

UNIVERSIDAD NACIONAL SAN AGUSTIN DE AREQUIPA
FACULTAD DE INGENIERÍAS
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



LABORATORIO 02

Asignatura: Ciencia de la Computación II

Docente: Rolando Jesus Cardenas Talavera

Grupo: B

Estudiante: Roni Ezequiel Montañez Pacco

Arequipa-Perú

25 de abril del 2024

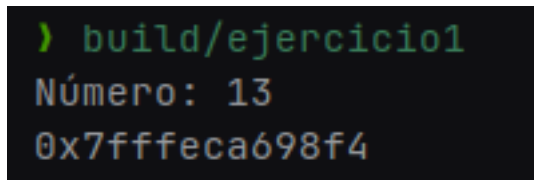
Ejercicios

1. Escriba un programa que permita ingresar un valor numérico y muestre la dirección de la variable

CÓDIGO

```
#include <iostream>
using namespace std;
int main(){
    int num;
    cout << "Número: ";
    cin >> num;
    cout << &num << endl;
}
```

EJECUCIÓN



```
build/ejercicio1
Número: 13
0x7fffeca698f4
```

2. Cree una función que reciba un puntero por referencia y asigne un valor de NULL. Muestre el contenido del puntero antes y después de llamar a esta función.

CÓDIGO

```
#include <iostream>
using namespace std;

void conversion(int *&num){
    num=NULL;
}

int main(){
    int *num = new int(8);
    cout << num << endl;
    conversion(num);
    cout << num << endl;
}
```

EJECUCIÓN

```
> build/ejercicio2
0x60a5944992b0
0
```

3. Escriba una funcion donde se declare una variable y se le asigne un valor numerico. Cree un puntero numerico que tenga asignado el valor de la variable anterior. Muestre el valor y la direccion actual del ambas variables.

CÓDIGO

```
#include <iostream>
using namespace std;
void func(){
    int num = 8;
    int *p = new int(num);
    cout << "\tVariable\nValor: " << num << endl;
    cout << "Direccion: " << &num << endl;
    cout << "\tPuntero\nValor: " << *p << endl;
    cout << "Direccion: " << &p << endl;
    delete p;
}
int main(){
    func();
}
```

EJECUCIÓN

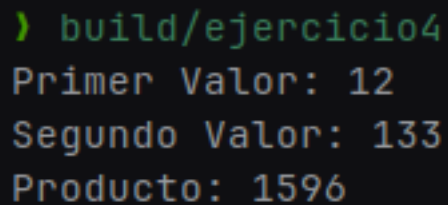
```
> build/ejercicio3
Variable
Valor: 8
Direccion: 0x7ffc99d46fbc
Puntero
Valor: 8
Direccion: 0x7ffc99d46fc0
```

4. Escriba un programa que reciba dos valores enteros en dos punteros y realice la multiplicacion de ambos valores.

CÓDIGO

```
#include <iostream>
using namespace std;
int main(){
    int *x = new int,*y = new int;
    cout << "Primer Valor: ";
    cin >> *x;
    cout << "Segundo Valor: ";
    cin >> *y;
    cout << "Producto: " << *x**y << endl;
    delete x;
    delete y;
}
```

EJECUCIÓN



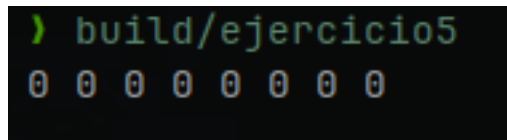
```
> build/ejercicio4
Primer Valor: 12
Segundo Valor: 133
Producto: 1596
```

5. Cree una funcion que reciba un puntero vacio y un número entero n, cree una lista de n elementos inicializados en cero y sea retornado en el puntero dado. Imprima los valores desde donde se invoco la funcion creada.

CÓDIGO

```
#include <iostream>
using namespace std;
void function(int *&p, int n){
    p = new int[n];
    for(int i=0;i<n;i++){
        *(p+i) = 0;
    }
}
int main(){
    int n=8;
    int *p=NULL;
    function(p,n);
    for(int i=0;i<n;i++){
        cout << *(p+i) << " ";
    }
    cout << endl;
}
```

EJECUCIÓN



6. Escriba una funcion que cree dos lista de punteros con igual cantidad de elementos de n elementos aleatorios. Escriba una segunda funcion que reciba las dos listas de valores almacenados en punteros, e intercambie ambos valores. Muestre el contenido de ambos punteros antes y despues del intercambio.

