1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | S1 = “001001001” | S2 = “0110101101” | S = “0000000000” | Soma |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 00 | 01 | 00 | 0 |
| 3 | 001 | 011 | 001 | 1 |
| 4 | 0011 | 0110 | 0010 | 1 |
| 5 | 00110 | 01101 | 00100 | 1 |
| 6 | 001100 | 011010 | 001000 | 1 |
| 7 | 0011001 | 0110101 | 0010001 | 2 |
| 8 | 00110010 | 01101011 | 00100010 | 2 |
| 9 | 001100100 | 011010110 | 001000100 | 2 |
| 10 | 0011001001 | 0110101101 | 0010001001 | 3 |

2)

|  |  |  |  |
| --- | --- | --- | --- |
| MAT[1, 1] = 2 | MAT[1, 2] = 5 | MAT[1, 3] = 1 | MAT[1, 4] = 6 |
| MAT[2, 1] = 1 | MAT[2, 2] = 0 | MAT[2, 3] = 4 | MAT[2, 4] = 2 |
| MAT[3, 1] = 8 | MAT[3, 2] = 9 | MAT[3, 3] = 1 | MAT[3, 4] = 10 |
| MAT[4, 1 ] = 1 | MAT[4, 2] = 3 | MAT[4, 3] = 4 | MAT[4, 4] = 8 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | j | k | L[k] | D[i] |
| 1 | 1 | 0+1 = 1 | [ 1 ] = MAT[1, 1] = 2 | [ 1 ] = MAT[1, 4] = 6 |
| 1 | 2 | 1+1 = 2 | [ 2 ] = MAT[1, 2] = 5 | ==//==//== |
| 1 | 3 | 2+1 = 3 | [ 3 ] = MAT[1, 3] = 1 | ==//==//== |
| 1 | 4 | 3+1 = 4 | [ 4 ] = MAT[1, 4] = 6 | ==//==//== |
| 2 | 1 | ==//==//== | ==//==//== | ==//==//== |
| 2 | 2 | 4+1 = 5 | [ 5 ] = MAT[2, 2] = 0 | [ 2 ] = MAT[2, 3] = 4 |
| 2 | 3 | 5+1 = 6 | [ 6 ] = MAT[2, 3] = 4 | ==//==//== |
| 2 | 4 | 6+1 = 7 | [ 7 ] = MAT[2, 4] = 2 | ==//==//== |
| 3 | 1...2 | ==//==//== | ==//==//== | ==//==//== |
| 3 | 3 | 7+1 = 8 | [ 8 ] = MAT[3, 3] = 1 | [ 3 ] = MAT[3, 2] = 9 |
| 3 | 4 | 8+1 = 9 | [ 9 ] = MAT[3, 4] = 10 | ==//==//== |
| 4 | 1...3 | ==//==//== | ==//==//== | ==//==//== |
| 4 | 4 | 9+1 = 10 | [ 10 ] = MAT[4, 4] = 8 | [ 4 ] = MAT[4, 1] = 1 |

|  |  |
| --- | --- |
| L[k] | L[k] = X |
| 1 | 2 |
| 2 | 5 |
| 3 | 1 |
| 4 | 6 |
| 5 | 0 |
| 6 | 4 |
| 7 | 2 |
| 8 | 1 |
| 9 | 10 |
| 10 | 8 |

|  |  |
| --- | --- |
| D[i] | D[i] = y |
| 1 | 6 |
| 2 | 4 |
| 3 | 9 |
| 4 | 1 |

4)

|  |
| --- |
| #include <stdio.h> |
|  | #include <stdlib.h> |
|  |  |
|  |  |
|  | void main(){ |
|  |  |
|  | // Variaveis |
|  | int M[3][4]; |
|  | int menor; |
|  | int i, j; |
|  | int indice\_i; |
|  | int indice\_j; |
|  |  |
|  |  |
|  | // Ler Matriz |
|  | for (i = 0; i <= 2; i++){ |
|  | for (j = 0; j <= 3; j++){ |
|  | printf("M[%d,%d] = ", (i + 1), (j + 1)); |
|  | scanf("%d", &M[i][j]); |
|  | } |
|  | } |
|  |  |
|  | // inicializa a variavel menor com o primeiro numero da matriz |
|  | menor = M[0][0]; // considera o numero ZERO como o primeiro elemento da Matriz |
|  | indice\_i = 0; // indice i da Matriz M[i][j] |
|  | indice\_j = 0; // indice j da Matriz M[i][j] |
|  |  |
|  | // Procurar o menor numero da Matriz |
|  | for (i = 0; i <= 2; i++){ |
|  | for (j = 0; j <= 3; j++){ |
|  | if (M[i][j] < menor){ |
|  | menor = M[i][j]; |
|  | indice\_i = i; |
|  | indice\_j = j; |
|  |  |
|  | } |
|  | } |
|  | } |
|  |  |
|  |  |
|  | // imprimir o menor numero |
|  | // considera o numero ZERO como o primeiro elemento da Matriz |
|  | printf("o menor numero ocorre em M[%d, %d] = %d", (indice\_i+1), (indice\_j+1), menor); |
|  |  |
|  | system("pause>>null"); |
|  |  |
|  | } |

5)

D