



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: **Manuel Enrique Castañeda Castañeda**

Asignatura:

Estructura de Datos y Algoritmos I

Grupo:

12

No de Práctica(s):

01 “Aplicaciones de Arreglos”

Integrante(s):

**Lemus Ambrosio Aline Andrea
Reyes Fuentes José Manuel
Wong Sánchez Yibran Lee
Sánchez Alvirde Andrés Iván**

*No. de Equipo de cómputo
empleado:*

-

No. de Lista o Brigada:

12

Semestre:

2021-2

Fecha de entrega:

12 de Marzo de 2021

Observaciones:

Calificación:

Estructura de Datos y Algoritmos I

Practica 01

Aplicaciones de Arreglos

INTRODUCCIÓN

Para comenzar a realizar esta práctica debemos considerar lo que es un arreglo y sus diferentes formas tanto lo que es un arreglo unidimensional como un arreglo multidimensional.

Comencemos definiendo lo que es un arreglo:

Un arreglo es una estructura, es decir es un conjunto de variables que se citan y manejan con un mismo nombre, y que permite además la utilización individual de sus elementos.

permite manipular datos de manera muy flexible. Combinándolas y anidándolas, es posible organizar información de manera estructurada para representar sistemas del mundo real. Más importante que la organización de los datos es la capacidad de hacer muchas operaciones a la vez sobre grandes conjuntos de datos numéricos de manera eficiente.

Un arreglo se puede ver como un conjunto de espacios finitos donde se almacenan elementos (todos del mismo tipo). Un arreglo también puede verse como cajas ordenadas en fila y numeradas, donde en cada caja se almacena un solo elemento u objeto.

Arreglos unidimensionales

Un arreglo se define como una colección finita, homogénea y ordenada de elementos

Es decir, la posición dentro del arreglo como se ha definido antes pero se caracteriza por ser unidimensional porque tiene una sola dimensión como se muestra a continuación.

Un arreglo tiene la característica de que puede almacenar a N elementos del mismo tipo y además permite el acceso a cada uno de estos elementos.

Se distinguen dos partes en los arreglos:

- Los componentes. Hacen referencia a los elementos que componen o forman el arreglo.

Es decir, son los valores que se almacenan en cada una de sus casillas.

- Los índices. Son los que permiten acceder a los componentes del arreglo en forma individual.

Arreglos multidimensional

Los arreglos multidimensionales tienen más de una dimensión. En C#, las dimensiones se manejan por medio de un par de corchetes, dentro de los que se escriben los valores de cada dimensión, separados por comas.

Los Arreglos multidimensionales no están limitados a dos índices (dos dimensiones). Pueden contener tantos índices como sea necesario, aunque es raro tener que representar más de 3 dimensiones. Solo considere la cantidad de memoria que un arreglo con muchos índices puede requerir. Por ejemplo:

```
char century [100][365][24][60][60];
```

asigna un char para cada segundo contenido en un siglo, que es más de 3 mil millones de chars! Esto consumiría cerca de 3000 megabytes de memoria RAM si pudiéramos declararlo. Los arreglos multidimensionales no son más que una abstracción, ya que podemos obtener los mismos resultados con un simple arreglo poniendo un factor entre sus índices

Los arreglos multidimensionales son una ampliación de las matrices de dos dimensiones y utilizan subíndices adicionales para la indexación. Un arreglo 3D, por ejemplo, utiliza tres subíndices. Los dos primeros son como una matriz, pero la tercera dimensión representa páginas u hojas de elementos.

