```
// PS: Open 7-Segment Display as 8 digit mode in CPUlator.



.global _start
_start:
.equ TIMER, 0xFFFEC600
.equ INTERRUPT, 0xFFFEC60C
.equ ADDR_7SEG1, 0xFF200020
.equ SWITCHES, 0xFF200040
.equ BUTTON, 0xFF20005C

Ldr R7, =500000000
Ldr R6, =TIMER
Ldr R8, =INTERRUPT
LDR R0, =BUTTON
ldr r1, =HEXTABLE // ACE, 2, 3, 4, 5, 6, 7, 8, 9, VALE, GIRL , PRINCE
ldr r2, =ADDR_7SEG1
ldr r3, =SWITCHES
LDR R4, =PLAYER //cards, cardsHex, aces, money, bet, condition
LDR R5, =DEALER //cards, cardsHex, aces



// Set Timer
str R7, [R6]
MOV r7, #1
str R7, [R8]
MOV r7, #3
str R7, [R6, #8]
LDR R7, =CARDS
LDR R11, =ORIGINALHEX
MOV R9, #15
```

```
STR R9, [R0]

LDR R8, =0x6D6D6D6D
STR R8, [R2]
STR R8, [R2, #16]

Money:
LDR R9, [R0]
CMP R9, #2
MOVNE R9, #13
STRNE R9, [R0]
BNE Money
MOV R9, #2
STR R9, [R0]
LDR R9, [R3]
STR R9, [R4, #12]



//Start the game
First_phase:
LDR R8, =0xFFF90701
STR R8, [R2]
MOV R8, #0
STR R8, [R2, #16]
LDR R8, [R6, #4] // Generating random numbers
LDR R9, [R0] // switch controls
CMP R9, #1
MOVNE R9, #14
STRNE R9, [R0]
BNE First_phase
```

```
MOV R9, #1
STR R9, [R0]
STR R9, [R0]

LDR R9, [R3] // Bet received from switches is hidden
LDR R12, [R4, #12]
CMP R12, R9 // Bet can not be greater than money" condition check
BLT First_phase
STR R9, [R4, #16]

MOV R9, #0 // Ace and card counters reset
STR R9, [R4, #8]
STR R9, [R5, #8]
STR R9, [R7]
STR R9, [R7, #4]
STR R9, [R7, #8]
STR R9, [R7, #12]
STR R9, [R7, #16]
STR R9, [R7, #20]
STR R9, [R7, #24]
STR R9, [R7, #28]
STR R9, [R7, #32]
STR R9, [R7, #36]
STR R9, [R7, #40]
STR R9, [R7, #44]
STR R9, [R7, #48]
```

```
MOV R9, R8
AND R8, #15
CMP R8, #12
SUBGT R8, R8, #12 // Generate first card

LDR R10, [R7, R8, lsl#2] // The card used is marked
ADD R10, #1
STR R10, [R7, R8, lsl#2]

LDR R10, [R1, R8, lsl#2] // The hexadecimal representation of the card has been retained
STR R10, [R4, #4]

CMP R8, #9 // 10, valet, queen and king status observed
MOVGE R10, #10 // 10 added to card value counter if status is valid
STRGE R10, [R4]

CMP R8, #0 // Ace status of the card has been observed
MOVEQ R10, #11 // added 11 to card value counter if status is valid
STREQ R10, [R4]
MOVEQ R10, #1 // Added 1 to the ace counter because the card is an ace
STREQ R10, [R4, #8]

CMPNE R8, #9 //Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
STRLT R8, [R4]



// Player's 2nd card determined
LSR R9, R9, #4
MOV R8, R9
```

```
AND R8, #15
CMP R8, #12
SUBGT R8, R8, #12
// The card used is marked
LDR R10, [R7, R8, lsl#2]
ADD R10, #1
STR R10, [R7, R8, lsl#2]

SUBGT R8, R8, #12
LDR R10, [R1, R8, lsl#2] // Hexadecimal representation of the card retained
LDR R12, [R4, #4]
LSL R12, #8
ADD R10, R10, R12
STR R10, [R4, #4]

CMP R8, #9 // 10, valet, queen and king status observed
LDRGE R10, [R4] // 10 added to card value counter if status is valid
ADDGE R10, #10
STRGE R10, [R4]

CMP R8, #0 // The ace status of the card is observed
LDREQ R10, [R4] // If the status is valid, 11 is added to the card value counter
