



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: M.I. Marco Antonio Martínez Quintana.

Asignatura: Estructura de datos y algoritmos I

Grupo: 15

No de Práctica(s): 1

Integrante(s): Moreno Razo Laura Mildred

*No. de Equipo de
cómputo empleado:* No aplica

No. de Lista o Brigada: No aplica

Semestre: 2021-2

Fecha de entrega: 15 de Marzo del 2021

Observaciones:

CALIFICACIÓN: _____

Objetivo

Utilizar arreglos unidimensionales y multidimensionales para dar solución a problemas computacionales.

Introducción

Un arreglo es un conjunto de datos del mismo tipo (enteros, flotantes, caracteres, etc.) que se almacenan de manera consecutiva y con una posición específica en la memoria. Para acceder a este espacio se utilizan índices dependiendo de la manera en que estén organizados.

Arreglos unidimensionales. La colección de datos esta ordenada en una sola dimensión, es decir podemos imaginarla como una fila o una columna con n elementos.

Arreglos multidimensionales. Poseen dos o más dimensiones, dentro de las cuales se guardan los datos.

Actividad 1. La escítala espartana

Código comentado

```
#include<stdio.h>
/*
Programa que realiza la implementación de la escítala espartana
Para cifrar y descifrar.
*/
//firma de las funciones cifrar y descifrar.
void crearMensaje();
void descifrarMensaje();
int main(){
    short opcion=0;

    //Menú utilizando while mientras el parámetro sea 1
    while (1){
        printf("\n\t*** ESCÍTALA ESPARTANA ***\n",214);
        printf("%cQu%c desea realizar?\n",168,130);
        printf("1) Crear mensaje cifrado.\n");
        printf("2) Descifrar mensaje.\n");
        printf("3) Salir.\n");
        scanf("%d", &opcion);
        switch(opcion){
            case 1:
                //invoca a la función para cifrar
                crearMensaje();
                break;
            case 2:
                //invoca a la función para descifrar
                descifrarMensaje();
                break;
            case 3:
```

```

        //El parámetro cambia a 0 por lo que termina el ciclo while
        return 0;
    default:
        printf("Opción no válida.\n");
    }
}
return 0;
}

void crearMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escitela:\n",164,161);
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);
    //Crea la matriz
    char escitala[ren][col];
    //Crea una cadena de caracteres con el tamaño de elementos de la matriz
    char texto[ren*col];
    printf("Escriba el texto a cifrar:\n");
    scanf("%s", texto);
    /*Mediante un bucle for anidado accede a las posiciones de la matriz e
    introduce los elementos del vector texto */
    for (i=0 ; i<ren ; i++)
        for (j=0 ; j<col ; j++)
            escitala[i][j] = texto[k++];

    printf("El texto en la tira queda de la siguiente manera:\n");

    /*Transpone la matriz y la imprime, invirtiendo los renglones y columnas para cambiar
    el orden de los elementos */
    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)
            printf("%c", escitala[j][i]);

    printf("\n");
}

//Realiza el mismo procedimiento que crearMensaje.
void descifrarMensaje(){
    int ren, col, i, j, k=0;
    printf("Ingresar el tamaño de la escitela:\n",164,161);
    printf("\nRenglones:");
    scanf("%i",&ren);
    printf("\nColumnas:");
    scanf("%i",&col);
    char escitala[ren][col];
    char texto[ren*col];
    printf("Escriba el texto a descifrar:\n");
    scanf("%s", texto);
    //Llena la matriz de arriba hacia abajo para trasponerla
    for (i=0 ; i<col ; i++)
        for (j=0 ; j<ren ; j++)
            escitala[j][i] = texto[k++];

    printf("El texto descifrado es:\n");
    for (i=0 ; i<ren ; i++)
        for (j=0 ; j<col ; j++)
            //Imprime la matriz transpuesta que contiene el mensaje original
            printf("%c", escitala[i][j]);
}