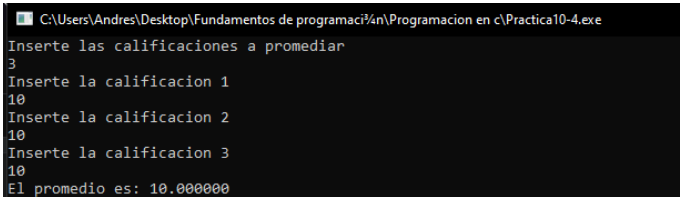
DESARROLLO

1. Almacenar los números pares de un arreglo y los impares en otro

CÓDIGO FUENTE	EJECUCIÓN
<pre>#include <stdio.h> #define p printf #define s scanf int main() { int n, A[n], i, k, P[i], r, l[r]; p("\n\n\t Dime el tamaño del arreglo "); s("%d", &n); for(int i=0; i<n; i++) { p(" Dame el dato [%d]", i+1); s("%d", &A[i]); } p("El arreglo ingresado fue: "); for(i=0; i<n; i++) { p("\n %d ", A[i]); } //obtener pares p("\n El arreglo de pares es: \n"); for(i=0; i<n; i++) { if(A[i]%2==0) { P[k]=A[i]; p("\n %d", P[k]); } } //obtener impares p("\n El arreglo de impares es: \n"); for(i=0; i<n; i++) { if(A[i]%2!=0)</pre>	<pre>Dame el dato [9]9 Dame el dato [10]10 El arreglo ingresado fue: 1 2 3 4 5 6 7 8 9 10 El arreglo de pares es: 2 4 6 8 10 El arreglo de impares es: 1 3 5 7 9</pre>

<pre> { l[r]=A[i]; p("\n %d", l[r]); } } }</pre>	
--	--

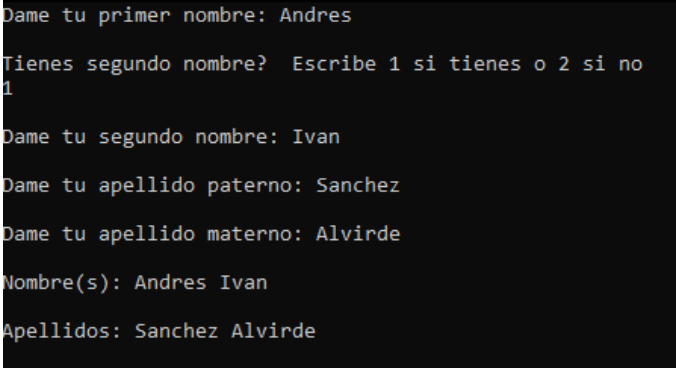
2. Obtener el promedio de n calificaciones almacenadas en un arreglo

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> int main() { int numeroCalificaciones; int i; float promedio = 0; float calificacion; printf("Inserte las calificaciones a promediar\n"); scanf("%d",&numeroCalificaciones); for(i = 1; i <= numeroCalificaciones; i++) { printf("Inserte la calificacion %d\n",i); scanf("%f",&calificacion); promedio+=calificacion; } promedio = promedio/numeroCalificaciones; printf("El promedio es: %f",promedio); return 0; }</pre>	 <pre> C:\Users\Andres\Desktop\Fundamentos de programaci4n\Programacion en c\Practica10-4.exe Inserte las calificaciones a promediar 3 Inserte la calificacion 1 10 Inserte la calificacion 2 10 Inserte la calificacion 3 10 El promedio es: 10.000000</pre>

3. Contar el número de vocales en una frase

CÓDIGO FUENTE	EJECUCIÓN
<pre>#include <stdio.h> #include <ctype.h> int contarVocales(char *cadena); int main(int argc, char const *argv[]) { char entrada[1000]; printf("Escribe una frase:\n"); gets(entrada); int vocales = contarVocales(entrada); printf("El numero de vocales que tiene tu frase es: %d\n", vocales); return 0; } int contarVocales(char *cadena){ int vocales = 0; for (int indice = 0; cadena[indice] != '\0'; ++indice){ char letraActual = tolower(cadena[indice]); if (letraActual == 'a' letraActual == 'e' letraActual == 'i' letraActual == 'o' letraActual == 'u') { vocales++; } } return vocales; }</pre>	

