุด (ซึ่งมีรูปแบบการเก็บเดียวกันกับไฟในมือผู้เล่น เพื่อง่ายต่อการจัดการและแสดงผล) ตามตำแหน่งเดียวกันกับหน้าไฟนั้น

```
69 : d_player_book:      .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0"
70 :                      .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0"
71 :                      .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0"
72 :                      .asciz  "\0\0\0\0\0\0\0\0\0\0\0\0"
---
769 : @ Check for book (return Book Found = r0; player = r1) <<<<<<<<<
770 : util_check_book:
771 :     push {lr}
772 :     push {r4}
773 :     mov r4, r1
774 :
775 :     @ Hand Pos
776 :     mov r0, #14
777 :     mul r1, r1, r0
778 :
779 :     mov r0, #0
780 :     ldr r2, =d_player_hand
781 :     add r2, r2, r1
782 :     mov r1, #0
783 :
784 : u_cb_loop:
785 :     ldrb r3, [r2,r1]
786 :     cmp r3,#4
787 :     bne u_cb_skip
788 :
789 :     @ Push Face to Book
790 :     push {r2-r4}
791 :     add r2, r2, #56 @ Hand + 56 = Book Pos
792 :
793 : u_cb_pf_loop:    @ Push To last
794 :     ldrb r3, [r2], #1
795 :     cmp r3,#0
796 :     bne u_cb_pf_loop
797 :
798 :     ldr r4, =d_card_face
799 :     ldrb r3, [r4,r1]
800 :     strb r3, [r2,#-1]
801 :
802 :     pop {r2-r4}
803 :
804 :     @ Set card to zero
```



```
805 :      mov r3, #0
806 :      strb r3, [r2,r1]
807 :
808 :      @ Print Book Get
---      //ข้ามการแสดงผลไปก่อน
820 :
821 :      @ Set Return to 1
822 :      mov r0, #1
823 :
824 : u_cb_skip:
825 :      add r1, r1, #1
826 :      cmp r1,#13
827 :      blo u_cb_loop
828 :
829 : u_cb_exit:
830 :      pop {r4}
831 :      pop {pc}
```

Count Card

ฟังก์ชันนับไพ่ในมือผู้เล่น โดยการวนลูปจำนวนไพ่ในมือผู้เล่น แล้วนำมาบวกกันทั้งหมด

ตำแหน่งของข้อมูลไพ่ที่เก็บไว้ของแต่ละผู้เล่นหาได้จาก ตำแหน่งของผู้เล่นคนแรก บวกด้วย ผู้เล่นที่ต้องการ * 13

```
65 : d_player_hand:      .asciz "\0\0\0\0\0\0\0\0\0\0\0\0"
66 :                      .asciz "\0\0\0\0\0\0\0\0\0\0\0\0"
67 :                      .asciz "\0\0\0\0\0\0\0\0\0\0\0\0"
68 :                      .asciz "\0\0\0\0\0\0\0\0\0\0\0\0"
---
678 : @ Count Card (return Card Count = r0; player = r1) <<<<<<<<<
679 : util_card_count:
680 :     push {lr}
681 :
682 :     @ Hand Pos
683 :     mov r0, #14
684 :     mul r1, r1, r0
685 :
686 :     ldr r2, =d_player_hand
687 :     add r2, r2, r1
688 :     add r1, r2, #13
689 :     mov r0, #0
690 :
691 : u_cc_loop:
692 :     ldrb r3, [r2], #1
693 :     add r0, r0, r3
694 :     cmp r2, r1
695 :     blo u_cc_loop
696 :
697 :     pop {pc}
```

Draw Card

ฟังก์ชันจั่วไพ่ โดยจะคืนค่าไพ่ที่จั่วได้ โดยจะทำการเช็คจำนวนไพในกอง หากเป็น 0 (ตำแหน่งไพ่เป็น -1) จะทำการคืนค่า -1 แต่ถ้าไม่จะไปดูไพในกองที่ตำแหน่งไพ่นั้น แล้วคืนค่าหน้านั้น ลดตำแหน่งไพในกองลง 1 แล้วทำการเพิ่มจำนวนไพของผู้เล่นตามหน้าไพ่

```
737 : @ Draw Card (return Card drawn = r0 (-1 = Out of card); player = r1)
738 : util_card_draw:
739 :     push {lr}
740 :     @ Hand Pos
741 :     ldr r3, =d_player_hand
742 :     mov r0, #14
743 :     mul r1, r1, r0
744 :     add r3, r3, r1
745 :
746 :     @ Deck count
747 :     ldr r1, =d_card_deck_count
748 :     ldr r2, [r1]
749 :
750 :     @ Check if no card to draw
751 :     cmp r2, #-1
752 :     moveq r0, #-1
753 :     beq u_cd_exit
754 :
755 :     @ Remove 1 Card from deck count
756 :     sub r0, r2, #1
757 :     str r0, [r1]
758 :
759 :     @ Put card to player hand
760 :     ldr r1, =d_card_deck
761 :     ldrb r0, [r1,r2]
762 :     ldrb r2, [r3,r0]
763 :     add r2, r2, #1
764 :     strb r2, [r3,r0]
765 :
766 : u_cd_exit:
767 :     pop {pc}
```

