include irvine32.inc

.data

;-----------------------TABLE VARAIBLES---------------------------

sp2 byte " ",0

sp1 byte " ",0

num byte 1,2,3,4,5,6,7,8,9,10

byte 11,12,13,14,15,16,17,18,19,20

byte 21,22,23,24,25,25,26,27,28,29,30

byte 31,32,33,34,35,36,37,38,39,40

byte 41,42,43,44,45,46,47,48,49,50

byte 51,52,53,54,55,56,57,58,59,60

byte 61,62,63,64,65,66,67,68,69,70

byte 71,72,73,74,75,76,77,78,79,80

byte 81,82,83,84,85,86,87,88,89,90

byte 91,92,93,94,95,96,97,98,99,100

msg1 byte "press one for player 1!",0

msg2 byte "press two for player 2!",0

msg3 byte " player one's score is : ",0

msg4 byte " player two's score is : ",0

;-----------------------RG VARAIBLES---------------------------

Rolling byte " Rolling Dice...",0

d1 byte ".",0

val1 dword 0

val2 dword 0

;-----------------------MENU VARAIBLES---------------------------

M1 byte " Press 1 for selecting level : ",0

M2 byte " Press 2 for insrtuctions: ",0

M3 byte " Press 3 for exit: ",0

M byte " MENU for Snake and Ladders ",0

;-----------------------INSTRUCTIONS VARAIBLES---------------------------

I1 byte "You have to press 1 for player one's Rolling Dice and Press 2 for player two's Rolling Dice.There are many snakes in the table as well as Ladders depending on the level of game.so beware of snakes and enjoy climbing of ladders.",0

;-----------------------other VARAIBLES---------------------------

WRONGE BYTE " Wrong entry input again!",0

YOU1 byte " CONGRATULATIONS! Player one have won the game! ",0

YOU2 byte " CONGRATULATIONS! Player two have won the game! ",0

w1 byte "You won!"

;m byte ?

;b1 byte 0

l4 byte "--->back to menu",0

p1 dword 0

p2 dword 0

;c dword ?

c1 dword ?

c2 dword ?

c3 dword ?

a byte 0

b byte 0

o byte 0

b1 byte 0

pr byte " Ladders are at position 3 , 19 , 15 , 50 , 71 , 84.",0

pr1 byte " Snakes are at position 30 , 51 , 67 , 90 , 99.",0

w12 byte " \_ \_ \_ \_ \_ \_ \_ \_ \_ ",0

w2 byte " \ /\ / | | | |\ /| | ",0

w3 byte " \ / \ / |\_ \_ \_ | | | \ / | |\_ \_ \_ ",0

w4 byte " \ / \ / | | | | \ / | | ",0

w5 byte " \/ \/ |\_ \_ \_ |\_ \_ \_ |\_ \_ \_ | \/ | |\_ \_ \_ ",0

w6 byte " \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_",0

w7 byte " | | | \_ \_ \_ | | \_ \_ \_ | | \_ \_ \_ \\_ \_ | | | \_ \_ \_ | | \_ \_ \_ \_ \_|",0

w8 byte " | | | | | | | | | | | | \\_ \_ | | | | | | | | | \_ \_ \_ ",0

w9 byte " | | | | | | | |\_ \_ \_| | | | \_ \_|| | | | | | | | | |\_ |",0

w10 byte" | |\_ \_ \_ | |\_ \_ \_| | | | | | | |\_ \_ \_/ \_ \_| | | | | | | | |\_ \_ \_| |",0

w11 byte" |\_|\_ \_ \_| |\_ \_ \_ \_ \_| |\_| |\_| |\_ \_ \_ \_ / |\_| |\_| |\_| |\_ \_ \_ \_ \_ \_|",0

.code

;----------------------------WELCOME-------------------------------

WELCOM proc

mov edx,offset w12

call writestring

call crlf

mov edx,offset w2

call writestring

call crlf

mov edx,offset w3

call writestring

call crlf

mov edx,offset w4

call writestring

call crlf

mov edx,offset w5

call writestring

call crlf

call crlf

call crlf

mov edx,offset w6

call writestring

call crlf

mov edx,offset w7

call writestring

call crlf

mov edx,offset w8

call writestring

call crlf

mov edx,offset w9

call writestring

call crlf

mov edx,offset w10

call writestring

call crlf

mov edx,offset w11

call writestring

call crlf

ret

WELCOM endp

;----------------------------WELCOME-------------------------------

;------------------------------Predictions-------------------------

Predictions proc

mov edx,offset pr

call writestring

call crlf

xor edx,edx

mov edx,offset pr1

call writestring

call crlf

ret

Predictions endp

;------------------------------End of Predictions-------------------------

;--------------------------------STARTING OF TABLE PROC-----------------------

table proc

mov eax,white

call settextcolor

xor eax,eax

mov ecx,111

mov esi,offset num

mov ebx,0

L2:

.if ebx==10

mov b,1

call crlf

mov ebx,0

.else

.if b==0

mov al,[esi]

call writedec

mov edx,offset sp2

call writestring

.else

mov al,[esi]

call writedec

mov edx,offset sp1

call writestring

.endif

add esi,1

inc ebx

.endif

loop L2

mov b,0

call crlf

call crlf

mov eax,red

call settextcolor

call Predictions

call crlf

mov edx,offset msg1

call writestring

mov edx,offset msg3

call writestring

xor eax,eax

mov eax,blue

call settextcolor

mov eax,p1

call writedec

call crlf

mov eax,red

call settextcolor

mov edx,offset msg2

call writestring

mov edx,offset msg4

call writestring

mov eax,blue

call settextcolor

xor eax,eax

mov eax,p2

call writedec

call crlf

ret

table endP

;--------------------------------END TABLE-----------------------

;--------------------------------RANDOM NUMBER GENERATOR-----------------------

RG proc

call clrscr

mov eax,yellow

call settextcolor

mov edx,offset Rolling

call writestring

mov eax,700

call delay

mov eax,offset d1

call writechar

mov eax,700

call delay

mov eax,offset d1

call writechar

mov eax,700

call delay

mov eax,offset d1

call writechar

call crlf

mov eax,7

call randomrange

push eax

mov eax,yellow

call settextcolor

pop eax

call writedec

mov val1,eax

mov eax,1000

call delay

ret

RG endP

;--------------------------------END OF RANDOM NUMBER GENERATOR-----------------------

;--------------------------------WINNING MESSAGE1-----------------------

WIN1 proc

mov edx,offset YOU1

call writestring

ret

WIN1 endP

;--------------------------------END OF WINNING MESSAGE1-----------------------

;--------------------------------WINNING MESSAGE2-----------------------

WIN2 proc

mov edx,offset YOU2

call writestring

ret

WIN2 endP

;--------------------------------END OF WINNING MESSAGE2-----------------------

;--------------------------------CHECKING OF SNAKES AND LADDERS1-----------------------

CONDITIONS1 proc

.if p1==100

mov b1,1

call WIN1

.elseif p1==3

add p1,10

.elseif p1==19

add p1,10

.elseif p1==15

add p1,45

.elseif p1==50

add p1,27

.elseif p1==71

add p1,10

.elseif p1==84

add p1,10

.elseif p1==30

sub p1,16

.elseif p1==51

sub p1,26

.elseif p1==67

sub p1,23

.elseif p1==90

sub p1,55

.elseif p1==99

sub p1,19

.endif

ret

CONDITIONS1 endP

;--------------------------------END OF CHECKING OF SNAKES AND LADDERS1-----------------------

;--------------------------------CHECKING OF SNAKES AND LADDERS2-----------------------

CONDITIONS2 proc

.if p2==100

mov b1,1

call WIN2

.elseif p2==3

add p2,10

.elseif p2==19

add p2,10

.elseif p2==15

add p2,45

.elseif p2==50

add p2,27

.elseif p2==71

add p2,10

.elseif p2==84

add p2,10

.elseif p2==30

sub p2,16

.elseif p2==51

sub p2,26

.elseif p2==67

sub p2,23

.elseif p2==90

sub p2,55

.elseif p2==99

sub p2,19

.endif

ret

CONDITIONS2 endP

;--------------------------------END OF CHECKING OF SNAKES AND LADDERS2-----------------------

;--------------------------------MENU PROC-----------------------

MENU proc

call clrscr

mov edx,offset m

call writestring

call crlf

call crlf

call crlf

call crlf

mov edx,offset m1

call writestring

call crlf

mov edx,offset m2

call writestring

call crlf

mov edx,offset m3

call writestring

call crlf

;call readint

ret

MENU endP

;--------------------------------END OF MENU PROC-----------------------

;--------------------------------INSTRUCTIONS-----------------------

INSTRUCTIONS proc

call clrscr

mov edx,offset I1

call writestring

call crlf

mov edx,offset L4

call writestring

call readchar

ret

INSTRUCTIONS endP

;--------------------------------INSTRUCTIONS-----------------------

;--------------------------------EASY-----------------------

EASY proc

.while b1==0

call clrscr

call table

p9:

call readint

mov c1,eax

.if c1==1

call RG

mov ebx,0

add ebx,val1

.if ebx<=100

add p1,ebx

call CONDITIONS1

.endif

.elseif c1==2

call RG

mov ebx,0

add ebx,val1

.if ebx<=100

add p2,ebx

call CONDITIONS2

.endif

.else

mov edx,offset WRONGE

call writestring

jmp p9

.endif

.endw

ret

EASY endP

;--------------------------------END OF EASY-----------------------

main proc

mov ecx,0

.while ecx!=5

call WELCOM

mov eax,700

call delay

call clrscr

mov eax,700

call delay

inc ecx

.endw

call clrscr

po:

call MENU

call readint

mov c3,eax

.if c3==1

call EASY

.elseif c3==2

call INSTRUCTIONS

jmp po

.elseif c3==3

exit

.else

mov edx,offset WRONGE

call writestring

jmp po

.endif

exit

main endP

end main