4. Leer del teclado tu nombre y apellido, almacenar cada palabra en un arreglo

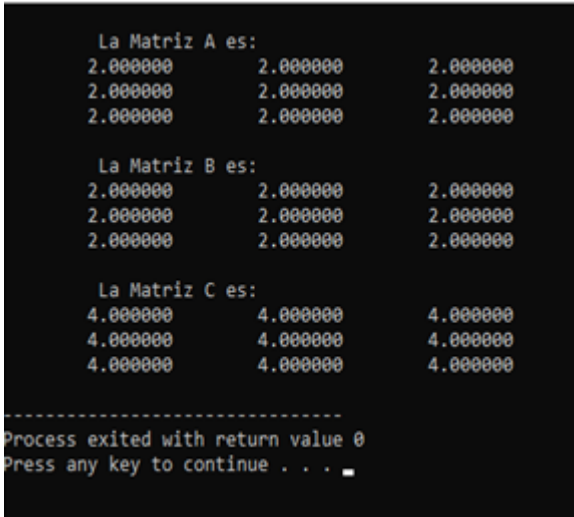
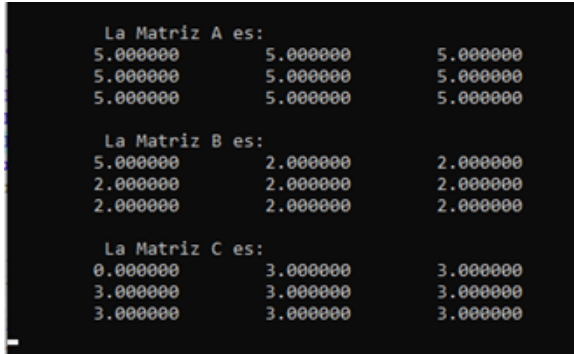
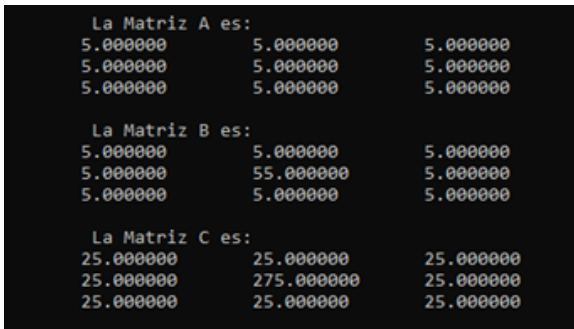
CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> #include <string.h> int main() { char Nombre1[1][20],Nombre2[1][20],Apellido1[1][20], Apellido2[1][20]; int r; for(int i=0;i<1;i++) { printf("Dame tu primer nombre: "); fflush(stdin); gets(Nombre1[i]); } printf("\nTienes segundo nombre\? "); printf(" Escribe 1 si tienes o 2 si no\n"); scanf("%d",&r); if(r==1) { for(int i=0;i<1;i++) { printf("\nDame tu segundo nombre: "); fflush(stdin); gets(Nombre2[i]); } } for(int i=0;i<1;i++) { printf("\nDame tu apellido paterno: "); fflush(stdin); gets(Apellido1[i]); } for(int i=0;i<1;i++) { printf("\nDame tu apellido materno: "); fflush(stdin); gets(Apellido2[i]); } for(int i=0;i>=0;i--) { </pre>	 <pre> Dame tu primer nombre: Andres Tienes segundo nombre? Escribe 1 si tienes o 2 si no 1 Dame tu segundo nombre: Ivan Dame tu apellido paterno: Sanchez Dame tu apellido materno: Alvirde Nombre(s): Andres Ivan Apellidos: Sanchez Alvirde </pre>


```

        printf("\nNombre(s): %s\n", Nombre1[i], Nombre2[i]);
        printf("\nApellidos: %s\n", Apellido1[i], Apellido2[i]);
    }
    return 0;
}