Winner

ฟังก์ชันหาผู้ชนะ โดยการหาจำนวนชุดที่มากที่สุดจากผู้เล่นทั้งหมด และนำมาเทียบกับจำนวนชุด
ของผู้เล่นทั้งหมดเพื่อหาจำนวนผู้ชนะและผู้ชนะ และนำผู้ชนะไปแสดงผลต่อไป

```
833 : @ Count Book (return Book Count = r0; player = r1) <<<<<<<<<
834 : util_book_count:
835 :     push {lr}
836 :
837 :     @ Hand Pos
838 :     mov r0, #14
839 :     mul r1, r1, r0
840 :
841 :     ldr r2, =d_player_book
842 :     add r2, r2, r1
843 :     add r1, r2, #13
844 :     mov r0, #0
845 :
846 : u_bc_loop:
847 :     ldrb r3, [r2], #1
848 :     cmp r3, #0
849 :     addne r0, r0, #1
850 :     cmp r2, r1
851 :     blo u_bc_loop
852 :
853 :     pop {pc}
854 :
855 : @ Find and Print Winner <<<<<<<<<
856 : util_print_winner:
857 :     push {lr}
858 :     push {r4-r7}
859 :
860 :     @ Count Book from all player (= r4-r7)
861 :     mov r1, #0
862 :     bl util_book_count
863 :     mov r4, r0
864 :
865 :     mov r1, #1
866 :     bl util_book_count
867 :     mov r5, r0
868 :
869 :     mov r1, #2
870 :     bl util_book_count
871 :     mov r6, r0
872 :
873 :     mov r1, #3
874 :     bl util_book_count
875 :     mov r7, r0
```

```

876 :
877 :     @ Count player
878 :     ldr r0, =d_player_count
879 :     ldr r0, [r0]
880 :     cmp r0,#3
881 :     blo u_pw_2p
882 :
883 :     cmp r0,#4
884 :     blo u_pw_3p
885 :
886 :     @ 4 Player
887 :
888 :     @ Find max book
889 :     mov r0, r4
890 :     cmp r0, r5
891 :     movlo r0, r5
892 :     cmp r0, r6
893 :     movlo r0, r6
894 :     cmp r0, r7
895 :     movlo r0, r7
896 :
897 :     @ Cmp max book with all player
898 :     cmp r0, r4
899 :     moveq r4, #-1
900 :     cmp r0, r5
901 :     moveq r5, #-1
902 :     cmp r0, r6
903 :     moveq r6, #-1
904 :     cmp r0, r7
905 :     moveq r7, #-1
906 :
907 :     @ Count winner
908 :     mov r0, #0
909 :     cmp r4, #-1
910 :     addeq r0, r0, #1
911 :     cmp r5, #-1
912 :     addeq r0, r0, #1
913 :     cmp r6, #-1
914 :     addeq r0, r0, #1
915 :     cmp r7, #-1
916 :     addeq r0, r0, #1
917 :
918 :     @ b Winner count
919 :     cmp r0, #4
920 :     beq u_pw_draw
921 :     cmp r0, #3
922 :     beq u_pw_4p_3w
923 :     cmp r0, #2
924 :     beq u_pw_4p_2w
925 :

```

```

926 :      @ 1 winner
927 :      cmp r4, #-1
928 :      moveq r1, #0
929 :      beq u_pw_win_1p
930 :      cmp r5, #-1
931 :      moveq r1, #1
932 :      beq u_pw_win_1p
933 :      cmp r6, #-1
934 :      moveq r1, #2
935 :      beq u_pw_win_1p
936 :      mov r1, #3
937 :      b    u_pw_win_1p
938 :
939 :      @ 2 winners
940 : u_pw_4p_2w:
941 :      cmp r4, #-1
942 :      moveq r1, #0
943 :      beq u_pw_4p_2w_p1w
944 :      cmp r5, #-1
945 :      moveq r1, #1
946 :      beq u_pw_4p_2w_p2w
947 :
948 :      @ Player 1&2 lose
949 :      mov r1, #2
950 :      mov r2, #3
951 :      b    u_pw_win_2p
952 :
953 :      @ Player 1 wins
954 : u_pw_4p_2w_p1w:
955 :      cmp r5, #-1
956 :      moveq r2, #1
957 :      beq u_pw_win_2p
958 :      cmp r6, #-1
959 :      moveq r2, #2
960 :      beq u_pw_win_2p
961 :      mov r2, #3
962 :      b    u_pw_win_2p
963 :
964 :      @ Player 2 wins
965 : u_pw_4p_2w_p2w:
966 :      cmp r6, #-1
967 :      moveq r2, #2
968 :      beq u_pw_win_2p
969 :      mov r2, #3
970 :      b    u_pw_win_2p
971 :

```

```

972 :      @ 3 winners
973 :  u_pw_4p_3w:
974 :      cmp r4, #-1
975 :      movne r1, #1
976 :      movne r2, #2
977 :      movne r3, #3
978 :      bne u_pw_win_3p
979 :      cmp r5, #-1
980 :      movne r1, #0
981 :      movne r2, #2
982 :      movne r3, #3
983 :      bne u_pw_win_3p
984 :      cmp r6, #-1
985 :      movne r1, #0
986 :      movne r2, #1
987 :      movne r3, #3
988 :      bne u_pw_win_3p
989 :      mov r1, #0
990 :      mov r2, #1
991 :      mov r3, #2
992 :      b    u_pw_win_3p
---      // ได้ฉายาเกินไปเลยยกตัวอย่างแค่แบบ 4 คนเท่านั้น และข้ามการแสดงผลไปก่อน
1106:  u_pw_exit:
1107:      pop {r4-r7}
1108:      pop {pc}

