

1. ¿De qué tipo de datos es cada una de las siguientes variables?:

a) `int *a, b;`

a puntero, b puntero a puntero,

b entero

a entero, b puntero

a entero, b entero

b) `int *a,*b;`

a puntero, b puntero a puntero,

b entero

a entero, b puntero

a entero, b entero

2. Analiza, prueba y comenta los dos programas (a y b) por separado, recuerda usar printf para visualizar los valores:

a)

`int a, b;`

`int *pa;`

`a = 5;`

`pa = &a;`

`b = *pa;`

b)

`int i, j,*p;`

`p=&i;`

`*p=21;`

`p=&j;`

`*p=1;`

3. Construye el siguiente programa, paso a paso:

- Declara a, b y c variables enteras, y p, q y r variables puntero a entero.
- Declara m, n variables de tipo float y s y t variables puntero a float.
- Asigna a p la dirección de a y a q la dirección de b.
- Asigna a b el valor 40 usando el puntero q. Mostrar el valor de b, la dirección de b , el valor de q, la dirección de q y el valor contenido en la dirección almacenada en q.
- Ingresar desde teclado el valor de a usando su dirección almacenada. Mostrar el valor de a, la dirección de a, el valor de p, la dirección de p y el valor contenido en la dirección almacenada en p.
- Asigna a r la dirección de a.
- Asigna a la posición de memoria contenida en el puntero r el valor 200. Mostrar el valor de a, la dirección de a , el valor de p, la dirección de p y el valor contenido en la dirección almacenada en p, y el valor de r, la dirección de r y el valor contenido en la dirección almacenada en r.
- Emite un mensaje indicando si el valor del puntero p es igual al valor del puntero q
- Ídem g indicando si el valor del puntero p es igual al valor del puntero r.
- Emite un mensaje indicando si el contenido de la posición de memoria apuntada por p es igual al contenido de la posición de memoria apuntada por q.
- Declara una variable de tipo puntero a entero y una variable entera. Asígnale un valor a la variable entera. Guarda la dirección. Luego lee el contenido de la variable puntero y muéstralo en hexadecimal.

4. Por qué da error la segunda instrucción?

`char c = 'A';`

`double *p = &c;`

5. Un programa en C contiene las siguientes sentencias:

`float a = 0.001, b = 0.003;`

`float c, *pa, *pb;`

`pa = &a;`

`*pa = 2 * a;`

`pb = &b;`

`c = 3 * (*pb - *pa);`

Responde:

- ¿Qué valor tiene a al finalizar el programa?
- ¿Qué valor tiene b al finalizar el programa?
- ¿Qué valor tiene c al finalizar el programa?
- ¿Qué valor tiene (*pa) al finalizar el programa?
- ¿Qué valor tiene (*pb) al finalizar el programa?

6. El siguiente código contiene un error, cuál es?:

```
int main () {  
    int x = 5; float y = 5;  
    int *xPtr = NULL;  
    xPtr = &y;  
    printf ("%d", *xPtr) ;  
    return 0;  
}
```

7. Cuál es el valor final de todas las variables en este programa?

```
int main(){  
    int a, b, *p, *q, *r;  
    char c, d, *m,*n;  
    a=5; b=7;  
    q=&a;  
    p=q;  
    *p=b;  
    printf("%d\\n\\n\\n", *q);
```

```
    r=&b;  
    *r=*q;  
    printf("a=%d, b=%d, *p=%d, *q=%d,  
    *r=%d\\n\\n\\n", a, b,*p,*q,*r);  
    c='A'; d='B'; m=&c; n=&d;  
    *m=*n;  
    return 0;  
}
```

8. Realiza la traza (s e g u i m i e n t o) del siguiente programa. Ejecutalo y comentar qué tarea realiza cada instrucción o línea del programa:

```
int main(){  
    int *p;  
    inta=1, b=2;  
    int **s;  
    p=&a;  
    b=*p+1;  
    s=&p;  
    printf("p=%d, *p=%d, &p=%d\\n", p,*p, &p);  
    printf("s=%d, *s=%d, &s=%d\\n", s,*s, &s);
```

```
    *s=&b;  
    printf("Se ejecutó *s=&b\\n y ahora...\\n");  
    printf("s=%d, *s=%d, &s=%d\\n", s,*s, &s);  
    **s=79;  
    printf("a=%d, &a=%d\\n", a, &a);  
    printf("b=%d, &b=%d\\n", b, &b);  
    return 0;  
}
```

9. Analiza el siguiente programa:

```
int main() {  
    int a,b,c; int *p1,*p2; p1 = &a;  
    *p1 = 1;  
    p2 = &b;  
    *p2 = 2;
```

```
    p1 = p2;  
    *p1 = 0;  
    p2 = &c;  
    *p2 = 3;  
    return 0;}
```