```

5. Calculadora de matrices

CÓDIGO FUENTE	EJECUCIÓN
<pre>#include<stdio.h> #include<conio.h> #include<stdlib.h> main() { int n; printf("¿Qu%c opci%cn desea?", 138,162); printf("\n1)Suma de matrices"); printf("\n2)Resta de matrices"); printf("\n3)Multiplicaci%cn de matrices", 162); printf("\n4)Divisi%cn de matrices", 162); printf("\nDame una opcion: "); scanf("%d",&n); switch (n) { case 1: int i,j; //Se define las variables de los arreglos float A[3][3], B[3][3], C[3][3]; //Se definen los arreglos printf("Ingresa los datos de la primera matriz"); for(i=0; i<=2; i++) //Ciclo para llenar la columna { for(j=0; j<=2;j++)//Ciclo para llenar la fila { printf("\n Teclear el elemento %d %d de la matriz \t", i,j); //Se ingresan los datos del arreglo scanf("%f",&A[i][j]); //Se guardan los datos } } }</pre>	<div></div> <div></div> <div></div>

```

system("cls"); //Limpia la pantalla
printf("Ingresa los datos de la segunda
matriz");
for(i=0; i<=2; i++) //Ciclo para llenar la
columna
{
    for(j=0; j<=2;j++)//Ciclo para llenar la fila
    {
        printf("\n Teclear el elemento %d %d de
la matriz \t", i,j);
        //Se ingresan los datos del arreglo
        scanf("%f",&B[i][j]); //Se guardan los datos
    }
}
system("cls"); //Limpia la pantalla
printf("\n \t La Matriz A es:\n"); //Impresion de
la matriz A
for(i=0; i<=2; i++)
{
    for(j=0; j<=2;j++)
    {
        printf("\t%f",A[i][j]); //Imprime los
resultados datos de salida
    }
    printf("\t\n");
}
printf("\n \t La Matriz B es:\n");//Impresion de
la matriz B
for(i=0; i<=2; i++)
{
    for(j=0; j<=2;j++)
    {
        printf("\t%f",B[i][j]);//Imprime los
resultados datos de salida
    }
    printf("\t\n");
}
for(i=0; i<=2; i++)
{
    for(j=0; j<=2;j++)
    {
        C[i][j]=A[i][j]+B[i][j];
//Operacion que suma las matrices
    }
}

printf("\n \t La Matriz C es:\n");//Impresion de
la matriz C
for(i=0; i<=2; i++)

```

```

La Matriz A es:
2.000000    2.000000    2.000000
2.000000    2.000000    2.000000
2.000000    2.000000    2.000000

La Matriz B es:
2.000000    2.000000    2.000000
2.000000    2.000000    2.000000
2.000000    2.000000    2.000000

La Matriz C es:
1.000000    1.000000    1.000000
1.000000    1.000000    1.000000
1.000000    1.000000    1.000000

```

```

        {
            for(j=0; j<=2;j++)
            {
                printf("\t%f",C[i][j]);//Imprime los
resultados datos de salida
            }
            printf("\t\n");
        }
        break;

        case 2:
            int k,l; //Se define las variables de los
arreglos
            float D[3][3], E[3][3], F[3][3]; //Se definen los
arreglos
            printf("Ingresa los datos de la primera
matriz");
            for(k=0; k<=2; k++) //Ciclo para llenar la
columna
            {
                for(l=0; l<=2;l++)//Ciclo para llenar la fila
                {
                    printf("\n Teclear el elemento %d %d de
la matriz \t", k,l);
                    //Se ingresan los datos del arreglo
                    scanf("%f",&D[k][l]); //Se guardan los
datos
                }
            }
            system("cls"); //Limpia la pantalla
            printf("Ingresa los datos de la segunda
matriz");
            for(k=0; k<=2; k++) //Ciclo para llenar la
columna
            {
                for(l=0; l<=2;l++)//Ciclo para llenar la fila
                {
                    printf("\n Teclear el elemento %d %d de
la matriz \t", k,l);
                    //Se ingresan los datos del arreglo
                    scanf("%f",&E[k][l]); //Se guardan los
datos
                }
            }
            system("cls"); //Limpia la pantalla
            printf("\n \t La Matriz A es:\n"); //Impresion de
la matriz A
            for(k=0; k<=2; k++)
            {

```

```

        for(l=0; l<=2;l++)
        {
            printf("\t%f",D[k][l]);    //Imprime
los resultados datos de salida
        }
        printf("\t\n");
    }
    printf("\n \t La Matriz B es:\n");//Impresion de
la matriz B
    for(k=0; k<=2; k++)
    {
        for(l=0; l<=2;l++)
        {
            printf("\t%f",E[k][l]);//Imprime los
resultados datos de salida
        }
        printf("\t\n");
    }
    for(k=0; k<=2; k++)
    {
        for(l=0; l<=2;l++)
        {
            F[k][l]=D[k][l]-E[k][l];
//Operacion que suma las matrices
        }
    }

    printf("\n \t La Matriz C es:\n");//Impresion de
la matriz C
    for(k=0; k<=2; k++)
    {
        for(l=0; l<=2;l++)
        {
            printf("\t%f",F[k][l]);//Imprime los
resultados datos de salida
        }
        printf("\t\n");
    }
    break;

    case 3:
    int p,u; //Se define las variables de los
arreglos
    float X[3][3], Y[3][3], Z[3][3]; //Se
definen los arreglos
    printf("Ingresa los datos de la primera
matriz");
    for(p=0; p<=2; p++) //Ciclo para llenar
la columna