```

Call Face

ฟังก์ชันเรียกไพ่ของผู้เล่น โดยเช็คไพ่ของผู้ถูกเรียกตำแหน่งเดียวกับหน้าไพ่ที่เรียก หากไม่เป็น 0 จะเช็คค่าเป็น 0 แล้วนำค่านั้นไปบวกกับจำนวนไพ่ของผู้เรียก และคืนค่าจำนวนไพ่ที่ได้

```
1156: @ Call Face (return Card Get = r0; caller player = r1, called player = r2,
face = r3) <<<<<<<<<
1157: util_call_face:
1158:     push {lr}
1159:     push {r4-r5}
1160:     mov r4, r1      @ (caller player = r4, face = r5)
1161:     mov r5, r3
1162:
1163:     @ Called Player Hand Pos (r1)
1164:     mov r0, #14
1165:     mul r2, r2, r0
1166:     ldr r1, =d_player_hand
1167:     add r1, r1, r2
1168:
1169:     mov r0, #0
1170:
1171:     @ Check for Face
1172:     ldrb r2, [r1,r5]
1173:     cmp r2, #0
1174:     beq u_cf_exit
1175:
1176:     @ Remove card
1177:     strb r0, [r1,r5]
1178:     mov r0, r2
1179:
1180:     @ Caller Player Hand Pos (r1)
1181:     mov r3, #14
1182:     mul r4, r4, r3
1183:     ldr r1, =d_player_hand
1184:     add r1, r1, r4
1185:
1186:     @ Add Card
1187:     ldrb r3, [r1,r5]
1188:     add r3, r3, r0
1189:     strb r3, [r1,r5]
1190:
1191: u_cf_exit:
1192:     pop {r4-r5}
1193:     pop {pc}
```


Out of card

ฟังก์ชันเช็คไพ่ของผู้เล่น หากไพ่ของผู้เล่นหมด จะทำการจั่วไพ่ตามที่เริ่มต้น หากไพ่หมดก่อนที่จะจั่วครบ
จะคืนค่า 0

```
1195:  @ Check out of card (return Can draw = r0; player = r1) <<<<<<<<<
1196:  util_card_noleft:
1197:      push {lr}
1198:      push {r4-r6}
1199:      mov r4, r1
1200:
1201:      @ Check if player has no card
1202:      bl  util_card_count
1203:      cmp r0, #0
1204:      movne r0, #-1
1205:      bne u_cn_exit
1206:
1207:      @ Draw card
1208:      mov r5, #7
1209:      ldr r0, =d_player_count
1210:      ldr r0, [r0]
1211:      cmp r0, #3
1212:      moveq r5, #6
1213:      cmp r0, #4
1214:      moveq r5, #5
1215:      mov r6, r5
1216:
1217:  u_cn_dc_loop:
1218:      mov r1, r4
1219:      bl  util_card_draw
1220:
1221:      cmp r0, #-1
1222:      beq u_cn_dc_exit
1223:      sub r5, r5, #1
1224:      cmp r5, #0
1225:      bne u_cn_dc_loop
1226:
```

```
1227: u_cn_dc_exit:
1228:     @ Print Result
1229:     mov r3, #5
1230:     mul r3, r4, r3
1231:     ldr r1, =s_name_player1
1232:     add r1, r1, r3
1233:
1234:     cmp r5, r6
1235:     beq u_cn_pr_nocard
1236:
1237:     sub r2, r6, r5
1238:     ldr r0, =s_card_noleft
1239:     bl printf
1240:
1241:     b    u_cn_pr_cont
1242:
1243: u_cn_pr_cont:
1244:     mov r1, r4
1245:     bl util_check_book
1246:     mov r0, #1
1247:     b    u_cn_exit
1248:
1249: u_cn_pr_nocard:
1250:     ldr r0, =s_card_nodeck
1251:     bl printf
1252:     mov r0, #0
1253:
1254: u_cn_exit:
1255:     pop {r4-r6}
1256:     pop {pc}
```

Player Turn

เมื่อถึงตาของผู้เล่น จะเช็คว่ามีไพ่อยู่ในมือหรือไม่ ถ้าไม่มีแล้วไพ่หมดกองพอดีจะจบตาทันที แต่ถ้ามีไพ่ในมือจะไปขั้นตอนต่อไปซึ่งก็คือการเรียกไพ่ หากไม่ใช่ COM จะทำการแสดงผลสถานะของเกมในปัจจุบันและให้ผู้เล่นใส่หน้าไพ่ที่จะเรียก (สามารถใส่ 0 เพื่อสุมหน้าไพ่ได้) หากเป็น COM จะทำการสุมหน้าไพ่ที่จะเรียก ต่อไปเป็นการเลือกผู้เล่นที่จะเรียกไพ่ (หากเล่น 2 คนจะเป็นผู้เล่นอีกคนโดยอัตโนมัติ) ซึ่งเหมือนกับการเลือกหน้าไพ่ หากไม่ใช่ COM จะให้ผู้เล่นใส่ผู้เล่นที่จะเรียก (สามารถใส่ 0 เพื่อสุมผู้เล่นได้) หากเป็น COM จะทำการสุมผู้เล่นที่จะเรียก การใส่ข้อมูลจะมีการเช็คถูกต้องหรือไม่ หากไม่ถูกต้องจะให้ใส่ใหม่จนกว่าจะถูกต้อง หลังจากได้หน้าไพ่และผู้เล่นที่จะเรียกไพ่ได้แล้วจะทำการเรียกไพ่ หากสำเร็จจะวนกลับไปตอนเริ่มต้น แต่ถ้าไม่สำเร็จจะไปจั่วไพ่ในกอง หากเป็นไพ่หน้าเดียวกันกับที่เรียกจะวนกลับไปตอนเริ่มต้น แต่ถ้าไม่ใช่ก็เป็นอันจบตาของผู้เล่นนั้น

```
201 : @ == Game Function ==
202 : @ Player play (Player = r1) <<<<<<<<<
203 : game_com_play:
204 :     push {lr}
205 :     push {r4-r6}
206 :     mov r4, r1
207 :
208 :     @ (r4 = Player, r5 = Call Face, r6 = Call Player)
209 :
210 :     @ Start of turn
211 : g_cp_start:
212 :     @ Check for empty hand
213 :     mov r1, r4
214 :     bl util_card_noleft
215 :     cmp r0, #0
216 :     beq g_cp_exit
217 :
218 :     @ If player turn, print game info
219 :     cmp r4, #0
220 :     bleq game_print_info
221 :
222 :     @ Player or COM
223 :     cmp r4, #0
224 :     bne g_cp_rf
225 :
```

```

226 : g_cp_in_fa:
227 :     @ Input face to call
228 :     ldr r0, =s_input_card
229 :     bl printf
230 :
231 :     ldr r0, =p_2char
232 :     ldr r1, =b_scanf_w1
233 :     ldr r2, =b_scanf_w2
234 :     bl scanf
235 :
236 :     @ Input Char
237 :     ldr r1, =b_scanf_w1
238 :     ldrb r1, [r1]
239 :
240 :     @ Random?
241 :     cmp r1, #48
242 :     beq g_cp_rf
243 :
244 :     @ Check for invalid input
245 :     bl util_in_face
246 :     cmp r0, #-1
247 :     ble g_cp_in_fa
248 :     mov r5, r0
249 :
250 :     b g_cp_ps
251 :
252 :     @ Random face to call
253 : g_cp_rf:
254 :     mov r1, r4
255 :     bl util_face_count
256 :     sub r1, r0, #1
257 :     bl util_rand
258 :     mov r1, r4
259 :     mov r2, r0
260 :     bl util_card_pos
261 :     mov r5, r0
262 :
263 : g_cp_ps:
264 :     @ Player or COM
265 :     cmp r4, #0
266 :     bne g_cp_rp
267 :
268 :     @ 1v1?
269 :     ldr r0, =d_player_count
270 :     ldr r0, [r0]
271 :     cmp r0, #2
272 :     moveq r6, #1
273 :     beq g_cp_cont
274 :