CÓDIGO

```
#include <iostream>
#include <ctime>
#include <cstdlib>

using namespace std;

void function(int *&p1, int *&p2, int n){
    p1 = new int[n];
    p2 = new int[n];
    srand(time(NULL));
    for(int i=0;i<n;i++){
        *(p1+i)=rand()%100;
        *(p2+i)=rand()%100;
    }
}

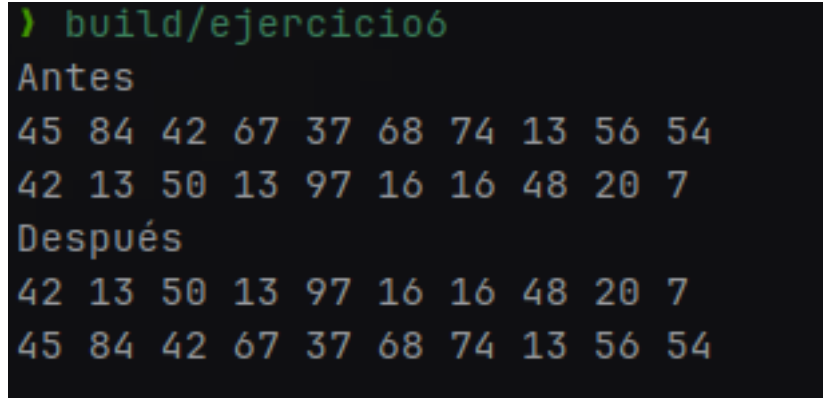
void function2(int *&p1, int *&p2, int n){
    for(int i=0;i<n;i++){
        int tmp = *(p1+i);
        *(p1+i) = *(p2+i);
        *(p2+i) = tmp;
    }
}

void print(int *p, int n){
    for(int i=0; i<n; i++){
        cout << *(p+i) << " ";
    }
    cout << endl;
}

int main(){
    int n=10;
    int *p1,*p2;
    function(p1,p2,n);
    cout << "Antes\n";
    print(p1,n);
    print(p2,n);
    function2(p1,p2,n);
    cout << "Después\n";
    print(p1,n);
}
```

```
    print(p2,n);  
}
```

EJECUCIÓN



```
) build/ejercicio6  
Antes  
45 84 42 67 37 68 74 13 56 54  
42 13 50 13 97 16 16 48 20 7  
Después  
42 13 50 13 97 16 16 48 20 7  
45 84 42 67 37 68 74 13 56 54
```

7. Cree una funcion que permita ingresar valores a un Array de valores decimales y muestre su contenido. Tanto el ingreso de los valores dentro del Array como su impresion de valores debe de realizarse con punteros.

CÓDIGO

```
#include <iostream>  
using namespace std;  
void function(double *&arr){  
    int tam;  
    cout << "Tamaño del array: ";  
    cin >> tam;  
    arr = new double[tam];  
    for(int i=0;i<tam;i++){  
        cout << "Ingrese: ";  
        cin >> *(arr+i);  
    }  
    for(int i=0; i<tam; i++){  
        cout << *(arr+i) << " ";  
    }  
    cout << endl;  
}  
int main(){  
    double *arr;  
    function(arr);  
    delete [] arr;  
}
```

EJECUCIÓN

```
) build/ejercicio7
Tamaño del array: 6
Ingrese: 1.2
Ingrese: 1.53
Ingrese: 6.35233
Ingrese: 6.2
Ingrese: 24.2
Ingrese: 2.4
1.2 1.53 6.35233 6.2 24.2 2.4
```

8. Escriba una funcion que copie el contenido de un Array de valores decimales a otro usando solo punteros. Su funcion no debe de tener la sentencia return.

CÓDIGO

```
#include <iostream>
using namespace std;
void function(double *p1, double *p2, int tam){
    p2 = new double[tam];
    for(int i=0;i<tam;i++){
        *(p2+i)=*(p1+i);
    }
}

void print(double *arr, int tam){
    for(int i=0;i<tam;i++){
        cout << *(arr+i) << " ";
    }
    cout << endl;
}

int main(){
    double *p1 = new double[6]{1.2,1.3,1.5,6.5,8.32,7.82};
    double *p2;
    function(p1,p2,6);
    cout << "Array 1\n";
    print(p1,6);
    cout << "Array 2\n";
    print(p2,6);
    delete [] p1;
    delete [] p2;
}
```

EJECUCIÓN

```
> build/ejercicio8  
Array 1  
1.2 1.3 1.5 6.5 8.32 7.82  
Array 2  
1.2 1.3 1.5 6.5 8.32 7.82
```