

Universidad Nacional del Litoral
Facultad de Ingeniería y Ciencias Hídricas
Departamento de Informática



Ingeniería Informática

**PROGRAMACIÓN ORIENTADA
A OBJETOS**

UNIDAD 1
Punteros

Actividades

Actividades

Ejercicios

1. Explique el significado de las siguientes expresiones de código C++.

- | | |
|---|--|
| a) <code>char *m;</code> | f) <code>int *a[30];</code> |
| b) <code>double x,y;</code>
<code>double *px, *py;</code> | g) <code>float f(int *a, int *b);</code> |
| c) <code>int a=25;</code>
<code>int *k=&a;</code> | h) <code>int (*pfunc)(void);</code> |
| d) <code>float (*q)[20];</code> | i) <code>char *a[3]={“rojo”, “gris”, “azul”};</code> |
| e) <code>float x,y;</code>
<code>float *px, *py=&y;</code> | j) <code>float(*pfunc)(int *a, char b);</code> |

2. Usando la sintaxis de C++ escriba el código para:

- Declarar un puntero a datos de tipo flotante y otro puntero a datos de doble precisión.
- Declarar un puntero a una lista de 10 arreglos contiguos de 15 elementos enteros cada uno. Reserve, además, la memoria necesaria.
- Declarar un arreglo de punteros para representar una matriz de 10x30 elementos flotantes.
- Declarar un puntero a una función que acepte 2 parámetros de doble precisión y no devuelva resultado alguno.
- Reservar memoria para un entero y un arreglo de 10 datos tipo float.
- Liberar la memoria reservada en el apartado anterior.
- Declarar una función que tenga otra función como argumento y devuelva un puntero a un entero. La función argumento debe tener 2 parámetros enteros y devuelve un carácter.

3. Observe la porción de código C++ del recuadro de la derecha y determine la salida que se obtiene de los flujos de salida **cout** propuestos. Considere que la variable **a** se ha almacenado en memoria a partir de la dirección **0000FF09**. Responda: ¿sería posible revisar si su respuesta fue correcta compilando y ejecutando este código? Justifique su respuesta.

```
.....
int a=90;
int *p=&a;
int b=(*p)++;
int *q=p+2;
cout<<p<<" "<<*p;
cout<<q<<" "<<*q;
cout<<a<<" "<<b;
p++; b=*(q--)-1;
a=(*p++)+1;
cout<<a<<" "<<b;
.....
```

4. Analice el código C++ del recuadro de abajo para responder lo siguiente:

- Qué tipo de parámetros actuales se emplean para llamara a **func**?
- Qué tipos de parámetros formales se definen en **func**?
- Qué tipo de información devuelve la función **func**?
- Cuál es la salida que se obtiene en el programa correspondiente al código propuesto para **func**?

```

void func(int *p, int *sum){
    int i; (*sum)=0;
    for(i=0;i<6;i++)
        (*sum)+=*(p+1);
    cout<<"sum="<<*sum<<endl;
}

int main(){
    int x[6]={12,34,56,78,39,90};
    int suma;
    func(x, &suma);
    x++;
    .....
    func(x, &suma);
    x++;
    .....
}

```

5. A continuación se declara un arreglo **a** de 10 elementos enteros. El elemento inicial **x[0]** se ubica en la dirección de memoria 000011E4: **int a[10]={110, 120, 130, 140, 150, 160, 170, 180, 190, 200};** Determine lo que que representan las siguientes expresiones:
 - a) **x;**
 - b) **(x+4);**
 - c) ***x;**
 - d) ***(x+3);**
 - e) ***x+3;**
6. Utilizando notación de punteros, generar un arreglo lineal de N elementos numéricos enteros, con valores aleatorios entre 1000 y 1500, y muestre en pantalla la dirección de memoria del mayor elemento. N es un dato ingresado por el usuario.
7. Amplíe el programa anterior para que luego de generar el arreglo aleatorio, permita ingresar un valor M que debe ser insertado en la posición 32 de dicho arreglo y muestre el vector modificado.
8. Modifique el ejercicio 6 para que el usuario pueda luego ingresar un entero C y se inserten esa cantidad de ceros al final del vector. Para realizar esto reserve dinámicamente un nuevo arreglo que contenga lugar para guardar los datos anteriores y los ceros que serán insertados, luego copie en la nueva memoria los datos del vector viejo y agregue los ceros. La memoria del primer arreglo debe ser liberada y el puntero del arreglo debe quedar apuntando a la nueva memoria con los ceros agregados. Finalmente, muestre el arreglo resultante.
9. Usando notación de punteros genere aleatoriamente una matriz de números reales de doble precisión de 10 filas por 6 columnas y determine e informe:
 - a) El promedio de la fila que el usuario ingrese como dato.
 - b) La suma de cada columna.

10. Escriba una función que utilice punteros para buscar e informar la dirección de un entero dentro de un arreglo. Se pasan como parámetros el arreglo y el entero a buscar. Si el dato no se encuentra, devolver la dirección de memoria nula (NULL).
11. Escriba la función Plural cuyo prototipo se expone a continuación. Debe retornar en el mismo parámetro, el plural correspondiente a la palabra. Considere agregar 's' si la palabra termina en vocal y 'es' si termina en consonante. Utilice notación de punteros para realizar todas las operaciones sobre la cadena.

void plural (char *p);

Cuestionario

1. ¿Qué es un puntero? ¿Cómo se declara en C++ ?
2. Indique dos formas de emplear el operador de referencia &.
3. ¿Cómo se expresa el contenido de una variable apuntada por un puntero *q*?
4. ¿Cuál es la diferencia entre las dos acciones siguientes que emplean el operador &?
char &a=b;
p=&b;
5. Proponga 2 maneras de acceder a la dirección del sexto elemento de un arreglo *x*.
6. ¿Qué operaciones pueden realizarse con punteros? Ejemplifique.
7. Si *p* es un puntero a enteros y *q* un puntero a datos de tipo double, cuál es la diferencia en bytes entre:
 - a) **p** y **p+4**
 - b) **p** y **p-1**
 - c) **q** y **q+5**
 - d) **q** y **q++**
8. Si una cantidad *n* es sumada a una variable puntero. Interprete el resultado de dicho cálculo. ¿Y si se resta *n* del puntero?.
9. De acuerdo con la siguiente declaración de las variables **dato** y **p**:
int dato=25;
int *const p=&dato;
10. Determine si son válidas las siguientes proposiciones. Justifique en caso de responder negativamente
 - a) **dato=100;**
 - b) **p=&(dato+1);**
 - c) **p++;**
 - d) **p=NULL;**
11. Explique la diferencia entre las dos declaraciones siguientes:
float func(); **float (*func)();**
12. ¿Cuál es la ventaja de emplear variables dinámicas ? ¿Cómo las define en C++?