```

```

275 : g_cp_in_ps:
276 :     @ Input player to call
277 :     ldr r0, =d_player_count
278 :     ldr r0, [r0]
279 :     cmp r0, #3
280 :     ldreq r0, =s_input_player3
281 :     ldrne r0, =s_input_player4
282 :     bl printf
283 :
284 :     ldr r0, =p_2char
285 :     ldr r1, =b_scanf_w1
286 :     ldr r2, =b_scanf_w2
287 :     bl scanf
288 :
289 :     @ Input Char
290 :     ldr r1, =b_scanf_w1
291 :     ldr r1, [r1]
292 :
293 :     @ Random?
294 :     cmp r1, #48
295 :     beq g_cp_rp
296 :
297 :     @ Check for invalid input
298 :     cmp r1, #48
299 :     bls g_cp_in_ps_no
300 :
301 :     ldr r0, =d_player_count
302 :     ldr r0, [r0]
303 :     add r0, #47
304 :
305 :     cmp r1, r0
306 :     bhi g_cp_in_ps_no
307 :
308 :     sub r1, #48
309 :     mov r6, r1
310 :     b g_cp_cont
311 :
312 : g_cp_in_ps_no:
313 :     ldr r0, =s_invalid_input
314 :     bl printf
315 :     b g_cp_in_ps
316 :
317 :     @ Random player to call
318 : g_cp_rp:
319 :     ldr r1, =d_player_count
320 :     ldr r1, [r1]
321 :     sub r1, r1, #2
322 :     bl util_rand
323 :     cmp r0, r4
324 :     addhs r0, r0, #1
325 :     mov r6, r0

```

```

326 :
327 : g_cp_cont:
328 :     @ Print card call
329 :     mov r0, #5
330 :     mul r2, r4, r0
331 :     ldr r1, =s_name_player1
332 :     add r1, r1, r2
333 :     mul r3, r6, r0
334 :     ldr r2, =s_name_player1
335 :     add r2, r2, r3
336 :     ldr r3, =d_card_face
337 :     ldrb r3, [r3,r5]
338 :     ldr r0, =s_card_call
339 :     bl printf
340 :
341 :     @ Call for card
342 :     mov r1, r4
343 :     mov r2, r6
344 :     mov r3, r5
345 :     bl util_call_face
346 :
347 :     @ Print result
348 :     cmp r0, #0
349 :     beq g_cp_pr_nope
350 :
351 :     @ Get card
352 :     mov r3, #5
353 :     mul r1, r6, r3
354 :     ldr r3, =s_name_player1
355 :     add r3, r3, r1
356 :     mov r1, r0
357 :     ldr r2, =d_card_face
358 :     ldrb r2, [r2,r5]
359 :     ldr r0, =s_card_get
360 :     bl printf
361 :
362 :     @ Check Book
363 :     mov r1, r4
364 :     bl util_check_book
365 :
366 :     @ Extra turn
367 :     b g_cp_start
368 :

```

```

369 :      @ Dont Get card
370 :  g_cp_pr_nope:
371 :      mov r3, #5
372 :      mul r2, r6, r3
373 :      ldr r1, =s_name_player1
374 :      add r1, r1, r2
375 :      ldr r2, =d_card_face
376 :      ldrb r2, [r2,r5]
377 :      ldr r0, =s_card_nope
378 :      bl  printf
379 :
380 :      @ Draw card
381 :      mov r1, r4
382 :      bl  util_card_draw
383 :
384 :      @ Check Catch
385 :      cmp r0, r5
386 :      bne g_cp_pr_end
387 :
388 :      mov r1, #5
389 :      mul r2, r4, r1
390 :      ldr r1, =s_name_player1
391 :      add r1, r1, r2
392 :      ldr r3, =d_card_face
393 :      ldr r2, [r3,r0]
394 :      ldr r0, =s_card_catch
395 :      bl  printf
396 :
397 :      @ Check Book
398 :      mov r1, r4
399 :      bl  util_check_book
400 :
401 :      @ Extra turn
402 :      b   g_cp_start
403 :
404 :  g_cp_pr_end:
405 :      cmp r0, #-1
406 :      bne g_cp_pr_pc
407 :
408 :      @ No card
409 :      ldr r0, =s_card_nodraw
410 :      bl  printf
411 :      b   g_cp_exit
412 :

