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

Laboratorios de computación salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodríguez Espino |
| *Asignatura:* | Fundamentos en programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 3 |
| *Integrante(s):* | Santos Martínez Daniela |
|  |  |
|  |  |
| *Semestre:* | Primero |
| *Fecha de entrega:* | 18-agosto-2017 |
| *Obervaciones:* |  |

Guía práctica de estudio 11: Arreglos unidimensionales y multidimensionales

Objetivo:

Reconocer la importancia y utilidad de los arreglos, en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo, así como trabajar con arreglos tanto unidimensionales como multidimensionales.

Actividades:

Primero se leyó la práctica, explicando cada ejemplo y explicando diferentes situaciones como la declaración de apuntadores (\*) y la dirección de memoria de un apuntador (&). También se vieron ejemplos utilizando cada uno y la diferencia entre

int a = 5, b = 10, c[10] = {5, 4, 3, 2, 1, 9, 8, 7, 6, 0};

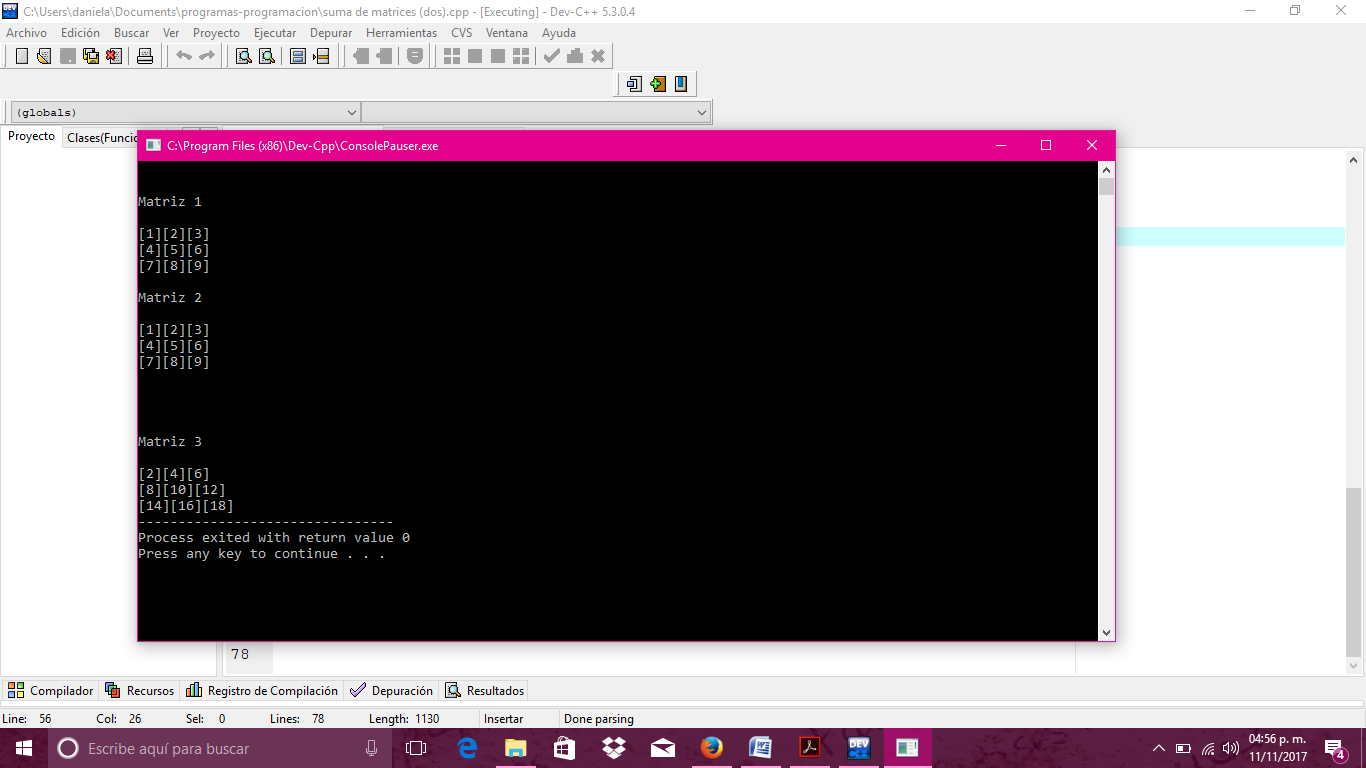
int \*apEnt;