```

```

{
for(u=0; u<=2;u++)//Ciclo para llenar la fila
{
printf("\n Teclear el elemento %d %d de
la matriz \t", p,u);
//Se ingresan los datos del arreglo
scanf("%f",&X[p][u]); //Se guardan los
datos
}
}

system("cls"); //Limpia la pantalla
printf("Ingresa los datos de la segunda
matriz");
for(p=0; p<=2; p++) //Ciclo para llenar
la columna
{
for(u=0; u<=2;u++)//Ciclo para llenar la fila
{
printf("\n Teclear el elemento %d %d de
la matriz \t", p,u);
//Se ingresan los datos del arreglo
scanf("%f",&Y[p][u]); //Se guardan los
datos
}
}

system("cls"); //Limpia la pantalla
printf("\n \t La Matriz A es:\n");
//Impresion de la matriz A
for(p=0; p<=2; p++)
{
for(u=0; u<=2;u++)
{
printf("\t%f",X[p][u]); //Imprime
los resultados datos de salida
}
printf("\t\n");
}
printf("\n \t La Matriz B es:\n");//Impresion de
la matriz B
for(p=0; p<=2; p++)
{
for(u=0; u<=2;u++)
{
printf("\t%f",Y[p][u]);//Imprime
los resultados datos de salida
}
printf("\t\n");
}
for(p=0; p<=2; p++)

```

```

        {
            for(u=0; u<=2;u++)
            {
                Z[p][u]=X[p][u]*Y[p][u];
//Operacion que suma las matrices
            }
        }

printf("\n \t La Matriz C es:\n");//Impresion de
la matriz C
for(p=0; p<=2; p++)
{
    for(u=0; u<=2;u++)
    {
        printf("\t%f",Z[p][u]);//Imprime
los resultados datos de salida
    }
    printf("\t\n");
}
break;
case 4:
    int o,n; //Se define las variables de los
arreglos
    float G[3][3], H[3][3], I[3][3]; //Se definen los
arreglos
    printf("Ingresa los datos de la primera
matriz");
    for(o=0; o<=2 ; o++) //Ciclo para llenar la
columna
    {
        for(n=0; n<=2;n++)//Ciclo para llenar la fila
        {
            printf("\n Teclear el elemento %d %d de
la matriz \t", o,n);
            //Se ingresan los datos del arreglo
            scanf("%f",&G[o][n]); //Se guardan los
datos
        }
    }
    system("cls"); //Limpia la pantalla
    printf("Ingresa los datos de la segunda
matriz");
    for(o=0; o<=2; o++) //Ciclo para llenar la
columna
    {
        for(n=0; n<=2;n++)//Ciclo para llenar la fila
        {
            printf("\n Teclear el elemento %d %d de
la matriz \t", o,n);

```

```

        //Se ingresan los datos del arreglo
        scanf("%f",&H[o][n]); //Se guardan los
datos
    }
}
system("cls"); //Limpia la pantalla
printf("\n \t La Matriz A es:\n"); //Impresion de
la matriz A
for(o=0; o<=2; o++)
{
    for(n=0; n<=2;n++)
    {
        printf("\t%f",G[o][n]); //Imprime
los resultados datos de salida
    }
    printf("\t\n");
}
printf("\n \t La Matriz B es:\n");//Impresion de
la matriz B
for(o=0; o<=2; o++)
{
    for(n=0; n<=2;n++)
    {
        printf("\t%f",H[o][n]);//Imprime
los resultados datos de salida
    }
    printf("\t\n");
}
for(o=0; o<=2; o++)
{
    for(n=0; n<=2;n++)
    {
        I[o][n]=G[o][n]/H[o][n];
//Operacion que suma las matrices
    }
}

printf("\n \t La Matriz C es:\n");//Impresion de
la matriz C
for(o=0; o<=2; o++)
{
    for(n=0; n<=2;n++)
    {
        printf("\t%f",I[o][n]);//Imprime los
resultados datos de salida
    }
    printf("\t\n");
}
break;

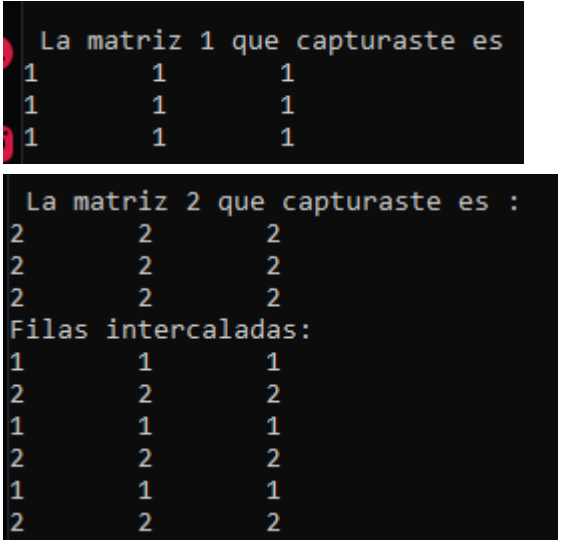
```

```

default:
    printf("Error");
}
getch();
}