```

Prueba código

```
*** ESCITALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
1
Ingresar el tamaño de la escítala:

Renglones:5

Columnas:3
Escriba el texto a cifrar:
hoyesunbuendia
El texto en la tira queda de la siguiente manera:
heneiosbnayuud

*** ESCÍTALA ESPARTANA ***
¿Qué desea realizar?
1) Crear mensaje cifrado.
2) Descifrar mensaje.
3) Salir.
2
Ingresar el tamaño de la escítala:

Renglones:5

Columnas:3
Escriba el texto a descifrar:
heneiosbnayuud
El texto descifrado es:
hoyesunbuendia
```

Actividad 2. Sudoku

Código.

```
#include <stdio.h>
#include <conio.h>
void llenar ();

int casillas [9][9]= {{0,0,0,0,0,0,0,0,0},{5,0,0,7,1,0,4,9,6},{0,0,0,0,5,0,8,0,3}
,{0,0,5,0,4,0,2,6,0},{0,0,0,2,6,5,3,0,0},{8,0,0,0,0,1,5,0,0},{0,0,8,0,2,0,0,0,4},
{0,0,2,0,3,0,6,5,0},{0,9,0,5,8,7,0,0,0}};
int i,j,c,f,new=0;
int main (){
char res[1];

//accede a los elementos de la matriz para imprimirlos
for (i=0 ; i<9 ; i++){
for (j=0 ; j<9 ; j++){

printf("%d\t", casillas[i][j]);
//separa la cuadrícula horizontalmente
if (j== 2|| j==5){
printf("\t");
}

printf("\n");
//separa la cuadrícula verticalmente
if (i== 2|| i==5){
printf("\n");
}

//Permite cambiar valores,se ejecutara una vez y mientras la respuesta sea "s"
do{
printf("\n\t%cQue posici%cn desea llenar o cambiar%c \n",168,162,63 );
printf("Fila:");
scanf ("%d",&f);
printf("\nColumna:");

printf("\nColumna:");
scanf ("%d",&c);
printf("\nIngresa el valor \t");
scanf ("%d",&new);
//
switch(f) {
case 1:
llenar ();
break;

case 2:
//condicional anidada para evitar la modificacion de casillas predeterminadas
if (c!= 1&&4) {
if (c!= 5&&7){
if (c!=8&&9){
//invoca la funcion para llenar casillas
llenar();
}
}
}

else{
printf ("Esta casilla no se puede modificar");
}
break;

case 3:
if (c!= 5&&7) {
if (c!= 9){
llenar();
}
}

else{
printf ("Esta casilla no se puede modificar");
}
```

```

case 4:
if (c!= 3&&5) {
    if (c!= 7&&8){
        llenar();
    }
}

else{
    printf ("Esta casilla no se puede modificar");
}
break;

case 5:
if (c!= 4&&5) {
    if (c!= 6&&7){
        llenar();
    }
}

else{
    printf ("Esta casilla no se puede modificar");
}
break;

case 6:
if (c!= 1&&6) {
    if (c!= 7){
        llenar();
    }
}

else{
    printf ("Esta casilla no se puede modificar");
}
}

```

```

}
break;
case 8:
if (c!= 3&&5) {
    if (c!= 7&&8){
        llenar();
    }
}

else{
    printf ("Esta casilla no se puede modificar");
}
break;

case 9:
if (c!= 2&&4) {
    if (c!= 5&&6){
        llenar();
    }
}

else{
    printf ("Esta casilla no se puede modificar");
}
break;
default:
printf("\n Fila no valida");
}

printf("\n\tSeguir llenando s/n \n" );
res[0]= getch();

}while (res[0]=='s');

```

```

        }while (res[0]=='s');

return 0;
}

//funcion que asigna e imprime los nuevos valores
void llenar (){
casillas[f-1][c-1]=new;
printf("\n\t\t\tEl sudoku queda de la siguiente manera\n");

    for (i=0 ; i<9 ; i++){
        for (j=0 ; j<9 ; j++){

            printf("%d \t", casillas[i][j]);
            if (j== 2|| j==5){
                printf("\t");
            }
        }

        printf("\n");
        if (i== 2|| i==5){
            printf("\n");
        }
    }
}

```

El sudoku es un arreglo bidimensional de 9 x 9 que se despliega por medio de un ciclo for anidado en otro que imprime los elementos, además de dos condicionales if que hacen un salto de línea y una tabulación para poder dividirlo en 9 cuadrículas.