X=\*(apArr)+1 que nos daría de resultado 6

X=\*(apArr+1) que nos daría de resultado 4

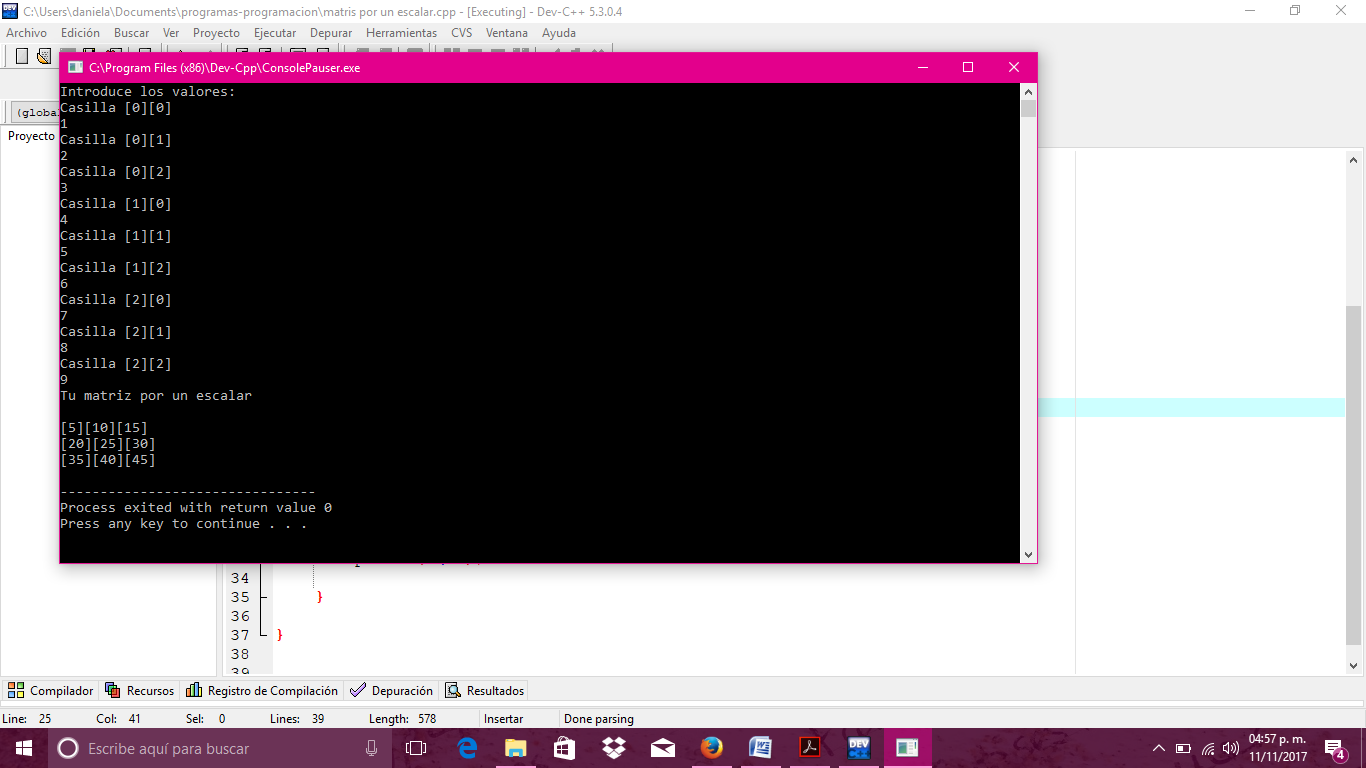
Programa 1: Suma de matrices 3x3

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  #include<conio.h>  int i,mat1[5][5],mat2[5][5],mat3[5][5],a,b;  main()  {  printf("Matriz 1");  for(a=1;a<=3;a++)  {  for(b=1;b<=3;b++)  {  printf("\n\nDame el numero [%d][%d]\n", a,b);  scanf("%d", &mat1[a][b]);  }    }  system("cls");  printf("Matriz 2");  for(a=1;a<=3;a++)//filas  {  for(b=1;b<=3;b++)//columnas  {  printf("\n\nDame el numero [%d][%d]\n", a,b);  scanf("%d", &mat2[a][b]);    }    }  system("cls");  printf("\n\nMatriz 1\n");  for(a=1;a<=3;a++)  {  printf("\n");  for(b=1;b<=3;b++)  {  printf("[%d]",mat1[a][b]);  }    }  printf("\n\nMatriz 2\n");  for(a=1;a<=3;a++)  {  printf("\n");  for(b=1;b<=3;b++)  {  printf("[%d]",mat2[a][b]);  }      }  for(a=1;a<=3;a++)  {  printf("\n");  for(b=1;b<=3;b++)  {  mat3[a][b]=(mat1[a][b] + mat2[a][b]);  }  }  printf("\n\nMatriz 3\n");  for(a=1;a<=3;a++)  {  printf("\n");  for(b=1;b<=3;b++)  {  printf("[%d]",mat3[ a ] [ b ]);  }  }  } |