```

```
413 :      @ Print card
414 :  g_cp_pr_pc:
415 :      mov r1, #5
416 :      mul r2, r4, r1
417 :      ldr r1, =s_name_player1
418 :      add r1, r1, r2
419 :      cmp r4, #0
420 :      ldrne r0, =s_card_draw
421 :      bne g_cp_pr_pc_p
422 :      mov r2, r0
423 :      ldr r0, =d_card_face
424 :      ldrb r2, [r0,r2]
425 :      ldr r0, =s_card_draws
426 :  g_cp_pr_pc_p:
427 :      bl printf
428 :
429 :      mov r1, r4
430 :      bl util_check_book
431 :  g_cp_exit:
432 :      pop {r4-r6}
433 :      pop {pc}
```


Game Loop

หลังจากแสดงหน้า Title จะให้ใส่จำนวนผู้เล่น (2-4) หลังจากนั้นจะทำการสับไฟแล้วแจก และสุ่มผู้เล่นที่จะเริ่มตาแรก หลังจากจบตาจะเช็คไฟที่เหลือในกอง หากยังไม่หมดจะวนลูปและเริ่มตาของผู้เล่นถัดไป แต่ถ้าหากหมดจะจบเกมและแสดงผู้ชนะ

```
95 :      @ Game =====
96 :
97 :      @ Print title
98 :      ldr r0, =s_title
99 :      bl  printf
100 :
101 :      @ Set player count
102 :      m_sp_loop:
103 :      ldr r0, =s_player_set
104 :      bl  printf
105 :
106 :      ldr r0, =p_2char
107 :      ldr r1, =b_scanf_w1
108 :      ldr r2, =b_scanf_w2
109 :      bl  scanf
110 :
111 :      @ Input Char
112 :      ldr r1, =b_scanf_w1
113 :      ldrb r1, [r1]
114 :
115 :      @ Check Input
116 :      cmp r1, #49
117 :      bls m_sp_no
118 :
119 :      cmp r1, #53
120 :      bhs m_sp_no
121 :
122 :      @ Set Value
123 :      sub r1, #48
124 :      ldr r0, =d_player_count
125 :      str r1, [r0]
126 :      b   m_sg
127 :      m_sp_no:
128 :      ldr r0, =s_invalid_input
129 :      bl  printf
130 :      b   m_sp_loop
```

```

131 : m_sg:
132 :     @ Start Game
133 :     ldr r0, =s_game_start
134 :     bl  printf
135 :
136 :     bl  game_deck_suff
137 :     bl  game_deal_card
138 :     bl  game_print_info
139 :
140 :     @ Random Start Player
141 :     ldr r10, =d_player_count @ r10 = Player Count
142 :     ldr r10, [r10]
143 :     sub r1, r10, #1
144 :     bl  util_rand
145 :     mov r4, r0 @ r4 = Player Turn
146 :
147 :     @ Print who goes first
148 :     ldr r0, =s_turn_start
149 :     mov r1, #5
150 :     mul r2, r4, r1
151 :     ldr r1, =s_name_player1
152 :     add r1, r1, r2
153 :     bl  printf
154 :
155 :     @ Main Game Loop
156 : gl_loop:
157 :     @ Game end if no card in deck
158 :     ldr r0, =d_card_deck_count
159 :     ldr r0, [r0]
160 :     cmp r0, #-1
161 :     ble gl_end
162 :
163 :     @ Print Turn
164 :     ldr r0, =s_game_turn
165 :     mov r1, #5
166 :     mul r2, r4, r1
167 :     ldr r1, =s_name_player1
168 :     add r1, r1, r2
169 :     bl  printf
170 :
171 :     @ Play turn
172 :     mov r1, r4
173 :     bl  game_com_play
174 :
175 :     @ Next Player Turn
176 :     add r4, r4, #1
177 :     cmp r4, r10
178 :     moveq r4, #0
179 :
180 :     b   gl_loop
181 :

```

```
182 : gl_end:
183 :     @ Game Over
184 :     bl game_print_info
185 :
186 :     ldr r0, =s_game_over1
187 :     bl printf
188 :
189 :     bl util_print_winner
190 :
191 :     ldr r0, =s_game_over3
192 :     bl printf
193 :
194 :     @ Game =====
```

Display

Title Screen



ทำหน้าที่ Title เป็นตัวอักษรเก็บเป็น .asciz ใน .data แล้วใช้ printf แสดงออกมา

[illegible]

Player Name & Card Face

```
YOU call COM1 for As.  
Gets 3 As from COM1.  
YOU Gets a book of four As.
```

ชื่อผู้เล่นเก็บใน .asciz ความยาว 4 ตัวอักษร + Null Character เวลาแสดงชื่อผู้เล่นก็นำตำแหน่ง

