

Temas:

- Variables puntero.

Objetivos:

Que el estudiante logre...

- Identificar las características y usos de las variables puntero mediante su aplicación en problemas básicos y en el recorrido de arreglos.

Condiciones de presentación.

- Este trabajo práctico debe realizarse en forma individual.
- El código de los programas solicitados debe estar correctamente rotulado para su identificación, incluyendo comentarios de seguimiento y deberá ser enviado mediante la plataforma CUV.FCEYT hasta el día indicado como límite de presentación.
- La resolución completa de este trabajo práctico, incluyendo diagramas de flujo, deberá pasar a integrar la carpeta de práctica y autoevaluación del alumno.
- **Criterios de evaluación y aprobación.** Este trabajo práctico recibirá una calificación de aprobado o desaprobado. Para aprobar la presentación debe cumplir como mínimo con los siguientes ítems:
- El trabajo práctico debe estar desarrollado completamente.
- La codificación en lenguaje C debe realizarse siguiendo las recomendaciones de la cátedra.
- La presentación del código de los enunciados solicitados deberá realizarse en tiempo y forma mediante la plataforma CUV.FCEYT: <http://cuv.unse.edu.ar>.

Recursos Bibliográficos

- Introducción al Lenguaje C - Santos Espino.
- Introducción a la Programación con C – Marzal.
- Diapositivas, lecciones y video sobre el tema, disponibles en plataforma del CUV.

Tareas a desarrollar para cada enunciado

- ❖ Desarrolle la diagramación de flujo y la codificación en Lenguaje C de los siguientes enunciados.
- ❖ La resolución de todos los enunciados deberá estar desarrollada en la carpeta de práctica de la asignatura.
- ❖ La entrega de los programas en lenguaje C de los enunciados indicados, deberán ser enviados mediante la plataforma CUV según las fechas y horarios de presentación de cada uno.

Enunciado 1

Toda variable posee un nombre, un tipo y una dirección de memoria. En ocasiones es conveniente acceder a la dirección de una variable en forma directa, sin utilizar su nombre, en esos casos usamos variables puntero.

En una variable puntero podemos almacenar la dirección de una variable. Tanto el puntero como la variable deben poseer el mismo tipo.

En el siguiente código se definen dos variables y un puntero, y se muestran las direcciones y contenidos de cada uno.

```
#include <stdio.h>

int main ()
{
    int j, k;
    int *ptr;

    j = 1;
    k = 2;
    ptr = &k;

    printf("\n");
    printf("j tiene el valor: %d y esta alojado en la dirección: %p\n", j, &j);
    printf("k tiene el valor: %d y esta alojado en la dirección: %p\n", k, &k);
    printf("ptr tiene el valor: %p y esta alojado en la dirección: %p\n", ptr, &ptr);
    printf("El valor del entero al que apunta ptr es: %d\n", *ptr);

    return 0;
}
```

Se pide:

- Dar entrada al código y ejecutarlo.
- Identificar los operadores de referencia y des-referencia que son usados en el código y su funcionamiento para compartirlo con la clase.

Enunciado 1 Presentación: discusión en clase.

Enunciado 2

Teniendo en cuenta el código presentado en el Enunciado 1, modificar el código de modo que se lean de entrada los valores de las variables j y k y se intercambien sus valores usando punteros. Por ejemplo:

Salida: Ingrese el valor de la variable j

Entrada: 22

Salida: Ingrese el valor de la variable k

Entrada: 165

Salida: Ahora el valor de j es 165 y el valor de k es 22

Enunciado 2 Presentación: código en C.

Fecha: 10/09.

Enunciado 3

Dado el ingreso de una cadena que almacena un número real, con tres posiciones enteras y dos posiciones decimales, convertir la cadena a su valor numérico y mostrar por pantalla el número ingresado multiplicado por 2.

Utilizar punteros siempre que sea posible --> usando punteros es posible acceder a los elementos de un arreglo de tipo char.

TRABAJO PRÁCTICO 2

Ejemplo 1:

Entrada: "71.55"

Salida: "Error en la cadena ingresada"

Ejemplo 2:

Entrada: "569.78"

Salida: 1139.56

Ejemplo 3:

Entrada: "1.55"

Salida: "Error en la cadena ingresada"

Enunciado 3 Presentación: codificación en C.

Fecha: 10/09.

Enunciado 4

Escribir un programa que, usando punteros sobre una cadena de caracteres, indique si la palabra almacenada es un palíndromo.

(*) Palíndromo: Se lee igual hacia delante que hacia atrás.

Por ejemplo: "Dábale arroz a la zorra el abad"

Nota: deben ignorarse los acentos, los espacios en blanco y las diferencias entre mayúsculas y minúsculas.

Enunciado 5

Crear un programa que dada la lectura de un string o cadena de caracteres, muestre en pantalla la cadena leída con los caracteres ordenados ascendentemente.

Acceder a la cadena usando punteros.

Enunciado 6

De entrada y verifique la ejecución del siguiente código:

```
#include <stdio.h>
int main()
{
    int i;
    int *ptrToi;           /* Puntero a entero */
    int **ptrToPtrToi;     /* Puntero a puntero a entero */

    ptrToPtrToi = &ptrToi; /* Puntero contiene dirección de puntero */
    ptrToi = &i;           /* Puntero contiene dirección de entero */

    i = 10;                /* Asignación directa */
    *ptrToi = 20;           /* Asignación indirecta */
    **ptrToPtrToi = 30;     /* Asignación con doble indirección */

    printf("La variable i almacena %d", i);

    return 0;
}
```

Envíe la captura de pantalla de su computadora con la codificación y ejecución de este código.

Enunciado 6 Presentación: captura de pantalla.

Fecha: 10/09.

Enunciado 7

Teniendo en cuenta los conceptos expuestos en Punteros a punteros, considere el siguiente código.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #define PALS 4
5  #define MAXLON 80
6  int main ()
7  {
8      char *listapal[PALS];
9      char linea[MAXLON + 1];
10     int i;
11
12     for (i = 0; i < PALS; i++)
13     {
14         printf ("Teclea una palabra: ");
15         gets(linea);
16         listapal[i] = malloc ((strlen (linea) + 1) * sizeof (char));
17         strcpy (listapal[i], linea);
18     }
19
20     for (i = 0; i < PALS; i++)
21         printf ("Palabra %i: %s\n", i, listapal[i]);
22
23     for (i = 0; i < PALS; i++)
24         free (listapal[i]);
25     return 0;
26 }
```

Se pide:

- Dar entrada al código y ejecutarlo.
- Analice y explique cada una de sus sentencias.

Enunciado 7 Presentación: texto en línea.

Fecha: 10/09.

Enunciado 8

Desarrolle un programa que cargue un arreglo entero y luego dado el ingreso de un entero z, desplace hacia la derecha los elementos del arreglo el número de posiciones indicado por z, de modo que los elementos que estaban a la derecha aparezcan por la izquierda.

Utilice punteros.

**** Para completar esta tarea debe COPIAR el código del programa en la ventana de la entrega.

Enunciado 8 Presentación: diagramación y codificación en C.

Fecha: 10/09.