10. Completa el siguiente programa:

```
int main(){  
    int num, n;  
    int *dir_n;  
    num=22;  
    n=7;  
    dir_n=&n;  
    printf("La variable num vale: %d\\n",____);  
    printf("La dirección de memoria donde esta almacenada la variable num es: %p\\n", ____);  
    printf("La dirección de memoria almacenada en dir_n es: %p\\n", ____);  
    printf("El valor de la variable apuntada por dir_n es: %d\\n",____);  
    return 0;  
}
```

11. Qué emite el siguiente programa?:

```
int main(){
    int u = 3, v;
    int *pu;
    int *pv;
    pu = &u;
    v = *pu;
    pv = &v;
    printf("\nu=%d &u=%X pu=%X *pu = %d", u, &u, pu, *pu);
    printf("\nv=%d &v=%X pv=%X *pv = %d", v, &v, pv, *pv);
    return 0; }
```

12. Investiga qué emite por pantalla el siguiente programa:

```
int main() {
int n = 8, * ptr;
    printf("ptr es%p\n", ptr);
    ptr = &n;
    printf("n es %d\n", n);
    printf("&n es %p\n", &n);
    printf("*ptr es %d\n", *ptr);
    printf("ptr es %p\n", ptr);
    n--;
    printf("n es despues de n-- %d\n", n);
    printf("&n es %p\n", &n);
    ptr++;
    printf("ptr es despues de ptr++ %p\n", ptr);
    printf("*ptr es %d\n", *ptr);
    return 0; }
```

13. Considera las siguientes instrucciones:

```
int *p; int i; int k;

    i = 42;

    k = i;

    p = &i;
```

Luego de esas instrucciones, cuál de las siguientes cambia el valor de i a 75?

- k = 75;
- *k = 75;
- p = 75;
- *p = 75;
- Dos o más de las anteriores.

14. Explica el funcionamiento del siguiente programa:

```
int main(){
int a, *p;
a=5;
p=&a;
*p+=7;
printf("\nEl valor final de a es: %d\n", a);
return 0;
}
```

15. Explica la salida del siguiente programa:

```
int main() {
    int n;
    int * ptr;
    n=8;
    printf("ptr es%p\n", ptr);
    ptr=&n;
    printf("n es %d\n", n);
    printf("&n es %p\n", &n);
    printf("*ptr es %d\n", *ptr);
    printf("ptr es %p\n", ptr);
    n--;
    printf("n es despues de n-- %d\n", n);
    printf("&n es %p\n", &n);
    ptr++;
    printf("ptr es despues de ptr++ %p\n", ptr);
    printf("*ptr es %d\n", *ptr);
    return 0;
}
```

16. Casting y punteros a void: Comenta el siguiente programa e identifica (si los hay) los errores:

```
int *p;
double *q;
void *r;
p=q;
p=(int*)q;
p=r=q
```

17. Transcribe el siguiente programa y extrae conclusiones:

```
int main() {
    void *generico;
    int *pint, x=3;
    char *pchar = "Punteros a caracteres";
    pint=&x;
    printf("El valor apuntado por pint es %d (valor de x)\n", *pint);
    generico=pchar;
    printf("\n\ngenerico (tipo void) apunta a objeto char: %s", generico);
    printf("\n\nImprimo ahora generico como char:\n\n");
    puts((char *)generico);
    return 0;
}
```

18. Sea a una variable entera: int a = 25; Se declaran 4 punteros:

```
int *ptr1, **ptr2, ***ptr3, ****ptr4;
```

¿Cómo podríamos imprimir el valor 25 de la variable a, con los cuatro punteros?. Construye el programa para verificar.

19. Determina la salida del siguiente programa:

```
int main () {
```

```

int x = 5;
int y = 10;
int *xPtr = NULL;
int *yPtr = NULL;
xPtr = &x;
yPtr = &x;
*xPtr = *xPtr + 1;
(*yPtr)--;
printf ("x:%d\n",x);
printf ("y:%d\n",y);
return 0; }

```

20. Investiga qué emite por pantalla el siguiente programa:

```

int main() {
    int n = 8, * ptr;
    printf("ptr es%p\n", ptr);
    ptr=&n;
    printf("n es %d\n", n);
    printf("&n es %p\n", &n);
    printf("*ptr es %d\n", *ptr);
    printf("ptr es %p\n", ptr);
    n--;
    printf("n es después de n--
%d\n", n);
    printf("&n es %p\n", &n);
    ptr++;
    printf("ptr es después de ptr++
%p\n", ptr);
    printf("*ptr es %d\n", *ptr);
return 0; }

