# PROGRAMACIÓN

# HOJA GUÍA

# PRÁCTICA: Punteros

**KEVIN GALARZA**

## Problemas planteados

1. Sin ejecutarlo, ¿qué mostraría el siguiente código?

int x = 5, y = 12, z;

int \*p1, \*p2, \*p3;

p1 = &x;

p2 = &y;

z = \*p1 \* \*p2;

p3 = &z;

(\*p3)++;

p1 = p3;

cout << \*p1 << " " << \*p2<< " "<< \*p3;

Respuesta: 61 12 61

1. Dada una lista de enteros desordenada, queremos ordenarla. Escribe los respectivos subprogramas de ordenación empleando cualquiera de los algoritmos vistos en clase para:
2. Una lista implementada como array de datos dinámicos.
3. Una lista implementada como array dinámico.

Respuesta:

//Pegar aquí el código

#include<iostream>

#include<cstdlib>

#include<ctime>

using namespace std;

void Datos();

void ordenar(int\* punt, int n);

void mostrar(int\* punt, int n);

int tamanio, \* arreglo;

int main() {

Datos();

cout << "\t\tARREGLO GENERADO ALETORIAMENTE: " << endl;

mostrar(arreglo, tamanio);

ordenar(arreglo, tamanio);

cout << "\t\tARREGLO GENERADO ALETORIAMENTE EN ORDEN: " << endl;

mostrar(arreglo, tamanio);

delete[] arreglo;

return 0;

}

void Datos() {

cout << endl << endl << "\t\tDigite el numero de elementos: ";

cin >> tamanio;

arreglo = new int[tamanio];

srand(time(0));

for (int i = 0; i < tamanio; i++) {

\*(arreglo + i) = (rand() % 65) + 4;

}

}

void ordenar(int\* punt, int n) {

int aux;

for (int i = 0; i < n; i++) {

for (int j = 0; j < n - 1; j++) {

if (\*(punt + j) > \*(punt + j + 1)) {

aux = \*(punt + j);

\*(punt + j) = \*(punt + j + 1);

\*(punt + j + 1) = aux;

}

}

}

}

void mostrar(int\* punt, int n) {

cout << "\t\t";

for (int i = 0; i < n; i++) {

cout << "[" << \*(punt + i) << "]";

}

cout << endl;

}

1. Los siguientes fragmentos de código emplean memoria dinámica, pero su funcionamiento no es evidente. Indica, para cada fragmento, cuál es el resultado de la ejecución y si el código tiene algún problema o defecto. Indica también en qué zona de la memoria se guarda cada uno de los datos.

Imagen que contiene texto

Descripción generada con confianza muy alta

Respuesta:

1. Falta eliminar la variable dinámica. Y además se declara dos veces como dinámica.
2. Se esta eliminando q, y esta variable no se esta declarando como dinámica. Además, falta eliminar p, que si es una variable dinámica.
3. qq=pp es incorrecto, ya que qq es una variable int y pp es un puntero int. Por otro lado, q=\*qq no puede realizarse poque la variable qq solo es un entero y no puntero. Tambien p=q no puede ser ya que uno es variable entera y el otro un puntero int. Se quiere imprimir \*qq y este es una variable tipo int, no es un puntero. Por ultimo, p no es declarada como variable dinámica y se intenta eliminarla sin serlo. Y cabe recalcar que se creo \*pp como variable dinámica pero nunca apunto a nada.
4. ¿Cómo declararías un puntero constante p para apuntar a una constante entera? (Repasa las diapositivas 31-32 de la presentación del tema, y el ejercicio de repl.it 4.27 Punteros y constantes).

Respuesta:

: EJEMPLO:

const int a=10;

const int \*const p=&a;

1. Como podemos tener punteros que apunten a cualquier tipo de datos, también podemos tener punteros que apunten a punteros:

int x = 5;

int \*p = &x; // Puntero a entero

int \*\*pp = &p; // Puntero a puntero a entero

Para acceder a x a través de p escribimos \*p.

Para acceder a x a través de pp escribimos \*\*pp, o \*(\*pp).

Con \*pp accedemos a p, el otro puntero.

Indica qué es lo que muestra el siguiente código:

* 1. int x = 5, y = 8;
  2. int \*px = &x, \*py = &y, \*p;
  3. int \*\*ppx = &px, \*\*ppy = &py, \*\*pp;
  4. p = px; px = py;
  5. py = p;
  6. pp = ppx;
  7. ppx = ppy;
  8. ppy = pp;
  9. cout << \*\*ppx << "  " << \*\*ppy;

Respuesta: 5 8

Para acceder a x atreves de px escribimos \*px;

Para acceder a x atreves de py escribimos \*py;

Para acceder a x a través de ppx escribimos \*\*ppx;

Para acceder a y a través de ppy escribimos \*\*ppy;

Para acceder a x a travez de p escribimos \*p;

Luego, para acceder a y a travez de px escribimos \*px;

Para acceder a x por py escribimos \*py;

Para acceder a x por pp escribimos \*pp;

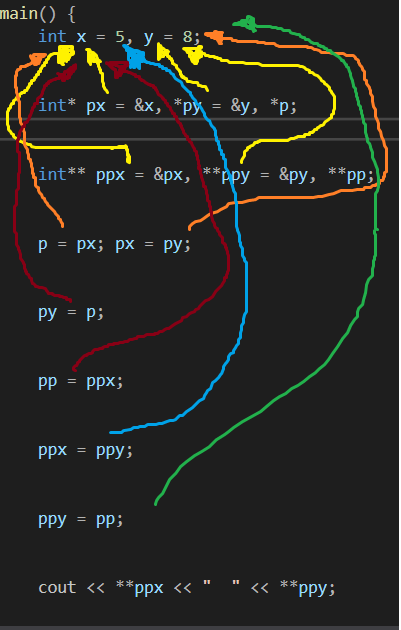
Para acceder a y por ppx escribimos \*\*ppx;

Para acceder a x por ppy escribimos \*\*ppy;

El programa va a imprimir: 5 8

También, dibuja los distintos datos y cómo van apuntando los punteros a los otros datos a medida que se ejecutan las líneas de instrucciones.

Respuesta:



## Informe

* **En este mismo documento se deberá resolver los problemas y subir este archivo al aula virtual, en la respectiva tarea en formato de Word.**