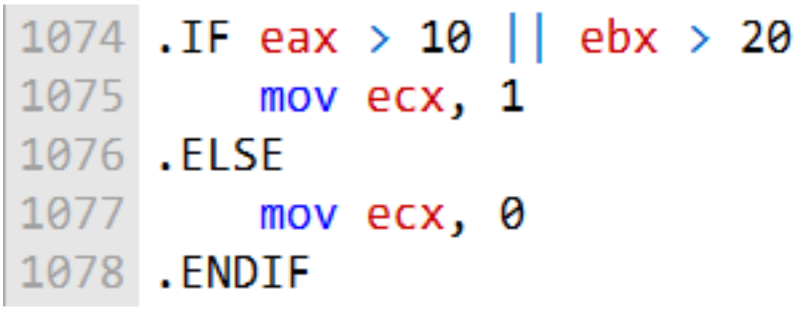
Compound Expressions

Compound Boolean expressions allow you to combine two or more Boolean expressions using the logical OR and AND operators.

The **logical OR operator (||)** returns true if either of the Boolean expressions is true. The logical AND operator (&&) returns true if both of the Boolean expressions are true. Using Compound Expressions with the .IF Directive

You can use compound expressions with the .IF directive to control the flow of execution of your program.

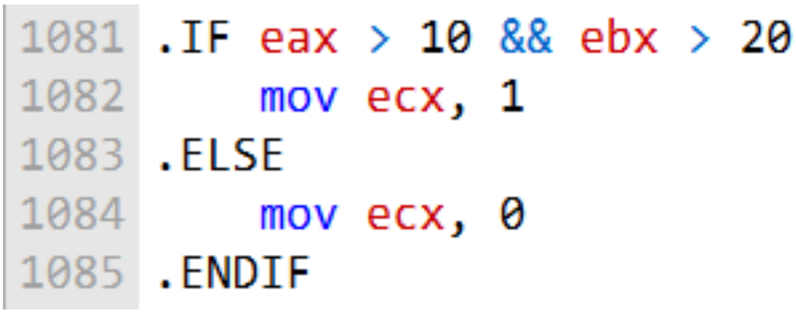
For example, the following .IF directive uses the logical OR operator to compare the values of the eax and ebx registers:



```
.IF eax > 10 || ebx > 20
    mov ecx, 1
.ELSE
    mov ecx, 0
.ENDIF
```

This code will move the value 1 to the ecx register if the value of eax is greater than 10 or the value of ebx is greater than 20. Otherwise, the code will move the value 0 to the ecx register.

The following .IF directive uses the logical AND operator to compare the values of the eax and ebx registers:



```
.IF eax > 10 && ebx > 20
    mov ecx, 1
.ELSE
    mov ecx, 0
.ENDIF
```

This code will move the value 1 to the ecx register only if the value of eax is greater than 10 and the value of ebx is greater than 20. Otherwise, the code will move the value 0 to the ecx register.

Compound Boolean expressions can be used to create more complex conditional statements using the .IF directive. This can be helpful for controlling the flow of execution of your program in response to different conditions.