```

21. Analiza el siguiente ejercicio:

```

int main(){
int a,b,*pta,**ptb;
pta=&a;
ptb=&pta;
*pta=8;
printf("ptb es: %d\n\n",ptb);
printf("Contenidos de *ptb después de &pta es: %d\n\n",*ptb);
printf("Contenidos de **ptb después de &pta es: %d\n\n",**ptb);

**ptb=**ptb+3;
printf("Contenidos de **ptb es: %d\n\n",**ptb);
printf("Contenidos de a es: %d\n\n",a);

*ptb=NULL;
printf("Contenidos de a después de *ptb=NULL, es: %d\n\n",a);
printf("Contenidos de *ptb después de NULL es: %d\n\n",*ptb);

b=8;
pta=&b;
printf("Contenidos de *ptb después de pta=&b, es: %d\n\n",*ptb);
printf("Contenidos de a es: %d\n\n",a);
printf("Contenidos de b después de pta=&b, es: %d\n\n",b);

```

```

printf("Contenidos de *pta después de pta=&b, es: %d\n\n",*pta);
printf("Contenidos de **ptb después de pta=&b, es: %d\n\n",**ptb);
printf("Contenidos de *ptb después de pta=&b, es: %d\n\n",*ptb);
**ptb=a+10;
printf("Contenidos de **ptb después de **ptb=a+10, es: %d\n\n",**ptb);
printf("Contenidos de *pta después de **ptb=a+10, es: %d\n\n",*pta);
printf("Contenidos de a después de **ptb=a+10, es: %d\n\n",a);
printf("Contenidos de b después de **ptb=a+10, es: %d\n\n",b);
printf("La suma de los contenidos de pta + ptb es: %d\n\n",(*pta) + (**ptb));
return 0;
}

```

Reconstruye los siguientes programas (de prácticas anteriores) utilizando sólo variables punteros

De la práctica sentencias simples

Ejercicio 24. Desarrolla un algoritmo que le permita leer un valor para radio (R), calcular el área (A) de un círculo y emitir su valor.

De la práctica de condicionales

Ejercicio 8. Desarrolla un algoritmo que le permita leer tres valores y almacenarlos en las variables A, B y C respectivamente. El algoritmo debe indicar cuál es el mayor. Para este caso se asume que los tres valores leídos por el teclado son valores distintos. Análisis: Es necesario leer los tres valores a comparar, cada uno de ellos se almacena en una variable que para el ejercicio será A, B y C. Para saber si A es el valor mayor se compara con las variables B y C respectivamente. En caso de ser mayor se escribe el mensaje, en caso contrario se sigue verificando otra variable caso B y si no por defecto se dirá que C es el mayor asumiendo que los tres valores almacenados son diferentes.

De la práctica de switch

Ejercicio 1. Construye un programa que ingrese un carácter y determine si es una vocal.

De la práctica de ciclos

Ejercicio 7. (while) Desarrolle un algoritmo que le permita leer un valor entero positivo N y decir si es primo o no.

Ejercicio 5. (do while) Pedro invierte u\$s 100 a una tasa del 8% anual. Jorge invierte u\$s 150 al 5% anual. ¿Después de cuántos años la cuenta de Pedro rebasará a la de Jorge?

Ejercicio 9. (for) Escribe un programa de sueldos para una pequeña compañía que tiene seis empleados. Para cada empleado/a, el programa recibe desde el teclado el nombre, sexo, horas trabajadas y sueldo por hora. Por cada empleado ingresado, el programa deberá mostrar en pantalla el nombre y el sueldo de cada empleado. Posteriormente debe informar el total de pagos para cada sexo, y el sueldo promedio de hombres y de mujeres.

De la práctica de ciclos anidados

Ejercicio 11. Construye un programa de gestión para una pequeña compañía que tiene seis empleados. Cada empleado tiene un número de legajo que está entre 111 y 999. Para cada empleado/a, el programa recibe desde el teclado el nombre, sexo, horas trabajadas y sueldo por hora. Pero además, por cada empleado, se deben descontar 3% por la obra social y 11% para la jubilación. Del resultado anterior se le descontarán una serie de valores correspondientes a adelantos de sueldo dados durante el mes.

Por cada empleado ingresado, el programa deberá emitir	Al finalizar el programa se debe informar:
a. el nombre y sueldo a cobrar.	c. el total de pagos para cada sexo, d. el sueldo promedio de hombres y de mujeres e. qué legajo obtuvo mayores ingresos f. total descontado de obra social g. total descontado de jubilación