```

6. A partir de dos matrices formar una tercera, se deben intercalar las filas

CÓDIGO FUENTE	EJECUCIÓN
<pre> #include <stdio.h> #include <stdlib.h> main() { int filas,columnas; int temporal; printf("Cuantas columnas tendra tu primera matriz? "); scanf("%d",&columnas); printf("Cuantas filas tendra tu primera matriz? "); scanf("%d",&filas); printf("\n Ingrese los valores de la matriz 1:\n"); int array[filas][columnas]; for(int i=0; i<filas; i++) { for(int j=0; j<columnas; j++) { scanf("%d",&array[i][j]); } } printf("\n La matriz 1 que capturaste es :\n"); for(int i=0;i<filas;i++) { for(int j=0;j<columnas;j++) { printf("%d \t",array[i][j]); } printf("\n"); } int filas2,columnas2; </pre>	 <pre> La matriz 1 que capturaste es 1 1 1 1 1 1 1 1 1 La matriz 2 que capturaste es : 2 2 2 2 2 2 2 2 2 Filas intercaladas: 1 1 1 2 2 2 1 1 1 2 2 2 1 1 1 2 2 2 </pre>


```

        printf("\n Cuantas columnas tendra tu
segunda matriz? ");
        scanf("%d",&columnas2);
        printf("Cuantas filas tendra tu segunda
matriz? ");
        scanf("%d",&filas2);
        printf("\n Ingrese los valores de la matriz
2:\n");
        int array2[filas2][columnas2];
        for(int k=0; k<filas2; k++)
        {
            for(int o=0; o<columnas2; o++)
            {
                scanf("%d",&array2[k][o]);
            }
        }
        printf("\n La matriz 2 que capturaste es
:\n");
        for(int k=0;k<filas2;k++)
        {
            for(int o=0;o<columnas2;o++)
            {
                printf("%d\t",array2[k][o]);
            }
            printf("\n");
        }
        int temp=0,temp2=0;
        printf("Filas intercaladas:\n");
        while(temp<columnas &&
temp2<columnas2)
        {
            for(int j=0;j<columnas;j++)
            {
                printf("%d\t",array[temp][j]);
            }temp++;
            printf("\n");
            for(int o=0;o<columnas2;o++)
            {
                printf("%d\t",array2[temp2][o]);
            }temp2++;
            printf("\n");
        }
        printf("\n");
        return 0;
    }

```

OBSERVACIONES (individuales)

- **Lemus Ambrosio Aline Andrea:**

En esta práctica el primer ejercicio nos costó un poco, ya que no sabíamos como bien cómo desarrollar el programa luego de pedirle al usuario que nos dijera de qué tamaño sería el arreglo, algunas cosas las pudimos resolver en equipo, y lo que no lo tuvimos que investigar por nuestra cuenta ya sea viendo videos o revisando ejercicios anteriores, de esta forma pudimos resolver el ejercicio. Para el ejercicio tres ya no me acordaba bien de cómo debía empezar a desarrollar el ejercicio, por lo que tuve que ver algunos apuntes y buscar en videos como se podían realizar ejercicios similares, gracias a esto fui comprendiendo que pasos seguir y para que me servía cada paso y las bibliotecas usadas, así fue como pude resolver este ejercicio.