ชื่อผู้เล่นคนแรก + ตำแหน่งผู้เล่นที่จะแสดง (เริ่มจาก 0) * 5

การแสดงผลหน้าไพ่ นำหน้าไพ่ที่จะแสดงที่เป็นตัวเลข 0-12 มาโหลดตำแหน่งกับ .asciz ที่เทียบตำแหน่งกับหน้าไพ่ที่ตรงกัน

```
16 : s_name_player1: .asciz "YOU "  
17 : s_name_player2: .asciz "COM1"  
18 : s_name_player3: .asciz "COM2"  
19 : s_name_player4: .asciz "COM3"  
---  
37 : s_card_call: .asciz "\n%s call %s for %cs."  
38 : s_card_get: .asciz "\nGets %d %cs from %s."  
---  
47 : s_book_get: .asciz "\n%s Gets a book of four %cs."  
---  
61 : d_card_face: .asciz "23456789TJQKA"  
---  
328 : @ Print card call  
329 : mov r0, #5  
330 : mul r2, r4, r0  
331 : ldr r1, =s_name_player1  
332 : add r1, r1, r2  
333 : mul r3, r6, r0  
334 : ldr r2, =s_name_player1  
335 : add r2, r2, r3  
336 : ldr r3, =d_card_face  
337 : ldrb r3, [r3,r5]  
338 : ldr r0, =s_card_call  
339 : bl printf  
---  
351 : @ Get card  
352 : mov r3, #5  
353 : mul r1, r6, r3  
354 : ldr r3, =s_name_player1  
355 : add r3, r3, r1  
356 : mov r1, r0  
357 : ldr r2, =d_card_face  
358 : ldrb r2, [r2,r5]  
359 : ldr r0, =s_card_get  
360 : bl printf  
---
```

```
808 :      @ Print Book Get
809 :      push {r0-r3}
810 :      mov r0, r1
811 :      mov r1, #5
812 :      mul r2, r4, r1
813 :      ldr r1, =s_name_player1
814 :      add r1, r1, r2
815 :      ldr r3, =d_card_face
816 :      ldr r2, [r3,r0]
817 :      ldr r0, =s_book_get
818 :      bl  printf
819 :      pop {r0-r3}
```

Game Info

```
-----Game Info-----
Card(s) left in pool: 2
      Book(s)      Hand
YOU   J            222333558TT
COM1                      5 Card(s)
COM2 4K            8 Card(s)
COM3 Q9           6 Card(s)
```

ฟังก์ชันแสดงสถานะของเกม จะแสดงจำนวนชุดที่ได้และไพ่ของผู้เล่น หากไม่ใช่ COM จะแสดงไพ่ที่ถืออยู่แทน และจำนวนไพ่ที่เหลืออยู่ในกอง

```
21 : s_game_info1:      .asciz  "\n-----Game Info-----\nCard(s) left in pool: %d\n      Book(s)      Hand\n%s %-15s "
22 : s_game_info2:      .asciz  "\n%s %-15s %-2d Card(s)"
---
435 : @ Print Game Info <<<<<<<<<
436 : game_print_info:
437 :     push {lr}
438 :
439 :     @ Debug Print Deck
440 :     @bl  debug_print_deck
441 :     @ldr  r0, =p_newline
442 :     @bl  printf
443 :
444 :     @ Print Header and Player info
445 :     ldr r0, =s_game_info1
446 :     ldr r1, =d_card_deck_count
447 :     ldr r1, [r1]
448 :     add r1, r1, #1
449 :     ldr r2, =s_name_player1
450 :     ldr r3, =d_player_book
451 :     bl  printf
452 :
453 :     mov r1, #0
454 :     bl util_card_print
455 :
456 :     @ COM1
457 :     mov r1, #1
458 :     bl  util_card_count
459 :     mov r3, r0
460 :     ldr r0, =s_game_info2
461 :     ldr r1, =s_name_player2
462 :     ldr r2, =d_player_book+14
463 :     bl  printf
464 :
```

```

465 :      @ Debug Print Card
466 :      @mov r1, #1
467 :      @bl util_card_print
468 :
469 :      @ COM2
470 :      ldr r0, =d_player_count
471 :      ldr r0, [r0]
472 :      cmp r0,#3
473 :      blo g_pi_skip
474 :
475 :      mov r1, #2
476 :      bl  util_card_count
477 :      mov r3, r0
478 :      ldr r0, =s_game_info2
479 :      ldr r1, =s_name_player3
480 :      ldr r2, =d_player_book+28
481 :      bl  printf
482 :
483 :      @ Debug Print Card
484 :      @mov r1, #2
485 :      @bl util_card_print
486 :
487 :      @ COM3
488 :      ldr r0, =d_player_count
489 :      ldr r0, [r0]
490 :      cmp r0,#4
491 :      blo g_pi_skip
492 :
493 :      mov r1, #3
494 :      bl  util_card_count
495 :      mov r3, r0
496 :      ldr r0, =s_game_info2
497 :      ldr r1, =s_name_player4
498 :      ldr r2, =d_player_book+42
499 :      bl  printf
500 :
501 :      @ Debug Print Card
502 :      @mov r1, #3
503 :      @bl util_card_print
504 :
505 : g_pi_skip:
506 :     pop {pc}
---
699 : @ Print Card (player = r1) <<<<<<<<<
700 : util_card_print:
701 :     push {lr}
702 :     push {r4}
703 :
704 :     @ Hand Pos
705 :     mov r0, #14
706 :     mul r1, r1, r0

```



```

707 :
708 :     ldr r2, =d_player_hand
709 :     add r2, r2, r1
710 :     mov r1, #0
711 :     ldr r4, =d_card_face
712 :
713 : u_cp_loop1: @ Loop all card
714 :     ldrb r3, [r2,r1]
715 :
716 : u_cp_loop2: @ Print card face
717 :     cmp r3, #0
718 :     beq u_cp_skip
719 :     sub r3, r3, #1
720 :
721 :     @ Printf
722 :     push {r0-r3}
723 :     ldr r0, =p_char
724 :     ldrb r1, [r4,r1]    @ Card Face Num to Char
725 :     bl printf
726 :     pop {r0-r3}
727 :     b    u_cp_loop2
728 :
729 : u_cp_skip:
730 :     add r1, r1, #1
731 :     cmp r1, #13
732 :     blo u_cp_loop1
733 :
734 :     pop {r4}
735 :     pop {pc}

