INGENIERIA EN COMPUTACION

Materia: Ingeniería en computación

Alumno: Ayon Dimas Andres

Matricula: 366845

Profesor: Pedro Nunes Yepiz

Actividad: #7

Tema: Cadenas y funciones

Ensenada B.C 25 de septiembre del 2023

INTRODUCCION

Aquí podremos encontrar lo que son las cadenas y las funciones, esto aplicando lo anterior de las practicas pasadas, como lo son los ciclos y las condiciones. La utilidad que tienen estas y la manera en la que se implementan y para qué sirven es lo que podremos ver y pudimos realizar en esta práctica, así como su uso y cuando deberían ser utilizadas. Pero esto tiene una particularidad, y es que es muy común utilizar el **strlen** para leer las cadenas, pero en esta ocasión lo haremos sin utilizarlas.

COMPETENCIA

Dominar las distintas sentencias que hasta esta práctica hemos podido realizar y comprender con forme hacemos las actividades. Es de suma importancia que comprendamos más allá de su ejecución, si no también debemos comprender su sintaxis y la lógica que implica realizar estas sentencias en los programas de C. Aprenderemos su uso de manera correcta y así mismo aprenderemos en que momento utilizarlas.

FUNDAMENTOS

Los arreglos son estructuras de datos consistentes en un conjunto de datos del mismo tipo. Los arreglos tienen un tamaño que es la cantidad de objetos del mismo tipo que pueden almacenar. Los arreglos son entidades estáticas debido a que se declaran de un cierto tamaño y conservan ´este todo a lo largo de la ejecución del programa en el cual fue declarado.

Un array de tipo unidimensional es básicamente un **vector de datos o lista**. Dicho de otro modo, es un conjunto de variables del mismo tipo y tamaño que ocupan posiciones consecutivas en una memoria. El tamaño de la memoria ocupada por el array es siempre fijo y no se puede variar.

int arreglo1[30] declara que arreglo1 es un arreglo que puede contener 30 enteros. #define TAMANIO 100 int arreglo2[TAMANIO] declara que arreglo2 es un arreglo que puede contener TAMANIO enteros. La ventaja de esta ´ultima declaración es que todos los programas que manipulen arreglo2 utilizaran como tamaño TAMANIO y si quiero cambiar el tamaño del **array** alcanza con cambiar la definición de TAMANIO. Observar que en la declaración se especifica: tipo de los elementos, número de elementos y nombre del arreglo. Un arreglo consta de posiciones de memoria contiguas. La dirección más baja corresponde al primer elemento y la más alta al ´ultimo. Para acceder a un elemento en

particular se utiliza un índice. En C, todos los arreglos usan cero como índice para el primer elemento y si el tamaño es n, el índice del último elemento es n-1.

PROCEDIMIENTO

```
/* Declaración de un array. */



        #include



        main() /* Rellenamos del 0 - 9 */

        {

                int vector[10],i;

                for (i=0;i<10;i++) vector[i]=i;

                for (i=0;i<10; i++) printf(" %d",vector[i]);

        }
```

Podemos inicializar (asignarle valores) un vector en el momento de declararlo. Si lo hacemos así no es necesario indicar el tamaño. Su sintaxis es:

tipo nombre []={ valor 1, valor 2...}

EJEMPLOS

```
int vector[]={1,2,3,4,5,6,7,8};
char vector[]="programador";
char vector[]={'p','r','o','g','r','a','m','a','d','o','r'};
```

Una particularidad con los vectores de tipo char (cadena de caracteres), es que deberemos indicar en que elemento se encuentra el fin de la cadena mediante el carácter nulo (\0). Esto no lo controla el compilador, y tendremos que ser nosotros los que insertemos este carácter al final de la cadena.

Por tanto, en un vector de 10 elementos de tipo char podremos rellenar un máximo de 9, es decir, hasta vector [8]. Si sólo rellenamos los 5 primeros, hasta vector[4], debemos asignar el caracter nulo a vector[5]. Es muy sencillo: vector[5]='\0';

Ejemplo con función en C:

```
#include

void main() {

int arraynota[ 9 ],i ; char Array [ ]={ ‘a’,’b’,’c’,’d’,’e’,’f’,’g’,’h’,’i’ }

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

printf("Ingrese la nota %d:",i+1); scanf("%d",&arraynota[ i ]);

}

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

printf("El elemento%d contiene%c y se encuentra en%x y ocupa %d

:",i,Array[ i ], &Arrai[ i ],sizeof(Array[ i])); /*Esta función

sizeof()retorna el numero de bytes que ocupa el elemento Array [ i ]*/

}

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

printf("El elemento%d contiene%d y se encuentra en%x y ocupa %d

:",i,arraynota[ i ], &arraynota[ i ],sizeof(arraynota[ i]));

}

return 0;

}
```

RESULTADOS Y CONCLUSIONES

Como pudimos observar en esta práctica, nos dimos cuenta de la importancia que tiene todo lo que hemos visto con el transcurso de estas mismas, pudimos ver que para cada nuevo tema que abordamos, utilizamos el tema visto anteriormente y esto con el fin de que todo nos quede de una manera clara y las dominemos de manera adecuada y así mismo con el fin de poder realizar estos programas propuestos por el profesor aplicando todo lo visto anteriormente. Lo que pudimos apreciar respecto al tema nuevo, es que hay una función para leer las cadenas y esta es STRLEEN, que viene incluida en la librería de string.h, pero en esta ocasión solo utilizamos los ciclos para leer una cadena y con esto podemos comprobar de que lo aprendido anteriormente nos sirve para este nuevo tema.

BIBLIOGRAFIAS

https://www.investigacion.frc.utn.edu.ar/tecnicasdigitales/pub/file/arreglos.pdf

https://disenowebakus.net/arrays.php