

Apuntadores

Apuntadores

Apuntador

AB01

-2,147,483,648 a -2,147,483,648

- Cuando se define una variable en C se aparta un espacio en la memoria, y su dirección se denota con valores hexadecimales.
- Un apuntador es una variable que almacena una dirección de memoria.
- La cual corresponde a una variable que contiene un dato.
- Una variable contiene un valor específico, y un apuntador contiene la dirección de la variable.

| | |
|------|-----------|
| AB01 | 1001 1101 |
| AB02 | 1100 0101 |
| AB03 | 0001 1101 |
| AB04 | 0000 0001 |

AB05

No necesariamente la variable se guarda en localidades contigua

Declaración de apuntadores

- `int *variableApuntador;`
- `int y=10;`
- `variableApuntador= &y`

*k

j

| Dirección | Valor |
|-----------|-------|
| HE01 | HE03 |
| HE02 | |
| HE03 | 0000 |
| HE04 | 0000 |
| HE05 | 0000 |
| HE06 | 1111 |
| HE07 | |
| 95AE | HE03 |
| 10FE | HE03 |

```
1. int main(){
2.     int j=5;
3.     int *apuntadorInt;
4.     apuntadorInt=&j;

6.     cout << "&j: " << &j<<endl;
7.     cout << "apuntadorInt: " << apuntadorInt<<endl;
8.     cout << "*apuntadorInt: " << *apuntadorInt<<endl;
9.     j=10;
10.    cout << "\r *apuntadorInt: " << *apuntadorInt<<endl;
11.    *apuntadorInt=15;
12.    cout << "j: " << j<<endl;
13.    return 0;
14.}
```

```
<terminated> (exit value: 0) Apuntadore
&j: 0x7ffeed1941c8
apuntadorInt: 0x7ffeed1941c8
*apuntadorInt: 5

    *apuntadorInt: 10
j: 15
```

Funciones por referencia

- Cuando se llama a una función con argumentos que deberán ser modificados se pasa la dirección de los argumentos.
- Esto se realiza utilizando el operador &.
- Los arreglos no son pasados mediante el operador & porque C pasa automáticamente la posición inicial en memoria del arreglo.

Ejemplo

- Ver código apuntadores

Calificador Const

- El calificador **const** permite al programador indicarle al compilador que el valor de una variable no deberá cambiar.

Ejemplo código de burbuja

```
void Ordenamiento::ordenaBurbuja(int *arreglo, const int size){  
    cout<<"size: "<<size<<endl;  
    int valor1, contadorA, contadorB;  
    for (contadorA=0; contadorA<size; contadorA++){  
        for(contadorB=0; contadorB<size-1; contadorB++){  
            if(arreglo[contadorB] > arreglo[contadorB+1]){  
                this-  
>intercambia(&arreglo[contadorB],&arreglo[contadorB+1]);  
            }  
        }  
    }  
}  
void Ordenamiento::intercambia(int *valor1, int *valor2){  
    int temporal;  
    temporal= *valor1;  
    *valor1=*valor2;  
    *valor2=temporal;  
}
```

El sizeof

- El lenguaje C proporciona un operador especial llamado sizeof, El operador regresa el tamaño de la estructura o el arreglo.

```
cout<< "\r Tamaño del arreglo: " << sizeof(edades);
```