```

Result

[illegible]

```
pi@raspberrypi: ~/Documents
File Edit Tabs Help
YOU has no 9s. Go fish!
COM1 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 36
  Book(s)      Hand
YOU           225TJJA
COM1          9 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for As.
COM1 has no As. Go fish!
YOU Fishes Q from the pool.

|== COM1 turn ==|
COM1 call YOU for 8s.
YOU has no 8s. Go fish!
COM1 Fishes a card from the pool.
COM1 Gets a book of four 9s.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 34
  Book(s)      Hand
YOU           225TJJQA
COM1 9         6 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for Js.
COM1 has no Js. Go fish!
YOU Fishes K from the pool.

|== COM1 turn ==|
COM1 call YOU for 4s.
YOU has no 4s. Go fish!
COM1 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 32
  Book(s)      Hand
YOU           225TJJQKA
COM1 9         7 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for Ks.
COM1 has no Ks. Go fish!
YOU Fishes A from the pool.

|== COM1 turn ==|
COM1 call YOU for 6s.
YOU has no 6s. Go fish!
COM1 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 30
  Book(s)      Hand
YOU           225TJJQKAA
COM1 9         8 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for As.
Gets 1 As from COM1.
-----Game Info-----
```

```
pi@raspberrypi: ~/Documents
File Edit Tabs Help
|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 4
      Book(s)      Hand
YOU  AJ5K          6667TTT
COM1 943           13 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for 6s.
Gets 1 6s from COM1.
YOU Gets a book of four 6s.
-----Game Info-----
Card(s) left in pool: 4
      Book(s)      Hand
YOU  AJ5K6         7TTT
COM1 943           12 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for Ts.
COM1 has no Ts. Go fish!
YOU Fishes 2 from the pool.

|== COM1 turn ==|
COM1 call YOU for Qs.
YOU has no Qs. Go fish!
COM1 Fishes a card from the pool.
COM1 Gets a book of four 8s.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 2
      Book(s)      Hand
YOU  AJ5K6         27TTT
COM1 9438          9 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

YOU call COM1 for Ts.
COM1 has no Ts. Go fish!
YOU Fishes Q from the pool.

|== COM1 turn ==|
COM1 call YOU for 7s.
Gets 1 7s from YOU .
COM1 Gets a book of four 7s.
COM1 call YOU for Qs.
Gets 1 Qs from YOU .
COM1 Gets a book of four Qs.
COM1 call YOU for 2s.
Gets 1 2s from YOU .
COM1 Gets a book of four 2s.
COM1 has no cards left. Draws 1 card(s) from the pool.
COM1 call YOU for Ts.
Gets 3 Ts from YOU .
COM1 Gets a book of four Ts.
COM1 has no cards left. No cards left in the pool to draw!
-----Game Info-----
Card(s) left in pool: 0
      Book(s)      Hand
YOU  AJ5K6
COM1 94387Q2T      0 Card(s)

No card in pool. GAME OVER!
=====
COM1 wins!
=====
pi@raspberrypi:~/Documents $
```

```
pi@raspberrypi: ~/Documents
File Edit Tabs Help
\___|\___/ | | | |___/ | | |
Developed by Natchapon Santiphiboon 60010285
This Project is for Computer Organization and Assembly Language Subject at KMITL

Set amount of player (2-4) : 3

Starting Game...
-----Game Info-----
Card(s) left in pool: 34
  Book(s)      Hand
YOU           334JQK
COM1          6 Card(s)
COM2          6 Card(s)

Randomize starting player...
>COM1 goes first!

|== COM1 turn ==|
COM1 call COM2 for Ts.
COM2 has no Ts. Go fish!
COM1 Fishes a card from the pool.

|== COM2 turn ==|
COM2 call YOU for Ks.
Gets 1 Ks from YOU .
COM2 call COM1 for 9s.
COM1 has no 9s. Go fish!
COM2 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 32
  Book(s)      Hand
YOU           334JQ
COM1          7 Card(s)
COM2          8 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 3

Select player to call (1,2) (0 for random) : 3

Invalid Input!
Select player to call (1,2) (0 for random) : 2

YOU call COM2 for 3s.
COM2 has no 3s. Go fish!
YOU Fishes 2 from the pool.

|== COM1 turn ==|
COM1 call YOU for Ts.
YOU has no Ts. Go fish!
COM1 Fishes a card from the pool.

|== COM2 turn ==|
COM2 call YOU for 9s.
YOU has no 9s. Go fish!
COM2 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 29
  Book(s)      Hand
YOU           2334JQ
COM1          8 Card(s)
COM2          9 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : █

documents pi pi@raspberrypi: ~/Do...
```

```
pi@raspberrypi: ~/Documents
File Edit Tabs Help

Set amount of player (2-4) : 4

Starting Game...
-----Game Info-----
Card(s) left in pool: 32
  Book(s)      Hand
YOU           23579
COM1          5 Card(s)
COM2          5 Card(s)
COM3          5 Card(s)

Randomize starting player...
>COM2 goes first!

|== COM2 turn ==|
COM2 call COM3 for Qs.
COM3 has no Qs. Go fish!
COM2 Fishes a card from the pool.

|== COM3 turn ==|
COM3 call COM2 for As.
COM2 has no As. Go fish!
COM3 Fishes a card from the pool.

|== YOU turn ==|
-----Game Info-----
Card(s) left in pool: 30
  Book(s)      Hand
YOU           23579
COM1          5 Card(s)
COM2          6 Card(s)
COM3          6 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 9

Select player to call (1,2,3) (0 for random) : 4

Invalid Input!
Select player to call (1,2,3) (0 for random) : 3

YOU call COM3 for 9s.
Gets 1 9s from COM3.
-----Game Info-----
Card(s) left in pool: 30
  Book(s)      Hand
YOU           235799
COM1          5 Card(s)
COM2          6 Card(s)
COM3          5 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

Select player to call (1,2,3) (0 for random) : 1

YOU call COM1 for 7s.
COM1 has no 7s. Go fish!
YOU Fishes 3 from the pool.

|== COM1 turn ==|
COM1 call YOU for Ks.
YOU has no Ks. Go fish!
COM1 Fishes a card from the pool.

|== COM2 turn ==|
COM2 call COM1 for Qs.
COM1 has no Qs. Go fish!
COM2 Fishes a card from the pool.
```

```
pi@raspberrypi: ~/Documents
File Edit Tabs Help
YOU T 46699
COM1 8 10 Card(s)
COM2 9 Card(s)
COM3 3QJK 2 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

Select player to call (1,2,3) (0 for random) : 0

YOU call COM3 for 9s.
Gets 1 9s from COM3.
-----Game Info-----
Card(s) left in pool: 2
Book(s) Hand
YOU T 466999
COM1 8 10 Card(s)
COM2 9 Card(s)
COM3 3QJK 1 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

Select player to call (1,2,3) (0 for random) : 0

YOU call COM3 for 9s.
COM3 has no 9s. Go fish!
YOU Catches 9 from the pool. Lucky!
YOU Gets a book of four 9s.
-----Game Info-----
Card(s) left in pool: 1
Book(s) Hand
YOU T9 466
COM1 8 10 Card(s)
COM2 9 Card(s)
COM3 3QJK 1 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

Select player to call (1,2,3) (0 for random) : 0

YOU call COM1 for 6s.
COM1 has no 6s. Go fish!
YOU Catches 6 from the pool. Lucky!
-----Game Info-----
Card(s) left in pool: 0
Book(s) Hand
YOU T9 4666
COM1 8 10 Card(s)
COM2 9 Card(s)
COM3 3QJK 1 Card(s)
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 0

Select player to call (1,2,3) (0 for random) : 0

YOU call COM2 for 4s.
COM2 has no 4s. Go fish!
No cards left in the pool to draw!
-----Game Info-----
Card(s) left in pool: 0
Book(s) Hand
YOU T9 4666
COM1 8 10 Card(s)
COM2 9 Card(s)
COM3 3QJK 1 Card(s)

No card in pool. GAME OVER!
=====
COM3 wins!
=====
pi@raspberrypi:~/Documents $
```

```

|== COM1 turn ==|
COM1 call COM3 for 5s.
COM3 has no 5s. Go fish!
COM1 Fishes a card from the pool.
-----Game Info-----
Card(s) left in pool: 0
      Book(s)      Hand
YOU   J           222333558TTT
COM1           6 Card(s)
COM2 4K          8 Card(s)
COM3 Q9          6 Card(s)

No card in pool. GAME OVER!
=====
          COM2 and COM3 win!
=====
pi@raspberrypi:~/Documents $

