

1. (25%) Un número oblongo es todo número natural que cumple que es el producto de dos naturales consecutivos. Realice la función **int oblongo(int n)** que retorne **1** si **n** es oblongo y **0** si no lo es.

```
int oblongo(int n)
{
    int factor;

    for (factor = 1; factor <= n / 2; factor ++ )
        if ( n % factor == 0 && n % (factor + 1) == 0 && factor * (factor+1) == n )
            return 1;
    return 0;
}
```

2. (25%) Realice una función **int overavg(int valores[], int n)** que recibe como parámetro un arreglo de **n** elementos enteros y retorne la cantidad de elementos que están por encima del promedio. Puede realizar tantas funciones adicionales como lo considere.

```
int overavg(int arr[], int n)
{
    int ind, total = 0, prom = promedio(arr,n);

    for ( ind = 0; ind < n; ind ++ )
        if ( arr[ind] > prom )
            total ++;

    return total;
}

float promedio(int arr[], int n)
{
    int ind;
    float total = 0;

    for ( ind = 0; ind < n; ind ++ )
        total += arr[ind];

    return total / n;
}
```

3. **(25%)** Dos cadenas son anagramas si contienen exactamente los mismos caracteres no importa el orden. **Escriba la función int anagramas(char \*s1, char \*s2)** que retorne 1 si “s1” y “s2” son anagramas y 0 si no lo son.

**Nota:** Existe la función **int contarchar( char s[ ], char c )** que retorna la cantidad de veces que se repite el caracter ‘c’ en la cadena ‘s’.

```
int anagramas( char s1[ ], char s2[ ] )
{
    int ind;

    if ( strlen(s1) != strlen(s2) )
        return 0;

    for ( ind = 0; s1[ind]; ind++ )
        if ( contarchar(s2, s1[ind]) != contarchar(s1, s1[ind]) )
            return 0;

    return 1;
}
```

4. **(25%)** Se desea saber en una matriz de números enteros la sumatoria de los números oblongos exceptuando los que se encuentran en la diagonal principal. Asuma que la función solicitada en el primer punto funciona perfectamente. Realice el prototipo de la función de la forma que lo crea conveniente.

```
int sumoblongos(int mat[][DIM], int d)
{
    int ind1, ind2, total = 0;

    for (ind1 = 0; ind1 < d; ind1 ++ )
        for ( ind2 = 0; ind2 < d; ind2 ++ )
            if ( ind1 != ind2 )
                if ( oblongo(mat[ind1][ind2]) )
                    total += mat[ind1][ind2];

    return total;
}
```