

[Página Principal](#) / [Mis cursos](#) / [2024-2R4-4-201](#) / Examen integrador / [Ejercicio 1 - prácticos \(5 puntos\)](#)

Comenzado el	miércoles, 6 de noviembre de 2024, 18:52
Estado	Finalizado
Finalizado en	miércoles, 6 de noviembre de 2024, 20:01
Tiempo empleado	1 hora 9 minutos
Calificación	Sin calificar aún

Pregunta 1

Finalizado

Puntúa como 1,00

(1 punto)

El siguiente código fuente en lenguaje C está incompleto:

```
#include <stdio.h>

int ponerBitAlto(unsigned int *dato, int b)
{
    // completar
}

int main()
{
    unsigned int valor;
    int bit;

    printf("Ingrese un valor entero positivo: ");
    scanf("%u", &valor);
    printf("Ingrese el bit a modificar: ");
    scanf("%d", &bit);

    // completar
    if(                != -1 )
        printf("El valor modificado es: %u\n", valor);
    else
        printf("Error!\n");

    return 0;
}
```

Se pide:

- Completar la implementación de la función `ponerBitAlto()`.
- Completar la sentencia del condicional `if` que se encuentra dentro de la función `main()`.

Requerimientos de la función `ponerBitAlto()`:

- Retornar -1 en caso de recibir un puntero `NULL`.
- Retornar -1 si la variable `b` está fuera del rango permitido.
- Realizar la operación de poner en uno el bit indicado por el argumento `b` sobre la variable apuntada por el argumento `dato`, sin o se cumplen las condiciones anteriores.
- Retornar 1 si se realizó la conversión del dato.

```
#include <stdio.h>

int ponerBitAlto(unsigned int *dato, int b)
{
    if (b>31 || b<0)    {
        return -1;
    }
    if (dato==NULL){
        return -1;
    }

    *dato |= (1<<b);

    return 1;
}

int main()
{
    unsigned int valor;
    int bit;
```

Pregunta **2**

Finalizado

Puntúa como 1,00

(1 punto)

Implementar una función como plantilla, de nombre `obtenerMaximo(...)` que reciba como parámetro un arreglo estilo C (de tipos de datos básico como `int`, `float`, etc.) y el tamaño del arreglo. La función debe devolver el valor máximo del arreglo.

Nota: no se pide implementar función `main()`, solo la función plantilla.

```
template <typename T>
T obtenerMaximo (T arreglo, int tam)  {
    T suma=0;
    for (int i=0;i<tam;i++){
        suma+=arreglo[i];
    }
    return (suma/tam);
}
```

Pregunta **3**

Finalizado

Puntúa como 1,00

(2 puntos)

Implementar una clase en C++ que permita manipular vectores en 3 dimensiones, **Vector3d**. La clase debe tener:

- Datos miembros para almacenar las 3 coordenadas del vector (x, y, z).
- Constructor predeterminado (con valores por defecto).
- Funciones miembros inline:
 - Funciones miembros setter y getter para acceder a las coordenadas del vector.
- Funciones miembros no inline:
 - Función miembro **productoPunto(...)** que calcule el producto punto o escalar entre 2 vectores. El producto punto entre dos vectores u y v , con $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ está dado por $p = v_x \cdot u_x + v_y \cdot u_y + v_z \cdot u_z$
 - Función miembro **productoCruz(...)** que calcule el producto cruz o vectorial entre 2 vectores. El producto cruz entre dos vectores u y v , con $u = (u_x, u_y, u_z)$ y $v = (v_x, v_y, v_z)$ está dado por $w = (w_x, w_y, w_z)$ donde:
$$\begin{aligned}w_x &= u_y \cdot v_z - u_z \cdot v_y \\w_y &= u_z \cdot v_x - u_x \cdot v_z \\w_z &= u_x \cdot v_y - u_y \cdot v_x\end{aligned}$$
 - Función miembro **imprimir(...)** que muestre en la salida estándar el vector.

A continuación se muestra la función **main()** que permite evaluar la implementación de la clase:

```
#include <iostream>

using std::cout;
using std::endl;

// Clase Vector3d

int main()
{
    Vector3d v0, v1(1, 2, 3), v2(3, 2, 1);

    cout << "v0 = "; v0.imprimir();
    cout << endl;
    cout << "v1 = "; v1.imprimir();
    cout << endl;
    cout << "v2 = "; v2.imprimir();
    cout << endl;

    // Producto punto o escalar
    cout << "v1 . v2 = " << v1.productoPunto(v2) << endl;

    // Producto cruz o vectorial
    Vector3d v1_cruz_v2 = v1.productoCruz(v2);
    cout << "v1 x v2 = "; v1_cruz_v2.imprimir();
    cout << endl;

    return 0;
}
```

 [Class Vector3d.cpp](#)[◀ Diagramas en bloques de los trabajos grupales](#)

Ir a...

[Ejercicio 2 - ejercicios cortos \(3 puntos\) ▶](#)