

Resuelva los siguientes problemas.

1. En una empresa de computadoras, los salarios de los empleados se van a aumentar según su sueldo actual:

Sueldo	Aumento en %
0 a 9000	20
9001 a 15000	10
15001 a 20000	5
Más de 20000	0

Realice una función que dado el sueldo de un empleado determine qué porcentaje de aumento le corresponde. **(15%)**

```
float porc_aumento(float sueldo)
{
    if ( sueldo <= 9000 )
        return 0.20;
    if ( sueldo <= 15000)
        return 0.10;
    if ( sueldo <= 20000)
        return 0.05;

    return 0;
}
```

2. **(20%)** El factorial de un número es el resultado de la multiplicación de todos los números menores o iguales que él. Por ejemplo: Factorial de 4 es igual $4 \times 3 \times 2 \times 1 = 24$. Realice la función **int factorial(int numero)**, que retorne el factorial del número “numero”.

```
int factorial(int numero)
{
    int mult = 1;

    while ( numero > 1 )
    {
        mult *= numero;  // mult = mult * numero;
        numero --;
    }

    return mult;
}
```

3. (25%) Se tiene un arreglo de **n** números flotantes de los cuales se desea obtener el total de la diferencia entre cada elemento y el promedio de los elementos del arreglo. Realice la función **float sum_dif_prom(float valores[], int n)**, que retorne lo requerido en la explicación anterior. Puede realizar cualquier otra función que necesite.

```
float sum_dif_prom(float valores[ ], int n)
{
    int ind;
    float total = 0, prom;

    prom = prom_val(valores,n);

    for ( ind = 0 ; ind < n; ind ++ )
        total += valores[ind] - prom;

    return total;
}

float prom_val(float val[], int n)
{
    int ind;

    float total = 0;

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

    return total / n;
}
```

4. (25%) Se conoce la función **int primo(int n)**, la cual retorna el valor 1 si el número “n” es primo y el valor 0 si el número “n” no es primo. Se desea realizar una función que retorne: 1 si hay más números primos por encima de la diagonal principal de la matriz que por debajo de la diagonal principal, 2 si hay menos números primos por encima de la diagonal principal que por debajo de la diagonal principal y 0 si hay la misma cantidad de números primos tanto por encima de la diagonal principal como por debajo de la diagonal principal.

```
int primosmat(int mat[][COL], int f, int c)
{
    int ind1, ind2, tot_prim_enc = 0, tot_prim_deb = 0;

    for (ind1 = 0 ; ind1 < f; ind1 ++ )
        for ( ind2 = 0; ind2 < c; ind2 ++ )
        {
            if ( ind1 < ind2 )
                tot_prim_enc += primo(mat[ind1][ind2]);
            else
                if ( ind1 > ind2 )
                    tot_prim_deb += primo(mat[ind1][ind2]);
        }

    if ( tot_prim_enc > tot_prim_deb )
        return 1;

    if ( tot_prim_enc < tot_prim_deb )
        return 2;

    return 0;
}
```

5. (15%) Realice la función **int sustchar(char cadena[], char charbus, char charsust)**, que sustituya en la cadena “cadena” el carácter “charbus” por el carácter “charsust” y retorne las cantidad de sustituciones realizadas.

```
int sustchar(char cadena[], char charbus, char charsust)
{
    int ind, cont=0;

    for ( ind = 0; cadena[ind] != '\0' ; ind ++ )
        if ( cadena[ind] == charbus )
        {
            cadena[ind] = charsust;
            cont++;
        }

    return cont;
}
```