=============================

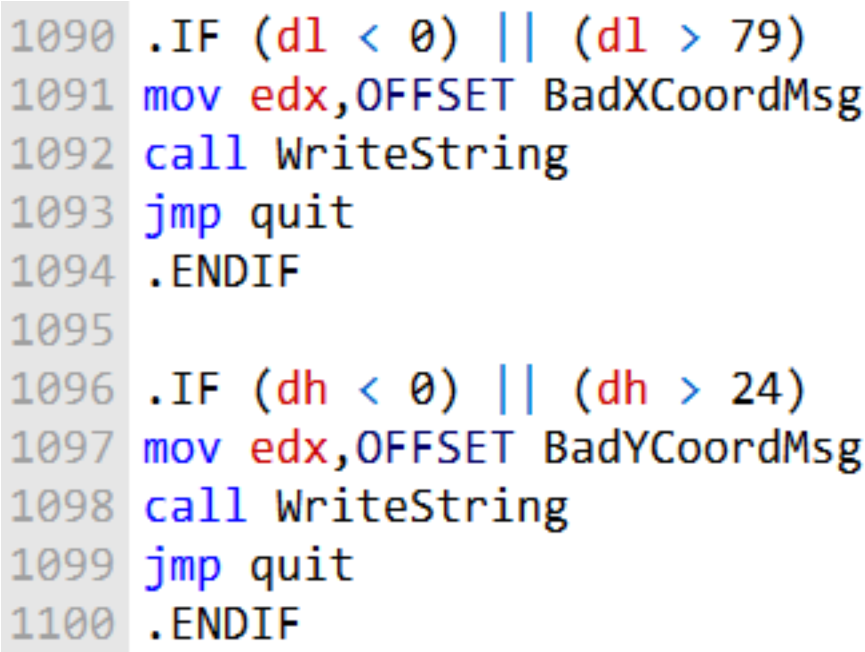
SetCursorPosition Example

=============================

The SetCursorPosition procedure sets the cursor position to the specified coordinates. It receives two input parameters: DL (X-coordinate) and DH (Y-coordinate).

The procedure first checks if the X-coordinate and Y-coordinate are within the valid ranges. If either coordinate is out of range, the procedure displays an error message and exits.

The following code shows the range checking code in the SetCursorPosition procedure



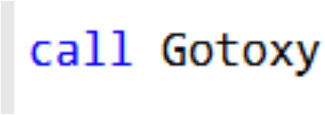
```
.IF (dl < 0) || (dl > 79)
mov edx,OFFSET BadXCoordMsg
call WriteString
jmp quit
.ENDIF

.IF (dh < 0) || (dh > 24)
mov edx,OFFSET BadYCoordMsg
call WriteString
jmp quit
.ENDIF
```

The .IF directive is used to check if the X-coordinate or Y-coordinate is out of range. The logical OR operator (||) is used to combine the two conditions.

If either condition is true, the procedure displays an error message and exits.

If the X-coordinate and Y-coordinate are within the valid ranges, the procedure calls the Gotoxy procedure to set the cursor position.

The following code shows the code that sets the cursor position:



```
call Gotoxy
```

The Gotoxy procedure is a built-in MASM procedure that sets the cursor position to the specified coordinates.

The SetCursorPosition procedure is an example of how to use the .IF directive to range check input parameters. This can be helpful for preventing errors in your program.

