

TITLE SIMON

DATA SEGMENT

```
selection db 0, "$"
rounds db 0, "$"
currentround db 0, "$"
color db 0, "$"
loopvar db 0, "$"
helpmessage db "rounds remaining: $"
delayperiod db 0, "$"
second_before db 0, "$"
second_after db 0, "$"
oldcolor db 0, "$"
loadingmessage db "Loading$"
loading db ".$"
message1 db 10, 13, "Congratulations, you've won!", 10, 13,
"$"
message6 db 10, 13, "You've lost in the $"
message8 db " round.$"
message7 db 10, 13, "insert color: $"
message2 db 10, 10, 10, 10, 10, 10, 10, 13, "Welcome to Simon
Says v1.0 made by Nick Kyparissas", 10, 13, "Press: ", 10, 13, "
'P' to play", 10, 13, " 'H' for help (game instructions)", 10, 13,
" 'X' to exit", 10, 13, "$"
message3 db 10, 13, "INSTRUCTIONS", 10, 13, "This is the
classic game of Simon Says", 10, 13, "using colors. The computer
will light ", 10, 13, "up a random series of 4 colored
buttons.You must then repeat what the computer", 10, 13, "did.",
10, 13, "The game progresses by adding", 10, 13, "a new tone to
the sequence ", 10, 13, "in every turn.", 10, 13, "The game ends
when the player ", 10, 13, "misses when repeating the sequence,
", 10, 13, "or when the player succeeds in the", 10, 13, "number
of rounds he chose." ,10, 13, "y for yellow.", 10, 13, "b for
blue.", 10, 13, "g for green.", 10, 13, "r for red.$"
message4 db 10, 13, "!ERROR!", 10, 13, "invalid input, please
try again", 10, 13, "$"
message5 db 10, 13, "--| GAME STARTS |--", 10, 13, "How many
rounds do you want to play?", 10, 13, "(2 - 40 rounds, Enter the
number of", 10, 13, " rounds you wish to play ", 10, 13, "and
press 'Enter (ret)'", 10, 13, "$"
numbers db 40 dup(0)
pointer db 0, "$"
number db 0, "$"
randomlimit db 0, "$"
```

DATA ENDS

.
.
.

```

.
.
.
valid_input:                                ;user's input vali
MOV delayperiod, 2                          ;checked and valid
CALL DELAY

MOV pointer, -1
CALL SET_GRAPHICS_MODE
LEA DX, loadingmessage
MOV AH, 09
INT 21H
random_numbers:
INC pointer
randomlimitagain:
MOV AH, 2CH
INT 21H
MOV randomlimit, DL
CMP randomlimit, 50
JB randomlimitagain
LEA DX, loading                            ;random filling of
MOV AH, 09                                  ;the colors buffer
INT 21H
CALL DELAY2
CALL RANDOM_NUMBER
MOV AL, pointer
CMP AL, rounds
JB random_numbers
////////////////////////////////////
CALL SET_GRAPHICS_MODE
MOV BL, rounds
MOV BH, 0
MOV CX, BX
CALL SET_GRAPHICS_MODE

MOV currentround, 0
.
.
.

```

.

.

.

fail:

_NUMBER PROC

PUSHA

randomgenerate:

MOV AH, 2CH

INT 21H

MOV BH, 0

MOV BL, pointer

MOV SI, BX

CMP DL, 25

JB green

CMP DL, 50

JB blue

CMP DL, 75

JB red

JMP yellow

green:

MOV numbers[SI], 2

JMP continuerandom

;random number

;via clock

[...blue...red...yellow...]

continuerandom:

CMP pointer, 0

JE endrandomnumber

MOV BH, 0

MOV BL, pointer

SUB BL, 1

MOV CH, numbers[BX]

CMP numbers[SI], CH

JE randomgenerate

endrandomnumber:

.

.

.