

Translate the following code into assembly language (MASM).

Assume that X, Y, Z, i and j are **16-bit signed integers**.

```
while( (i>14) AND (j <= 5)){
```

```
    X = Y + Z;
```

```
    Y = Y + 1;
```

```
    if (Z == 7)
```

```
break;
```

```
    else
```

```
        Z = Z + 2;
```

```
        j = j + 1;
```

```
        i = i - 1;
```

```
}
```

You need to translate all the parts of the code above, even if they don't execute. You are not allowed to make any logical reduction to the code. You are NOT allowed to use directives - (.IF, .ELSE, .REPEAT, .UNTIL, .WHILE, etc). You need to implement all the conditional control flows

Answer:

begin_while:

CMP i, 14

JLE next

CMP j, 5

JG next

MOV EAX, Y

ADD EAX, Z

MOV x, EAX

INC y

CMP z, 7

JE next

ADD z, 2

INC j

DEC i

JMP begin_while

Next: