

Universidad Autónoma de Baja California



Ingeniería en Software y Tecnologías Emergentes.

Taller 7. Apuntadores y caracteres.

Materia: Lenguaje C.

Maestro: Yulith Altamirano

Alumno: Diego Quiros 372688

Fecha: 10/25/2023

Repositorio:

C/C++

https://github.com/diegovq12/LenguajeC-QVDD_932/tree/main/Talleres/Taller7DDQV

Instrucciones:

Durante este taller, vamos a resolver ejercicios relacionados con la unidad 3. Para cada ejercicio, declaren las variables, constantes y funciones necesarias para llevar a cabo la tarea requerida. Realicen el código a mano de los siguientes ejercicios*.

Ejercicios.

1. Declarar un arreglo de apuntadores a caracteres llamado nombres que contenga tres nombres (cadenas de texto). Luego, escribe un bucle que recorra Este arreglo y, en cada iteración, imprima el nombre correspondiente.

C/C++

```
int main()
{
    char* nombres[]={ "Diego", "Demian", "Juanito"};
    int largo = sizeof(nombres) / sizeof(nombres[0]);
    for (int i = 0; i < largo; i++)
    {
        printf("Nombre %d: %s.\n", i+1, nombres[i]);
    }
}
```

2. Escribe un programa que permita al usuario ingresar tres nombres y almacene estos nombres en el arreglo de apuntadores a caracteres nombres. Luego, usa un bucle para imprimir los nombres almacenados.

C/C++

```
int main()
{
    char *nombres[3];

    for (int i = 0; i < 3; i++)
    {
        char temp[100];
        printf("Ingresa nombre %d:\n", i + 1);
```

```

    fgets(temp, sizeof(temp), stdin);
    temp[strcspn(temp, "\n")] = '\0'; // Elimina el carácter de nueva línea del
final

    nombres[i] = strdup(temp);
}

printf("Nombres ingresados:\n");

for (int j = 0; j < 3; j++)
{
    printf("Nombre %d: %s.\n", j + 1, nombres[j]);
}

return 0;
}

```

3. Para los ejercicios anteriores crea una función que sirva para ambos ejercicios donde escribas una función llamada `imprimirNombres` que tome un arreglo de punteros a caracteres como argumento y lo recorra para imprimir cada nombre en una línea separada.

```

C/C++
void imprimirNombres(char *nombres[], int num)
{
    for (int j = 0; j < num; j++)
    {
        printf("Nombre %d: %s.\n", j + 1, nombres[j]);
    }
}

```

4. Escribe una función llamada `invertirCadena` que tome un puntero a `char` como argumento y revierta la cadena original sin crear una nueva cadena. Luego, utiliza esta función para invertir una cadena y mostrar el resultado.

```

C/C++
void invertirCadena(char *cadena);

int main() {
    char nombre[100];
    char temp[100];
    printf("Ingresa cadena: ");
    fgets(temp, sizeof(temp), stdin);

    temp[strcspn(temp, "\n")] = '\0';
}

```

```
strcpy(nombre, temp);
invertirCadena(nombre);
printf("%s\n", nombre);
return 0;
}

void invertirCadena(char *cadena) {
    char caracter;
    int size, j, i;
    size = strlen(cadena);
    for (i = 0, j = size - 1; i < j; i++, j--) {
        caracter = cadena[i];
        cadena[i] = cadena[j];
        cadena[j] = caracter;
    }
}
```