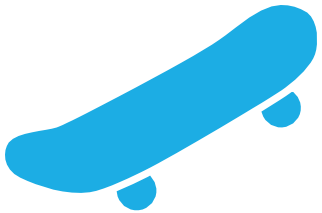
PROGRAMACIÓN AVANZADA

Ejercicios sobre apuntadores

EJERCICIO 1

Realicen su propia implementación de la función MEMSET, pueden agregar que se modifique un rango de caracteres.



SOLUCIÓN

```
int main(void) {
  char someString[] = "We all like programming in
C.";
  printf("Original string: %s\n", someString);
  mymemset(someString, '$', 7, 10);
  printf("Modified string: %s\n", someString);
}
```

```
void mymemset(char *sPtr, char c, int
init, int end) {
  int pos = 0;
  while(*sPtr != 0) {
      if (pos >= init && pos <= end)
          *sPtr = c;
      ++sPtr;
      pos++; } }
```

EJERCICIO 2

Realice un programa en C que defina una estructura para representar puntos en tres dimensiones.

Su programa solo podrá acceder a la información almacenada en la estructura a través de punteros.

Escriba una función que, dado dos punteros a puntos en 3D, devuelva la distancia que hay entre ellos.

SOLUCIÓN

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>


typedef struct {
    int x;
    int y;
    int z;
} Point3D;
```

```c
double Euclidean(Point3D *p1, Point3D *p2) {
    double difx = pow(p1->x - p2->x, 2);
    double dify = pow(p1->y - p2->y, 2);
    double difz = pow(p1->z - p2->z, 2);
    return sqrt(difx + dify + difz); }


int main() {
    Point3D punto1 = {1, 2, 3};
    Point3D punto2 = {2, 3, 4};
    double distancia = Euclidean(&punto1, &punto2);
    printf("La distancia es: %.2f", distancia); }
```

EJERCICIO 3

Escriba un programa en C que defina un estructura para almacenar los datos de un rectángulo. Estos datos serían: dos números para representar la coordenada correspondiente al punto inferior izquierdo; el ancho; y el alto del rectángulo.

Su programa también debe capturar y almacenar las coordenadas de un punto en el plano.

Finalmente, el programa revisará, utilizando dos punteros a los datos anteriores, si el punto está contenido, o no, dentro del rectángulo.

SOLUCIÓN

```
int RectangleContainsPoint(const Point2D const * ptrPoint, const
Rectangle const * ptrRectangle) {

    if(ptrPoint->x >= (ptrRectangle->x) && ptrPoint->x <=
 (ptrRectangle->x + ptrRectangle->width))

 {

     if(ptrPoint->y >= (ptrRectangle->y) && ptrPoint->y <=
 (ptrRectangle->y + ptrRectangle->height)) {

     return 1;

    }

 }

 else {return 0;}}
```

TAREA 4

MUCHAS GRACIAS