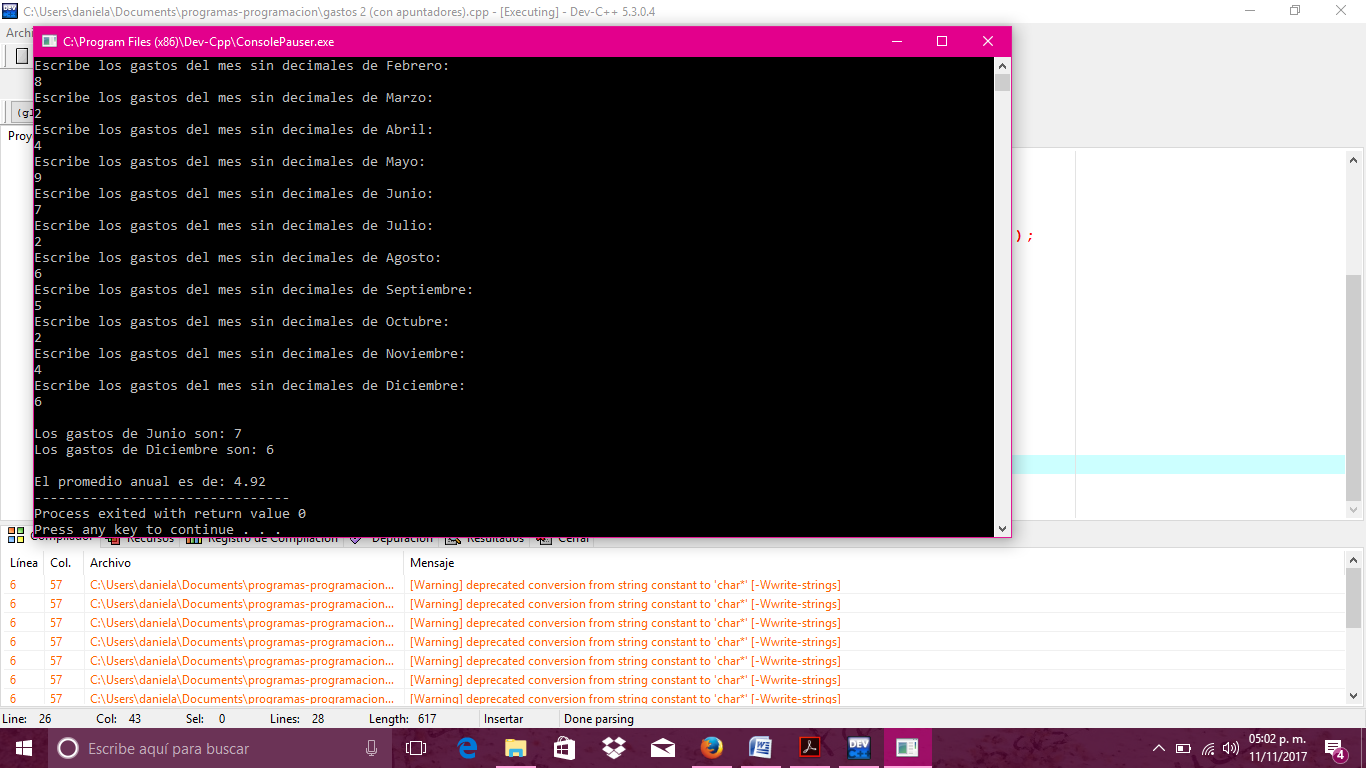
Programa 2: Matriz por un escalar

|  |
| --- |
| #include<stdio.h>  #include<stdlib.h>  int mat1[3][3],mat2[3][3], i,a,b,num;  main()  {  printf("Dame el numero por el que quieres multiplicar tu matriz\n");  scanf("%d",&num);  system("cls");    printf("Introduce los valores:\n");    for(a=0; a<=2; a++)  {  for(b=0;b<=2;b++)  {  printf("Casilla [%d][%d]\n", a,b);  scanf("%d", &mat1[a][b]);  }  }    for(a=0; a<=2; a++)  {  for(b=0;b<=2;b++)  {  mat2[a][b]=mat1[a][b]\*num;  printf("[%d]",mat2[a][b]);  }  printf("\n");  }  } |



Programa 3: Gastos mensuales con apuntador

|  |
| --- |
| //gastos mensuales  #include<stdio.h>  #include<conio.h>  char \*mes[15]={"Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio",  "Agosto","Septiembre","Octubre","Noviembre","Diciembre",};  int sum=0,i,prom, op[15];  int main()  {  for(i=0;i<=11;i++)  {  printf("Escribe los gastos del mes sin decimales de %s:\n", mes[i]);  scanf("%d", &op[i]);  }    printf("\nLos gastos de %s son: %d\n",mes[5], op[5]);  printf("Los gastos de %s son: %d\n", mes[11], op[11]);    for(i=0;i<=11;i++)  {  sum=sum+op[i];  }  float prom= (float)sum/12;  printf("\nEl promedio anual es de: %.2f",prom);  } |



Conclusiones:

Como conclusión podemos decir que el uso de apuntadores y arreglos nos pueden facilitar en varias ocasiones la elaboración del programa. Puede llegar a confundir el uso de apuntadores, más al hacer operaciones con el valor apuntado, pero con la práctica se facilitara la aplicación.