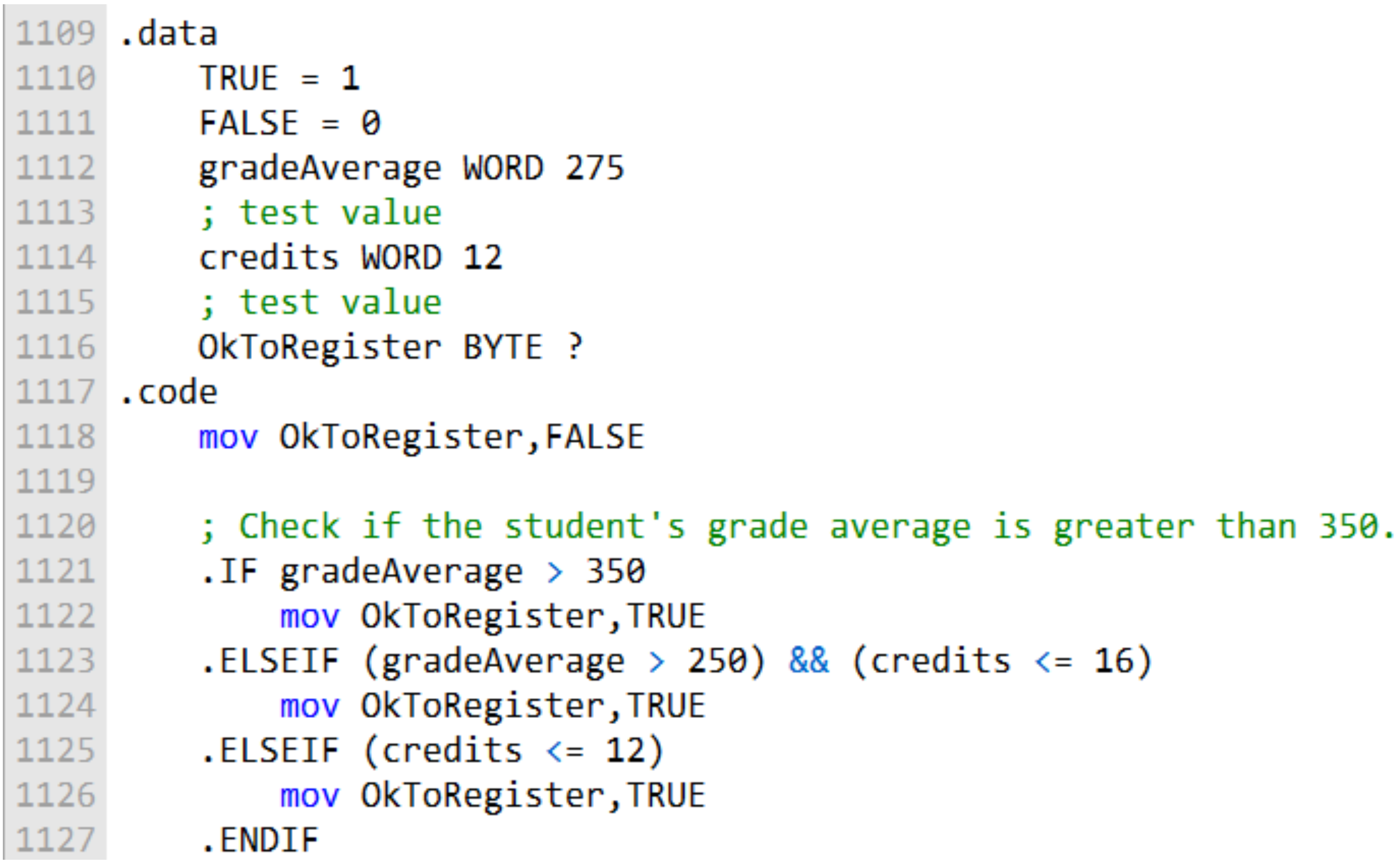
==================================

College Registration Example

===================================

The college registration example you provided uses the .IF, .ELSEIF, and .ENDIF directives to implement a multiway branch structure. The structure checks the student's grade average and number of credits to determine whether or not the student can register.

The following is a simpler explanation of the code:



```
.data
    TRUE = 1
    FALSE = 0
    gradeAverage WORD 275
    ; test value
    credits WORD 12
    ; test value
    OkToRegister BYTE ?
.code
    mov OkToRegister,FALSE

    ; Check if the student's grade average is greater than 350.
    .IF gradeAverage > 350
        mov OkToRegister,TRUE
    .ELSEIF (gradeAverage > 250) && (credits <= 16)
        mov OkToRegister,TRUE
    .ELSEIF (credits <= 12)
        mov OkToRegister,TRUE
    .ENDIF
```