- **Reyes Fuentes José Manuel:**

Los problemas que se suscitaron a lo largo de esta práctica como lo fue: los arreglos por su tamaños, que no registraba todos los datos y nos arrojaba datos en la memoria o símbolos extraños; fueron resueltos gracias a la ayuda del profesor y algunos ejercicios hechos en la anterior asignatura que fue fundamentos de programación e incluso videos en youtube que nos mencionaba en que estábamos mal.

Claro también en algunos programas decidimos darle unos pequeños arreglos para que fuera más agradable a la vista del usuario como acentos y la letra "ñ" que se tuvieron que usar codigos y este símbolo "%c" de carácter.

- **Wong Sánchez Yibran Lee:**

En la parte en la que debíamos pedirle al usuario el tamaño del arreglo y cada uno de los elementos el syntaxis nos falló, otro de los problemas presentados fue el uso de ciclos para los arreglos pero investigando y usando ejemplos hechos en fundamentos logramos desarrollar los ejercicios.

Muchos de los problemas los resolvimos con facilidad en equipo, lo que ayuda demasiado,

- **Sánchez Alvirde Andrés Iván:**

Al principio el ejercicio que hicimos en equipo costó ya que no encontrábamos donde estaba el error que impedía que se ejecutara como queríamos , le solicitamos ayuda al profesor y él hizo señalizaciones de errores que teníamos ya que queríamos imprimir dos valores pero en realidad solo tenía que ser una, por lo que después de eso fue corregido, el profesor se fue y comenzamos a desarrollar ya el ejercicio ya que por fin nos quitamos ese problema que teníamos y así logramos acabar el ejercicio.

Luego de eso acabo la clase y pues por mi cuenta decidí realizar algunos ejercicios que me costaron como el 6 ya que no encontraba el modo de cómo imprimir de manera intercalada cada fila de ambos arreglos, consulte webs intentando buscar algo para guiarme hasta que encontré uno que explicaba cómo imprimir una sola fila del arreglo, en ese mismo momento decidí probar con eso de manera individual en cada arreglo, viendo que servía ahora el problema era hacer que se imprimieran las restantes, entre tiempo pensando se me ocurrió realizar un un ciclo con while, cuando probé lo hice con una sola fila del mismo arreglo, vi que funciona así que decidí complementarlo

con el segundo arreglo y logre conseguir que se imprimen de manera intercalada .

CONCLUSIONES (individuales)

- **Lemus Ambrosio Aline Andrea:**

El ejercicio que resolví por mi cuenta y el resuelto en equipo los pude comprender y pese a que tenía dudas o me llegaba a confundir viendo videos y escuchando las soluciones de mis compañeros fue como pude comprender como es que se debían de resolver los ejercicios, por lo que en esta práctica comprendí mejor a como usar los arreglos

- **Reyes Fuentes José Manuel:**

Aprendí a usar los arreglos como una herramienta que se pueden guardar valores tanto como letras y números, no solo eso también poder hacer algunas funciones con los mismos como lo fue una calculadora con un arreglo multidimensional que es conocido como una “matriz”, quizás se le pueda dar una utilidad a la hora de hacer cálculos en algunas materias. Con los conocimientos anteriores también aprendí a hacer un juego como lo fue el ahorcado y ahora en este semestre se tuvo la oportunidad de hacer una lotería que es para poder divertirse aprendiendo a como programar, realmente ha sido fascinante como podemos manipular los datos de la memoria del ordenador.

- **Wong Sánchez Yibran Lee:**

Pude concluir que los arreglos son de gran utilidad ya que son cadenas de caracteres o datos que nos permite almacenar información de diferentes tipos, También aprendí el manejo de información, acerca de los bits y bytes, también me ayudo demasiado el trabajo en equipo ya que nos apoyamos de muy buena manera y logramos resolver los ejercicios.

- **Sánchez Alvirde Andrés Iván:**

Para concluir quiero decir que refuerce los conocimientos en usos de arreglos e incluso aprendí a optimizarlos un poco más y también aprendí nuevas funciones como la generación de números aleatorios cosa que en primer semestre solo lo vimos pero en realidad nunca tuvimos que usarlo en su totalidad pero aquí aprendí a ver como se le podía dar un uso.