

El Lenguaje de Programación en C

Moisés García Villanueva

1 de Octubre de 2012.

Abstract

El alumno identificará los elementos de programación del lenguaje imperativo C, desarrollará las habilidades del lenguaje e implementará los ejercicios básicos.

1 Principios de C

1.1 Estructura básica de un programa en C

Estructura básica de un programa en C

Librería

Función principal

```
#include<stdio.h>
main()
{
    //instrucciones del
    //lenguaje
}
```

1.2 Variables, operadores aritméticos, instrucciones, estructuras de datos, apuntadores y funciones.

1. Hacer un programa en C que lea un archivo de texto e imprima únicamente los caracteres con valor ASCII que sean módulo 2.
2. Hacer un programa en C que lea una matriz de números complejos de la forma $(4,-3i)$ y nos imprima la matriz transpuesta.
3. Crear un programa en el lenguaje C que contenga las siguientes partes:
 - Una estructura de datos para leer una imagen en el formato .ppm
 - Utilizar memoria dinámica para almacenar los datos de la imagen.
 - Implementar la función rotar que aplica dicha transformación a la imagen almacenada en la estructura.

- Implementar la función GuardarEnArchivo que te permite almacenar la imagen que dicha función recibe como parámetro en el archivo cuyo nombre se recibe como parámetro también en la función.

Memoria Dinámica en C

Como crear un vector utilizando memoria dinámica en C, utilizando `void *malloc(size_t tamaño_tipo)`.

```
#include<stdio.h>
#include<stdlib.h>
main(){
    int n; // número de elementos del vector
    int *vector;
    int i;

    printf("Dame el número de elementos del vector: ");scanf("%d",&n);

    // Creamos dinámicamente el vector con malloc
    vector = malloc (n * sizeof(int));
    // Leemos los números y los almacenamos en vector
    for(i=0;i<n;i++){
        scanf("%d",&(vector[i]));
    }
    // imprimimos el vector
    for(i=0;i<n;i++)
        printf("%d ",*vector++);
    printf("\n");
}
```

La función `calloc` funciona de modo similar a `malloc`, pero además de reservar memoria, inicializa a 0 la memoria reservada. Se usa comúnmente para arreglos y matrices. Está definida de esta forma:

```
void *calloc(size_t nmemb, size_t size);
```

El parámetro `nmemb` indica el número de elementos a reservar, y `size` el tamaño de cada elemento. El ejemplo anterior se podría reescribir con `calloc` de esta forma:

```
#include<stdio.h>
#include<stdlib.h>
main(){
    int n; // número de elementos del vector
    int *vector;
    int i;

    printf("Dame el número de elementos del vector: ");scanf("%d",&n);

    // Creamos dinámicamente el vector con malloc
    vector = calloc (n, sizeof(int));
    // Leemos los números y los almacenamos en vector
    for(i=0;i<n;i++){
        scanf("%d",&(vector[i]));
    }
    // imprimimos el vector
    for(i=0;i<n;i++)
        printf("%d ",*vector++);
    printf("\n");
}
```