Primero solicita la fila y columna que se desea llenar, además del valor a insertar. Después vuelve a desplegar los valores con la posición actualizada y pregunta si deseamos continuar llenando casillas, si presionamos "s" vuelve a preguntar qué posición queremos cambiar, de lo contrario finaliza la ejecución.

```
C:\Users\mildr\Desktop\C>sudoku.exe
0 0 0 0 0 0 0 0 0
5 0 0 7 1 0 4 9 6
0 0 0 0 5 0 8 0 3

0 0 5 0 4 0 2 6 0
0 0 0 2 6 5 3 0 0
8 0 0 0 0 1 5 0 0

0 0 8 0 2 0 0 0 4
0 0 2 0 3 0 6 5 0
0 9 0 5 8 7 0 0 0

¿Que posición desea llenar o cambiar?
Fila:1
Columna:1
Ingresa el valor 4

El sudoku queda de la siguiente manera
4 0 0 0 0 0 0 0 0
5 0 0 7 1 0 4 9 6
0 0 0 0 5 0 8 0 3

0 0 5 0 4 0 2 6 0
0 0 0 2 6 5 3 0 0
8 0 0 0 0 1 5 0 0

0 0 8 0 2 0 0 0 4
0 0 2 0 3 0 6 5 0
0 9 0 5 8 7 0 0 0

Seguir llenando s/n
¿Que posición desea llenar o cambiar?
```

Si seleccionamos una de las posiciones con valores predeterminados se nos informa que no se pueden modificar.

Por ejemplo, en la imagen 1 intentamos cambiar el 5 en la fila 2, columna 1.

```

El sudoku queda de la siguiente manera
4      0      0      0      0      0      0      0      0
5      0      0      7      1      0      4      9      6
0      0      0      0      5      0      8      0      3

0      0      5      0      4      0      2      6      0
0      0      0      2      6      5      3      0      0
8      0      0      0      0      1      5      0      0

0      0      8      0      2      0      0      0      4
0      0      2      0      3      0      6      5      0
0      9      0      5      8      7      0      0      0

Seguir llenando s/n

¿Que posición desea llenar o cambiar?
Fila:2

Columna:1

Ingresa el valor      6
Esta casilla no se puede modificar
Seguir llenando s/n

```

Imagen 1.

Como se observa es posible colocar todos los valores necesarios para su resolución, presentada a continuación:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 4 | 6 | 1 | 8 | 9 | 3 | 7 | 2 | 5 |
| 5 | 8 | 3 | 7 | 1 | 2 | 4 | 9 | 6 |
| 2 | 7 | 9 | 6 | 5 | 4 | 8 | 1 | 3 |
| 1 | 3 | 5 | 9 | 4 | 8 | 2 | 6 | 7 |
| 9 | 4 | 7 | 2 | 6 | 5 | 3 | 8 | 1 |
| 8 | 2 | 6 | 3 | 7 | 1 | 5 | 4 | 9 |
| 3 | 5 | 8 | 1 | 2 | 6 | 9 | 7 | 4 |
| 7 | 1 | 2 | 4 | 3 | 9 | 6 | 5 | 8 |
| 6 | 9 | 4 | 5 | 8 | 7 | 1 | 3 | 2 |

Conclusión

Los arreglos son una herramienta muy útil para la solución de problemas mediante la programación, ya que nos permiten almacenar datos de manera ordenada y realizar operaciones con ellos. Una de sus aplicaciones en la vida cotidiana puede ser el almacenar una lista de calificaciones escolares dentro de un arreglo, donde se tenga el número de cuenta del alumno en la primera fila y sus calificaciones en las columnas. Dentro de la computación tiene cientos de usos, como el uso de transformación de matrices para el análisis de redes, otro ejemplo es la estadística donde podemos guardar valores en arreglos para después trabajar con estos y calcular moda, mediana, media etc.

En la práctica analizamos el procedimiento para la implementación de algunas de sus aplicaciones como lo es en criptografía básica o el almacenamiento de datos de un sudoku dentro de un arreglo de 2 dimensiones. Por lo tanto, se cumplió el objetivo.

Referencias

No utilizadas.