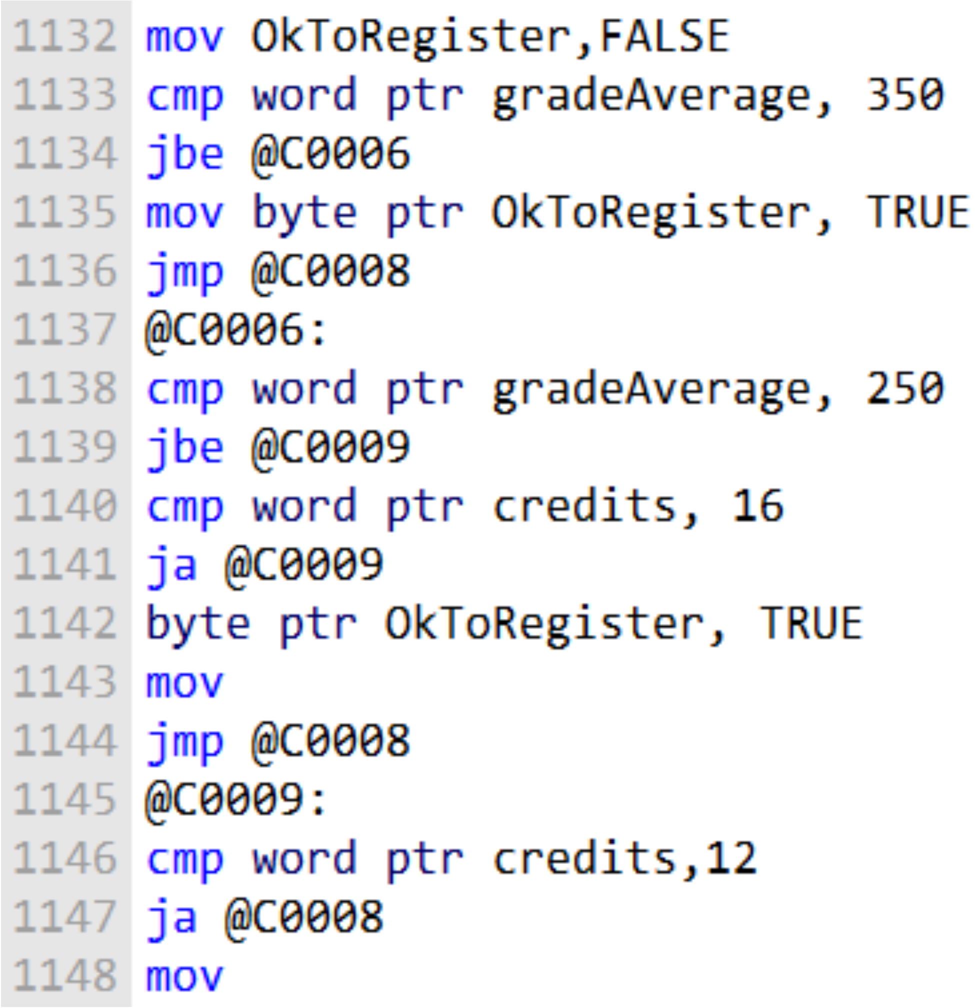
The .IF directive is used to check the first condition: whether the student's grade average is greater than 350. If the condition is true, the mov instruction sets the OkToRegister variable to TRUE.

The .ELSEIF directive is used to check the second condition: whether the student's grade average is greater than 250 and the number of credits the student wants to take is less than or equal to 16. If both conditions are true, the mov instruction sets the OkToRegister variable to TRUE.

The .ELSEIF directive is used to check the third condition: whether the number of credits the student wants to take is less than or equal to 12. If the condition is true, the mov instruction sets the OkToRegister variable to TRUE.

If none of the conditions are true, the OkToRegister variable will remain set to FALSE.

The following is a breakdown of the generated code that you sent:



```
mov OkToRegister,FALSE
cmp word ptr gradeAverage, 350
jbe @C0006
mov byte ptr OkToRegister, TRUE
jmp @C0008
@C0006:
cmp word ptr gradeAverage, 250
jbe @C0009
cmp word ptr credits, 16
ja @C0009
byte ptr OkToRegister, TRUE
mov
jmp @C0008
@C0009:
cmp word ptr credits,12
ja @C0008
mov
```

- The first line of code moves the value FALSE to the OkToRegister variable.

- The next two lines of code compare the student's grade average to 350. If the grade average is greater than 350, the program jumps to the label @C0008. Otherwise, the program continues to the next line of code.

- The next three lines of code compare the student's grade average to 250 and the number of credits the student wants to take to 16. If both conditions are true, the program jumps to the label @C0008. Otherwise, the program continues to the next line of code.

- The next two lines of code compare the number of credits the student wants to take to 12. If the number of credits is less than or equal to 12, the program jumps to the label @C0008. Otherwise, the program continues to the next line of code.

- The label @C0008 is a jump target. If the program jumps to this label, the OkToRegister variable will be set to TRUE.

- The program exits at the end of the code.