Problemas

Tema 2: Algoritmos sobre matrices. (Primera Parte)

Vectores/matrices comunes o implementados usando contenedores.

Ejemplo previo

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1, 6};
    int pares = 0, impares = 0;

    for (int i = 0; i < n; i++)
        if (A[i] % 2 == 0) {
            pares++;
        }
        else {
           impares++;
        }
}
```

¿Está todo correcto en este ejemplo?

Ejercicio 1

a) Dado un vector identifique los elementos **múltiplos de 3**.
b) Inserte en un vector dinámico **el índice** de los múltiplos de 3 y en un segundo vector dinámico el valor de los no múltiplos de 3.
c) Realizar la suma de los elementos en posiciones **multiplos/nomultiplos**, comprobar también si el **índice del elemento** es par/impar.
d) Cálculo el módulo del vector de los elementos **nomultiplos.**

a)

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1, 6};
    int multiplo = 0, nomultiplo = 0;

    for (int i = 0; i < n; i++)

        if (A[i] % 3 == 0) {
            multiplo++;
        }
        else {
            nomultiplo++;
        }


    cout << "Multiplos de 3: " << multiplo <<
endl;
}
```

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1, 6};
    int multiplo = 0, nomultiplo = 0;
    int* T = new int[n];
    int* N = new int[n];
    for (int i = 0; i < n; i++)
        if (A[i] % 3 == 0) {
            multiplo++;
            T[i] = i;
        }
        else {
            nomultiplo++;
            N[i] = A[i];
        }


    cout << "Multiplos de 3: " << multiplo << endl;

    for (int i = 1; i <= multiplo; i++)
      cout << T[i] << " ";

    delete[] T, N;
}
```

¿Soluciona este código el inciso b) ?

b)

b)

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1, 6};
    int multiplo = 0, nomultiplo = 0;
    int* T = new int[n];
    int* N = new int[n];
    for (int i = 0; i < n; i++)
        if (A[i] % 3 == 0) {
            T[multiplo] = i;
            multiplo++;
        }
        else {
            N[nomultiplo] = A[i];
            nomultiplo++;
        }


    cout << "Multiplos de 3: " << multiplo << endl;

    for (int i = 1; i <= multiplo; i++)
      cout << T[i] << " ";

    delete[] T, N;
}
```

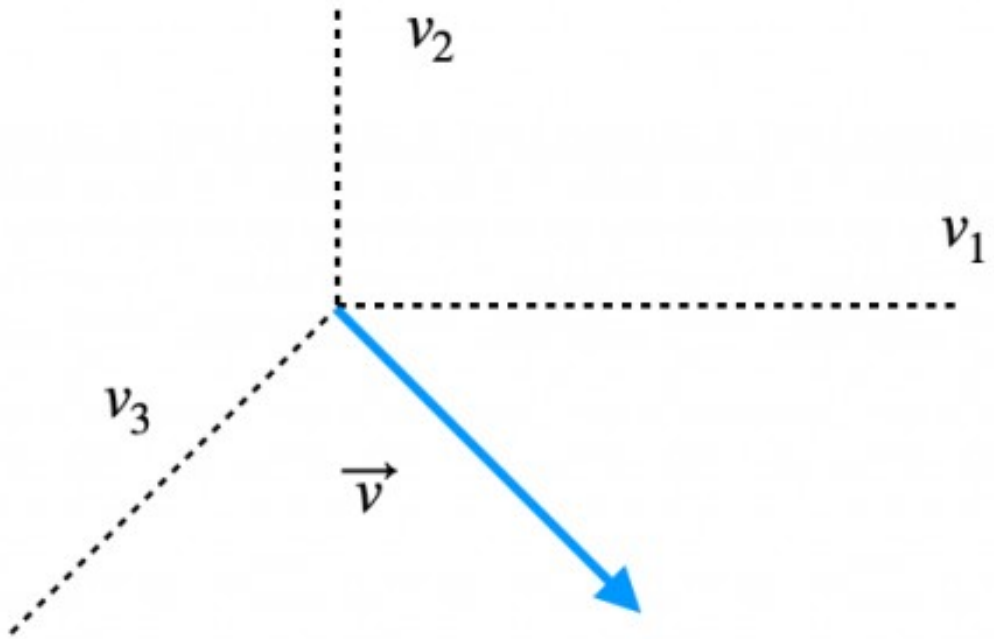
c)