```

```

|== COM2 turn ==|
COM2 call COM1 for 6s.
COM1 has no 6s. Go fish!
COM2 Fishes a card from the pool.
-----Game Info-----
Card(s) left in pool: 0
      Book(s)      Hand
YOU   73          666KAA
COM1 J295         10 Card(s)
COM2 8            8 Card(s)

No card in pool. GAME OVER!
=====
          COM1 wins!
=====
pi@raspberrypi:~/Documents $

```

```

YOU call COM1 for As.
Gets 3 As from COM1.
YOU Gets a book of four As.
YOU has no cards left. No cards left in the pool to draw!
-----Game Info-----
Card(s) left in pool: 0
      Book(s)      Hand
YOU   K367824A
COM1 9TJQ5         0 Card(s)

No card in pool. GAME OVER!
=====
          YOU wins!
=====
pi@raspberrypi:~/Documents $

```

```

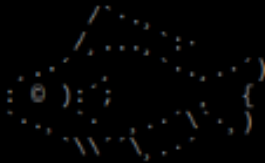
|== COM3 turn ==|
COM3 call COM2 for Ts.
COM2 has no Ts. Go fish!
COM3 Fishes a card from the pool.
-----Game Info-----
Card(s) left in pool: 0
      Book(s)      Hand
YOU           2788TTJA
COM1           11 Card(s)
COM2           14 Card(s)
COM3           19 Card(s)

No card in pool. GAME OVER!
=====
          Nobody wins :(
=====
pi@raspberrypi:~/Documents $ 0

```



```
pi@raspberrypi:~/Documents $ ./Assignment2
```



```
Developed by Natchapon Santiphiboon 60010285  
This Project is for Computer Organization and Assembly Language Subject at KMITL
```

```
Set amount of player (2-4) : 1
```

```
Invalid Input!
```

```
Set amount of player (2-4) : 5
```

```
Invalid Input!
```

```
Set amount of player (2-4) : █
```

```
|== YOU turn ==|
```

```
-----Game Info-----
```

```
Card(s) left in pool: 32
```

```
Book(s) Hand
```

```
YOU 334JQ
```

```
COM1 7 Card(s)
```

```
COM2 8 Card(s)
```

```
Select face to call (2,3,4,5,6,7,8,9,T,J,Q,K,A) (0 for random) : 3
```

```
Select player to call (1,2) (0 for random) : 3
```

```
Invalid Input!
```

```
Select player to call (1,2) (0 for random) : 2
```