1. In code fragment *R*, suppose that variables *y* and *z* are declared as *DWORD*, and *z* contains a non-negative value. Assuming that the value of *z* is small enough so that no overflow occurs, what is calculated and stored in memory location *y*? (Your answer should be an algebraic expression, not a literal value.)

**;code fragment R**

**mov eax,1**

**mov ebx,3**

**mov ecx,z**

**cmp ecx,0**

**je store**

**top:**

**mul ebx**

**loop top store:**

**mov y,eax**

Y= 3^z

1. Given the signed numbers -25 and 124
   1. Convert them to **8-bit binary** numbers

0001 1001

Reverse

1110 0110

Add 1

1110 0111

-25 = 1110 0111

0111 1100

* 1. **Add them** together and show that your answer is valid by converting back to decimal

1110 0111

0001 1001

110

Use the following declarations and address information for problems # 3 – 5

**MAXSIZE = 100**

**.data**

**list DWORD MAXSIZE DUP (?)**

Assume the *list* array starts at memory address 1000h. All addresses are given in hexadecimal; other values are in decimal.

1. How many bytes of memory are reserved by the declaration of *list*?

**400 bytes**

1. After code fragment *S* is executed, what are the contents of the first 6 elements of the *list* array.

**;code fragment S**

**mov edi,OFFSET list mov eax,MAXSIZE**

**top:**

**cmp eax,0**

**jle quit**

**mov [edi],eax dec eax**

**add edi,TYPE list jmp top**

**quit:**

100, 99, 98, 97

1. After code fragment *T* is executed, what are the contents of the first 6 elements of the *list* array.

**;code fragment T**

**mov edi,OFFSET list mov ebx,0**

**mov eax,1**

**mov ecx,MAXSIZE**

**top:**

**mov [edi+ebx],eax inc eax**

**add ebx,TYPE list loop top**

1, 2 , 3 4, 4,5

1. Given the truth table below, find the Boolean expression, simplify it and then draw the digital circuit.

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **C** | **Y** |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

¬A ¬B ¬C + ¬A ¬B C + ¬ AB¬C + A¬B¬C + AB¬C + ABC

¬B C = B¬C

¬A¬B(C + ¬C) + B¬C(A + ¬A) + AC(¬B+B)

¬A¬B + B¬C + AC

1. Given the **odd-parity** Hamming code 0000 1010 1111. Determine if it includes errors and if it does, fix the code

**My odd parity:**

**1 0 0 0 1 0 1 1 1 1 1 1**

**There are errors in the 5th bit and the last bit.**

1. Give the postfix (RPN) form of the infix expression **a - (b + c) / (d \* e)**

**a bc+ de\* / -**

1. Find the integer value of the postfix (RPN) expression

**2 7 + 3 2 - + 5 / 2 \***

**2((2+7) + (3-2) )/ 5**

**2( 9 + 1) / 5**

**2 ( 10 ) / 5**

**20/5**

**4**

1. Give the infix form of the postfix (RPN) expression **a b c d + - e / \***

(a\*((b-(c+d))/e))