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1, 6};
    int multiplo = 0, nomultiplo = 0;
    int multiplos_suma = 0, nomultiplos_suma = 0;
    int* T = new int[n];
    int* N = new int[n];

    for (int i = 0; i < n; i++) {
        if (i % 3 == 0) {
            multiplos_suma += A[i];
            T[multiplo++] = i;
        }
        else {
            nomultiplos_suma += A[i];
            N[nomultiplo++] = A[i];
        }
        if (i % 2 == 0)
            cout << "El índice " << i << " es par." << endl;
        else
            cout << "El índice " << i << " es impar." << endl;
    }

    delete[] T, N;

    return 0;
}
```

Módulo de un vector… Caso^{n} dimensional **d)**

$$\vec{v}=(v_1, v_2, v_3)$$

$$|\vec{v}| = \sqrt{v_1^2 + v_2^2 + v_3^2}$$



Módulo de un vector de tres dimensiones

Módulo de un vector… Caso n dimensional

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1,
6};
    int multiplo = 0, nomultiplo = 0;
    int* T = new int[n];
    int* N = new int[n];

    for (int i = 0; i < n; i++) {
        if (i % 3 == 0) {
            T[multiplo++] = i;
        }
        else {
            N[nomultiplo++] = A[i];

        }

    }
```

¿Soluciona este código el inciso d) ?

```
    float sumatoria = 0.0;
    for (int k = 0; k < n; k++) {
        sumatoria = sumatoria + N[k] * N[k];
    }

    cout << "El módulo del Vector es : " <<
sqrt(sumatoria) << endl;

    delete[] T, N;
}
```

Módulo de un vector… Caso n dimensional

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1,
6};
    int multiplo = 0, nomultiplo = 0;
    int* T = new int[n];
    int* N = new int[n];

    for (int i = 0; i < n; i++) {
        if (i % 3 == 0) {
            T[multiplo++] = i;
        }
        else {
            N[nomultiplo++] = A[i];

        }

    }
```

```
    float sumatoria = 0.0;
    for (int k = 0; k < nomultiplo; k++) {
        sumatoria = sumatoria + N[k] * N[k];
    }

    cout << "El módulo del Vector es : " <<
sqrt(sumatoria) << endl;

    delete[] T, N;
}
```

Módulo de un vector… Caso n dimensional

```
int main() {
    int A[n] = {5, 3, 2, 1, 5, 1, 4, 2, 1,
6};
    int multiplo = 0, nomultiplo = 0;
    int* T = new int[n];
    int* N = new int[n];
    float sumatoria = 0.0;

    for (int i = 0; i < n; i++) {
        if (i % 3 == 0) {
            T[multiplo++] = i;
        }
        else {
            N[nomultiplo++] = A[i];
            sumatoria += A[i];

        }

    }
```

```
    cout << "El módulo del Vector es : " <<
sqrt(sumatoria) << endl;

    delete[] T, N;
}
```

Ángulo entre dos vectores. ¿Cómo lo solucionamos ?

$$\cos(\alpha) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}| \cdot |\vec{v}|}$$

Ejercicios de Estudio Individual

Codificar en C++

- Recorrido un vector en orden inverso (del último al primer elemento) e impresión sus elementos (variante de "Recorrido de un vector (1)")
- Almacenamiento en un vector de los n primeros términos de la Sucesión de Fibonacci (variante de "Recorrido de un vector (1)")
- Búsqueda de un elemento dentro de un vector y devolución del índice en caso de encontrarlo (variante de "Búsqueda secuencial de un elemento")
- Contabilización de las ocurrencias de los elemento dentro de un vector que sean mayores o iguales que otro valor dado (variante de "Contabilización de las ocurrencias de un elemento dentro de un vector")
- Búsqueda del menor/mayor elemento de un vector y la posición donde se encuentra la última ocurrencia del menor/mayor, inicializando valores al primer elemento del vector (variante de "Búsqueda del menor/mayor elemento")

Codificar en C++

- Suma de los elementos de valores múltiplos de 3 e impares de un vector (variante de "Suma de los elementos de posiciones pares/impares de un vector")
- Implementación de una matriz sobre un vector almacenado los elementos ordenados por columnas (variante de "Implementación de una matriz sobre un vector)
- Recorrido de una matriz por columnas/filas e impresión de sus elementes en ese orden (impresión traspuesta) (variante de "Recorrido de una matriz por filas/columnas")
- Suma de los elementos con valor positivo de una matriz (variante de "Suma de los elementos tal que la suma de índices es par")
- Recorrido de la diagonal principal/secundaria incluso en matrices no cuadradas (variante de "Recorrido de la diagonal principal/secundaria")