****

**COAL**

***Project Proposal***

***CLASS ID: 107924***

**Project Title:**

**Quiz Management System**

**Project Member:**

|  |  |  |
| --- | --- | --- |
| **Sno** | **Student ID** | **Student Name** |
| **1** | **11508** | **Muzamil Khan** |
| **2** | **10150** | **Maryam Javed** |
| **3** | **10657** | **Somia Akhtar** |

**Project Discription:**

**We are using assembly langauge in our project. This is simple mini quiz management project on console where user answer following question asked by computer similar to quiz and atlast result appear after question’s limit reached and graded pass or fail on the basis of correct answers by user. Following are some screen pages which we will use in our project.**

1. **Home Screen**
2. **Main Quiz Screen**
3. **Result Screen**

**Coding:**

Include Irvine32.Inc

.data

myvar1 BYTE "Please Enter Y To Continue Or N To Exit: ",0

myvar2 BYTE "Welcome To Quiz Management System: ",0

myvar3 BYTE "Game Exit",0

mychar BYTE ?

msg1 byte "\*\*\*\*\*\* Welcome To Quiz System \*\*\*\*\*\* ",0

msg2 byte " BELOW MENTIONED SOME RULES ",0

msg3 byte " Rule 1: If your answer is correct you will get 1 point.",0

msg4 byte " Rule 2: But if your answer is wrong 1 point will be deducted from your points",0

msg5 byte " Press enter to start your turn",0

msg6 byte "Good! Your answer is Right..",0

msg7 byte "Ohh Sorry!! Wrong answer..",0

msg8 byte "Your quiz has been completed",0

msg9 byte "And here comes your total obtained scores: ",0

msg10 byte "Kindly press 1 IF you want to restart the quiz ELSE press 0 to exit",0

msg11 byte "Thank You For being Here, You played so well",0

msg12 byte "Press 1 to restart quiz:",0

msg13 byte "GAME END!!!!:",0

Question1 byte "Q1) Which one of the following is not a primary color?:",0

QA1 byte "1) yellow 2)red 3) green 4) blue",0

Question2 byte "Q2) Which is the colour of sky",0

QA2 byte "1) violet 2) blue 3) black 4) red ",0

Question3 byte "Q3) Which of the following is not a natural number:",0

QA3 byte "1) 16 2) 32 3) 4 4) 0",0

Question4 byte "Q4) Which one of following has largest precedence:",0

QA4 byte "1) () 2)\* 3) + 4) /",0

Question5 byte "Q5) How many alphabets are there in english:",0

QA5 byte "1) 22 2) 223 3) 23 4) 26",0

Question6 byte "Q6) Pakistan is in:",0

QA6 byte "1) iceland 2) africa 3) asia 4) europe",0

Question7 byte "Q7) 1 kilo is equal to:",0

QA7 byte "1) 1900grams 2) 1000 gram 3) 100 gram 4) 2000gramss",0

Question8 byte "Q8) Total provinces of pakistan:",0

QA8 byte "1) 4 2) 5 3) 3 4) 6",0

Question9 byte "'Q9) When pakistan came into being:",0

QA9 byte "1) 1947 2) 1948 3) 1945 4) 1949",0

Question10 byte "Q10) Total numbers of cards in the deck are:",0

QA10 byte "1) 52 2) 53 3) 54 4) 51",0

Question11 byte "Q11) 2 + 2 =:",0

QA11 byte "1) 4 2) 53 3) 54 4) 51",0

Question12 byte "Q12) How Many Days in February :",0

QA12 byte "1) 28 2) 53 3) 54 4) 51",0

Question13 byte "Q13) Total No of Wheels in Alto Car:",0

QA13 byte "1) 4 2) 8 3) 3 4) 1",0

Question14 byte "Q14) Karachi is the biggest city of pakistan? 1 for yes and 2 for no:",0

QA14 byte "1) 1 2) 2 3) 7 4) 8",0

Question15 byte "Q15) 2 is even no ? 1 for yes and 2 for no :",0

QA15 byte "1) 1 2) 2 3) 8 4) 7",0

Points dword 0

Value dword ?

.code

main PROC

call msg

call crlf

call crlf

mov edx, offset myvar1

call writestring

call readchar

call writechar

call crlf

mov mychar,al

.if(mychar == "Y")

mov al, mychar

mov edx, offset myvar2

call writestring

call crlf

call clrscr

mov dh,3

mov dl,20

call GotoXY

call game

.elseif(mychar == "y")

mov al, mychar

mov edx, offset myvar2

call writestring

call crlf

call clrscr

mov dh,3

mov dl,20

call GotoXY

call game

.else

mov edx, offset myvar3

call writestring

.endif

; message procedure

msg proc

mov edx,0

mov edx,offset msg1

call writestring

call crlf

mov edx,0

mov edx,offset msg2

call writestring

call crlf

mov edx,0

mov edx,offset msg3

call writestring

call crlf

mov edx,0

mov edx,offset msg4

call writestring

call crlf

mov edx,0

mov edx,offset msg5

call writestring

call crlf

ret

msg endp

game proc

start:

call crlf

call crlf

call crlf

mov bx,0

;Question no 1

mov edx,0

mov edx,offset Question1

call writestring

call crlf

mov edx,0

mov edx,offset QA1

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal

jne Notequal

Equal:

mov eax,0

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp labelQ2

Notequal:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 2

labelQ2:

mov edx,0

mov edx,offset Question2

call writestring

call crlf

mov edx,0

mov edx,offset QA2

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 2

cmp Value,eax

je Equal2

jne Notequal2

Equal2:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q3

Notequal2:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 3

Q3:

mov edx,0

mov edx,offset Question3

call writestring

call crlf

mov edx,0

mov edx,offset QA3

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 4

cmp Value,eax

je Equal3

jne Notequal3

Equal3:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q4

Notequal3:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 4

Q4:

mov edx,0

mov edx,offset Question4

call writestring

call crlf

mov edx,0

mov edx,offset QA4

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal4

jne Notequal4

Equal4:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q5

Notequal4:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 5

Q5:

mov edx,0

mov edx,offset Question5

call writestring

call crlf

mov edx,0

mov edx,offset QA5

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 4

cmp Value,eax

je Equal5

jne Notequal5

Equal5:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q6

Notequal5:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 6

Q6:

mov edx,0

mov edx,offset Question6

call writestring

call crlf

mov edx,0

mov edx,offset QA6

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 3

cmp Value,eax

je Equal6

jne Notequal6

Equal6:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q7

Notequal6:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 7

Q7:

mov edx,0

mov edx,offset Question7

call writestring

call crlf

mov edx,0

mov edx,offset QA7

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 2

cmp Value,eax

je Equal7

jne Notequal7

Equal7:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q8

Notequal7:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 8

Q8:

mov edx,0

mov edx,offset Question8

call writestring

call crlf

mov edx,0

mov edx,offset QA8

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal8

jne Notequal8

Equal8:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q9

Notequal8:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 9

Q9:

mov edx,0

mov edx,offset Question9

call writestring

call crlf

mov edx,0

mov edx,offset QA9

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal9

jne Notequal9

Equal9:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q10

Notequal9:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

;Question no 10

Q10:

mov edx,0

mov edx,offset Question10

call writestring

call crlf

mov edx,0

mov edx,offset QA10

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal10

jne Notequal10

Equal10:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp q11

;jmp Result

Notequal10:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

q11:

mov edx,0

mov edx,offset Question11

call writestring

call crlf

mov edx,0

mov edx,offset QA11

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal11

jne Notequal11

Equal11:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q12

Notequal11:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

q12:

mov edx,0

mov edx,offset Question12

call writestring

call crlf

mov edx,0

mov edx,offset QA12

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal12

jne Notequal12

Equal12:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q13

Notequal12:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

q13:

mov edx,0

mov edx,offset Question13

call writestring

call crlf

mov edx,0

mov edx,offset QA13

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal13

jne Notequal13

Equal13:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q14

Notequal13:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

q14:

mov edx,0

mov edx,offset Question14

call writestring

call crlf

mov edx,0

mov edx,offset QA14

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal14

jne Notequal14

Equal14:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Q15

Notequal14:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

q15:

mov edx,0

mov edx,offset Question15

call writestring

call crlf

mov edx,0

mov edx,offset QA15

call writestring

call crlf

call Readint

mov Value, eax

mov eax,0

mov eax, 1

cmp Value,eax

je Equal15

jne Notequal15

Equal15:

mov eax,Points

inc eax

mov Points,eax

mov edx,0

mov edx,offset msg6

call writestring

call crlf

call crlf

jmp Result

Notequal15:

mov edx,0

mov edx,offset msg7

call writestring

call crlf

call crlf

Result:

mov edx,0

mov edx,offset msg9

call writestring

mov eax, 0

mov eax,Points

call writeint

call crlf

mov edx,0

mov edx,offset msg12

call writestring

call readint

cmp eax,1

je start

jne endd

endd:

call clrscr

mov edx,offset msg13

call writestring

call readint

ret

game endp

exit

main ENDP

END main

**Screenshot Of Output:**

