|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Julio Cesar Saynez Fabian |
| *Asignatura:* | EDA |
| *Grupo:* | 17 |
| *No de Práctica(s):* | 1 |
| *Integrante(s):* | JONATHAN OMAR CALZADA MARTINEZ |
|  |  |
|  |  |
| *Semestre:* | 2018-2 |
| *Fecha de entrega:* | 26 de febrero de 2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

|  |
| --- |
| EJERCICIO 1  Escribe un programa que lea una cadena(máx. 250 caracteres) y cambie la primera letra de cada palara a mayúscula y el resto en minúscula  Ej : Entrada : Buen día a todos  Salida : Buen Día A Todos  (2 puntos) |
| * 1. [CÓDIGO] #include <stdio.h>   2. #define s scanf   3. #define p printf   4. char car[250];   5. int i;   6. main (void){   7. p("\nIngrese la cadena \n");   8. for (i=0;i<250;i++)   9. {   10. s("%[^\n]", car);   11. for (i=0;i<250;i++){   12. if (car[i]==car[0])   13. car[i]-=32;   14. if (car[i]==32)   15. car[i+1]-=32;   16. }   17. p("\n%s\n",car);   18. }   } |

|  |
| --- |
| EJERCICIO 2  Lee dos fechas en el formato dd-mm-año (también es valido dd/mm/año), después imprimir ambas fechas comenzando por la más reciente y la salida debe ser en un formato extendido.  (2 puntos)  Escribe dos fechas:  01/04/2017 (enter)  01/01/2018 (enter)  Salida en forma extendida:  1 de enero de 2018  1 de abril de 2017 |
| * 1. [CÓDIGO] #include <stdio.h>   2. #include<stdlib.h>   3. #define s scanf   4. #define p printf   5. char vec1[11],vec2[11];   6. int i,sum,sum2,cont=1;   7. main (void){   8. p("\nIngrese la fecha \n");   9. fflush(stdin);   10. s("%[^\n]",vec1);   11. p("\nIngrese la fecha \n");   12. fflush(stdin);   13. s("%[^\n]",vec2);   14. sum=vec1[3]+vec1[4];   15. sum2=vec2[3]+vec2[4];   16. sum-=48;   17. sum2-=48;   18. if(sum<13){   19. switch (sum){   20. case '1':   21. p("Fecha 1: %i de Enero de %i\n",vec1,sum);   22. break ;   23. case '2':   24. p("Fecha 1: %i de Feb de %i\n",vec1,sum);   25. break;   26. case '3':   27. p("Fecha 1: %i de Marzo de %i\n",vec1,sum);   28. break;   29. case '4':   30. p("Fecha 1: %i de Abril de %i\n",vec1,sum);   31. break;   32. case '5':   33. p("Fecha 1: %i de Mayo de %i\n",vec1,sum);   34. break;   35. case '6':   36. p("Fecha 1: %i de Junio de %i\n",vec1,sum);   37. break;   38. case '7':   39. p("Fecha 1: %i de Julio de %i\n",vec1,sum);   40. break;   41. case '8':   42. p("Fecha 1: %i de Agosto de %i\n",vec1,sum);   43. break;   44. case '9':   45. p("Fecha 1: %i de Septiembre de %i\n",vec1,sum);   46. break;   47. case '10':   48. p("Fecha 1: %i de Octubre de %i\n",vec1,sum);   49. break;   50. case '11':   51. p("Fecha 1: %i de Noviembre de %i\n",vec1,sum);   52. break;   53. case 12:   54. p("Fecha 1: %i de Dicembre de %i\n",vec1,sum);   55. break;   56. }}   57. if(sum2<13){   58. switch (sum2){   59. case '1':   60. p("Fecha 1: %i de Enero de %i\n",vec2,sum);   61. break ;   62. case '2':   63. p("Fecha 1: %i de Feb de %i\n",vec2,sum);   64. break;   65. case '3':   66. p("Fecha 1: %i de Marzo de %i\n",vec2,sum);   67. break;   68. case '4':   69. p("Fecha 1: %i de Abril de %i\n",vec2,sum);   70. break;   71. case '5':   72. p("Fecha 1: %i de Mayo de %i\n",vec2,sum);   73. break;   74. case '6':   75. p("Fecha 1: %i de Junio de %i\n",vec2,sum);   76. break;   77. case '7':   78. p("Fecha 1: %i de Julio de %i\n",vec2,sum);   79. break;   80. case '8':   81. p("Fecha 1: %i de Agosto de %i\n",vec2,sum);   82. break;   83. case '9':   84. p("Fecha 1: %i de Septiembre de %i\n",vec2,sum);   85. break;   86. case '10':   87. p("Fecha 1: %i de Octubre de %i\n",vec2,sum);   88. break;   89. case '11':   90. p("Fecha 1: %i de Noviembre de %i\n",vec2,sum);   91. break;   92. case 12:   93. p("Fecha 1: %i de Dicembre de %i\n",vec2,sum);   94. break;   95. }}   96. p("\n%s\n",vec1);   97. p("\n %c %c\n",sum,sum2);   98. p("Fecha 1: %i de Dicembre de %i\n",vec2,sum);   99. } |

|  |
| --- |
| EJERCICIO 3  Implementar el algoritmo que permita leer una matriz de dos dimensiones, como una cadena utilizando los símbolos [ ], para indicar el inicio y terminación de la matriz y el de cada línea. Considere valores enteros con signo.  (3 puntos)  Ejemplo  Estribe la matriz: [ [ 1 , 2] , [ 3, 4] ]   1. 2   3 4 |
| [CÓDIGO] |

|  |
| --- |
| EJERCICIO 4  Escribir un programa que lea las dimensiones de una matriz (n x m ) y la llene la matriz con números enteros desde 0, con un incremento de 3 , después calcular la transpuesta de la matriz .  Sea ***A***  una matriz con ***m*** filas y ***n*** columnas. La matriz transpuesta, denotada con ***At***  Está dada por: (A^t)_{ij} = A_{ji},\ 1\le i\le n,\ 1\le j\le m  En donde el elemento ***aji*** de la matriz original ***A*** se convertirá en el elemento ***aij***  de la matriz transpuesta ***At***  Nota : Validar para matrices rectangulares.  (3 puntos) |
| [CÓDIGO] |

Conclusiones:

Bueno, pues se me complico la practica y aquí me doy cuenta que necesito practicar y estudiar bastante, solo pude hacer correctamente el primero, el segundo lo intenté con la instrucción switch y no me salió, y hasta el final pensé que era mejor con arreglos pero ya no me daba tiempo. Bueno no tengo nada más que decir más que ponerme a estudiar.