ADDEQ R10, #11
STREQ R10, [R4]
LDREQ R10, [R4, #8]  // Added 1 to the ace counter because the card is an ace
ADDEQ R10, #1
STREQ R10, [R4, #8]

CMPNE R8, #9 // Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
LDRLT R10, [R4]
```

```
ADDLT R8, R8, R10
STRLT R8, [R4]



// The dealer's first card is determined
LSR R9, R9, #4
MOV R8, R9
AND R8, #15
CMP R8, #12
SUBGT R8, R8, #12
// The card used is marked
LDR R10, [R7, R8, lsl#2]
ADD R10, #1
STR R10, [R7, R8, lsl#2]

LDR R10, [R1, R8, lsl#2] // The hexadecimal representation of the card is preserved
STR R10, [R5, #4]

CMP R8, #9 // 10, jack, queen and king status observed
MOVGE R10, #10 // If the status is valid, 10 is added to the card value counter
STRGE R10, [R5]

CMP R8, #0
MOVEQ R10, #11 // The ace status of the card is observed
STREQ R10, [R5]
MOVEQ R10, #1 // Added 1 to the ace counter because the card is an ace
STREQ R10, [R5, #8]

CMPNE R8, #9 // Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
```

```
STRLT R8, [R5]

// The dealer's 2nd card is determined
LSR R9, R9, #4
MOV R8, R9
AND R8, #15
CMP R8, #12
SUBGT R8, R8, #12

// The card used is marked
LDR R10, [R7, R8, lsl#2]
ADD R10, #1
STR R10, [R7, R8, lsl#2]

LDR R10, [R1, R8, lsl#2] // The hexadecimal representation of the card is preserved
LDR R12, [R5, #4]
LSL R12, #8
ADD R10, R10, R12
STR R10, [R5, #4]

CMP R8, #9 // 10, jack, queen and king status observed
LDRGE R10, [R5] // If the status is valid, 10 is added to the card value counter
ADDGE R10, #10
STRGE R10, [R5]

CMP R8, #0 // Ace status of the card has been observed
LDREQ R10, [R5] // Added 11 to card value counter if status is valid
ADDEQ R10, #11
STREQ R10, [R5]
```

```
LDREQ R10, [R5, #8]  // Added 1 to the ace counter because the card is an ace
ADDEQ R10, #1
STREQ R10, [R5, #8]

CMPNE R8, #9 // Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
LDRLT R10, [R5]
ADDLT R8, R8, R10
STRLT R8, [R5]



// Double ace draw case CROWER
LDR R8, [R5]
CMP R8, #22
MOVEQ R8, #12
STREQ R8, [R5]



// Double ace draw PLAYER
LDR R8, [R4]
CMP R8, #22
MOVEQ R8, #12
STREQ R8, [R4]
// Check (Does the result pass 21?)
CMP R8, #21
BEQ End_The_Game

//Player card screen came up
LDR R10, [R4, #4]
STR R10, [R2]
LDR R10, =0xf3380000
```

```
STR R10, [R2, #16]

Button:
LDR R8, [R6, #4]
LDR R9, [R0]
CMP R9, #1
MOVEQ R9, #1
STREQ R9, [R0]
BEQ Change_Display
CMP R9, #2
MOVEQ R9, #2
STREQ R9, [R0]
BEQ Draw_a_Card
CMP R9, #4
MOVEQ R9, #4
STREQ R9, [R0]
BEQ End_The_Game
MOVNE R9, #8
STRNE R9, [R0]
BNE Button

Change_Display:
LDR R8, [R2, #16]
LSR R8, R8, #28
CMP R8, #0
LDREQ R9, =0xf3380000 // Pl Display
STREQ R9, [R2, #16]
```

```
LDREQ R9, [R4, #4]
STREQ R9, [R2]
BEQ Button
CMP R8, #3
LDREQ R9, =0x07013800 // Tl Display
STREQ R9, [R2, #16]
MOVEQ R8, #0
LDREQ R9, [R4, #12]
BEQ Money_Displayer
LDR R9, =0x3ff9f738 // Deal Display
STR R9, [R2, #16]
LDR R9, [R5, #4]
LSR R9, R9, #8
STR R9, [R2]
B Button

Money_Displayer:
CMP R9, #10
SUBGE R9, R9, #10
ADDGE R8, R8, #1
BGE Money_Displayer
LDR R10, [R11, R9, lsl#2]
STR R10, [R2]
CMP R8, #0
BEQ Button
CMP R8, #10
MOVGE R9, #0
BGE Money_Displayer_2
LDRLT R9, [R11, R8, lsl#2]
LSLLT R9, R9, #8
ADDLT R9, R9, R10
```

```
STRLT R9, [R2]

BLT Button


Money_Displayer_2:

SUB R8, R8, #10

ADD R9, R9, #1

CMP R8, #10

BGE Money_Displayer_2

LDR R10, [R11, R8, lsl#2]

LDR R12, [R2]

LSL R10, R10, #8

ADD R12, R12, R10

CMP R9, #10

LDREQ R8, =0x63F0000

ADDEQ R12, R12, R8

STREQ R12, [R2]

BEQ Button

LDR R10, [R11, R9, lsl#2]

LSL R10, R10, #16

ADD R12, R12, R10

STR R12, [R2]

B Button



Draw_a_Card:

// Player's new card determined

LDR R9, [R6, #4]

MOV R8, R9

AND R8, #15

CMP R8, #12

SUBGT R8, R8, #12
```

```
LDR R10, [R7, R8, lsl#2] // The card used is marked
CMP R10, #4 // 5th card blocked
LSREQ R8, R8, #4
BEQ Draw_a_Card
ADD R10, #1
STR R10, [R7, R8, lsl#2]

LDR R10, [R1, R8, lsl#2] // The hexadecimal representation of the card is preserved
LDR R12, [R4, #4]
LSL R12, #8
ADD R10, R10, R12
STR R10, [R4, #4]

CMP R8, #9 // 10, jack, queen and king status observed
LDRGE R10, [R4] // If the status is valid, 10 is added to the card value counter
ADDGE R10, #10
STRGE R10, [R4]

CMP R8, #0 // The ace status of the card is observed
LDREQ R10, [R4] // If the status is valid, 11 is added to the card value counter
ADDEQ R10, #11
STREQ R10, [R4]
LDREQ R10, [R4, #8]  // Added 1 to the ace counter because the card is an ace
ADDEQ R10, #1
STREQ R10, [R4, #8]

CMPNE R8, #9 // Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
LDRLT R10, [R4]
ADDLT R8, R8, R10
```

```
STRLT R8, [R4]


LDR R9, =0xf3380000
STR R9, [R2, #16]
LDR R9, [R4, #4]
STR R9, [R2]


LDR R8, [R4] // Did the result exceed 21?
CMP R8, #21


BLT Button
BEQ End_The_Game

LDR R8, [R4, #8]
CMP R8, #0
BEQ Lose
SUB R8, R8, #1
STR R8, [R4, #8]
LDR R8, [R4]
SUB R8, R8, #10
STR R8, [R4]
B Button


End_The_Game:
LDR R8, [R5]
CMP R8, #17
BLT Draw_Dealer
```

```
LDR R9, [R4]
CMP R9, R8
BGT Win
BLT Lose
BEQ Draw



Draw_Dealer:
// Dealer's new card determined
LDR R9, [R6, #4]
MOV R8, R9
AND R8, #15
CMP R8, #12
SUBGT R8, R8, #12

LDR R10, [R7, R8, lsl#2] // The card used is marked
CMP R10, #4 // 5th card blocked
LSREQ R8, R8, #4
BEQ Draw_Dealer
ADD R10, #1
STR R10, [R7, R8, lsl#2]

LDR R10, [R1, R8, lsl#2] // The hexadecimal representation of the card is preserved
LDR R12, [R5, #4]
LSL R12, #8
ADD R10, R10, R12
STR R10, [R5, #4]

CMP R8, #9 // 10, jack, queen and king status observed
```

```
LDRGE R10, [R5] // If the status is valid, 10 is added to the card value counter
ADDGE R10, #10
STRGE R10, [R5]

CMP R8, #0 // The ace status of the card is observed
LDREQ R10, [R5] // If the status is valid, 11 is added to the card value counter
ADDEQ R10, #11
STREQ R10, [R5]
LDREQ R10, [R5, #8]  // Added 1 to the ace counter because the card is an ace
ADDEQ R10, #1
STREQ R10, [R5, #8]

CMPNE R8, #9 // Added valid number to card value counter for values 2-9
ADDLT R8, R8, #1
LDRLT R10, [R5]
ADDLT R8, R8, R10
STRLT R8, [R5]

LDR R8, [R5] // If the dealer goes over 21, the player wins
CMP R8, #21

LDRGT R8, [R5, #8]
CMPGT R8, #0

BEQ Win


SUBGE R8, R8, #1
STRGE R8, [R5, #8]
LDRGE R8, [R5]
SUBGE R8, R8, #10
```

```
STRGE R8, [R5]


B End_The_Game


Win:
// Edit money
LDR R8, [R4, #12]
LDR R9, [R4, #16]
ADD R8, R8, R9
STR R8, [R4, #12]


LDR R9, =0x1C0C0454 //Display
STR R9, [R2, #16]
MOV R9, #0
STR R9, [R2]


MOV R9, #0 //Condition
STR R9, [R4, #20]
B After_Game


Lose:
LDR R8, [R4, #12]
LDR R9, [R4, #16]
SUB R8, R8, R9
STR R8, [R4, #12]
```

```
LDR R9, =0x383F6DF9

STR R9, [R2, #16]

MOV R9, #0

STR R9, [R2]


MOV R9, #1

STR R9, [R4, #20]

B After_Game


Draw:

MOV R9, #2

STR R9, [R4, #20]


LDR R9, =0x40404040

STR R9, [R2, #16]

MOV R9, #0

STR R9, [R2]



B After_Game


After_Game:

LDR R9, [R0]

CMP R9, #1

MOVEQ R9, #1

STREQ R9, [R0]

BEQ AfterGame_Display

CMP R9, #2

MOVEQ R9, #2

STREQ R9, [R0]

BEQ First_phase
```

```
MOVNE R9, #12
STRNE R9, [R0]
BNE After_Game

AfterGame_Display:
LDR R8, [R2, #16]
LSR R8, R8, #24
AND R8, #15
CMP R8, #0
LDREQ R9, =0xf3380000
STREQ R9, [R2, #16]
LDREQ R9, [R4, #4]
STREQ R9, [R2]
BEQ After_Game
CMP R8, #8
LDREQ R9, =0xf3380000
STREQ R9, [R2, #16]
LDREQ R9, [R4, #4]
STREQ R9, [R2]
BEQ After_Game
CMP R8, #12
LDREQ R9, =0xf3380000
STREQ R9, [R2, #16]
LDREQ R9, [R4, #4]
STREQ R9, [R2]
BEQ After_Game


CMP R8, #3
LDREQ R9, =0x3ff9f738
STREQ R9, [R2, #16]
```

```
LDREQ R9, [R5, #4]

STREQ R9, [R2]

BEQ After_Game



CMP R8, #15

LDREQ R12, [R4, #20]

CMPEQ R12, #1


LDRLT R9, =0x1C0C0454

STRLT R9, [R2, #16]

MOVLT R8, #0

LDRLT R9, [R4, #16]

BLT Bet_Displayer


LDREQ R9, =0x383F6DF9

STREQ R9, [R2, #16]

MOVEQ R8, #0

LDREQ R9, [R4, #16]

BEQ Bet_Displayer


LDRGT R9, =0x40404040

STRGT R9, [R2, #16]

MOVGT R8, #0

LDRGT R9, [R4, #16]

BGT Bet_Displayer





B After_Game
```

```
Bet_Displayer:
CMP R9, #10
SUBGE R9, R9, #10
ADDGE R8, R8, #1
BGE Bet_Displayer
LDR R10, [R11, R9, lsl#2]
STR R10, [R2]
CMP R8, #0
BEQ After_Game
CMP R8, #10
MOVGE R9, #0
BGE Bet_Displayer_2
LDRLT R9, [R11, R8, lsl#2]
LSLLT R9, R9, #8
ADDLT R9, R9, R10
STRLT R9, [R2]
BLT After_Game

Bet_Displayer_2:
SUB R8, R8, #10
ADD R9, R9, #1
CMP R8, #10
BGE Bet_Displayer_2
LDR R10, [R11, R8, lsl#2]
LDR R12, [R2]
LSL R10, R10, #8
ADD R12, R12, R10
CMP R9, #10
LDREQ R8, =0x63F0000
ADDEQ R12, R12, R8
```

```
STREQ R12, [R2]
BEQ After_Game
LDR R10, [R11, R9, lsl#2]
LSL R10, R10, #16
ADD R12, R12, R10
STR R12, [R2]
B After_Game



Done: B end
end: B end

HEXTABLE: .word 0xF7, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x7, 0xFF, 0x6F, 0x3F, 0x1E, 0xE7, 0xF6
CARDS: .word 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0, 0x0
PLAYER: .word  0x0, 0x0, 0x0, 0x0, 0x0, 0x0
DEALER: .word  0x0, 0x0, 0x0
ORIGINALHEX: .word 0x3F, 0x6, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x7, 0xFF, 0x6F
 .end
```