

Ejercicios Resueltos - Vectores

- 1) Calcular la cantidad de alumnos que obtuvieron nota inferior al promedio del curso en cierta materia. Hay 20 alumnos, y todos rindieron. Las notas van del 0 al 100 (se asume que todas las notas son correctas).

```
#include<stdio.h>

int main(){
    int i, sum=0, nota[20], con=0;
    float prom;
    for(i=0;i<20;i++){
        printf("Ingrese la nota del alumno %d: ",(i+1));
        scanf("%d",&nota[i]);
        sum+=nota[i];
    }
    prom=1.0*sum/20;
    for(i=0;i<20;i++){
        if(nota[i]<prom) con++;
    }
    printf("\nLa cantidad de alumnos con nota inferior al promedio es:
%d",con);
    return 0;
}
```

- 2) Diseñar un algoritmo que obtenga el producto escalar de dos vectores (de enteros) de tamaño n .

```
#include<stdio.h>

int main(){
    int n,i;
    //Lectura de datos
    printf("Ingrese el tamaño de los vectores: ");
    scanf("%d",&n);
    int A[n],B[n];
    printf("\nIngrese los elementos del vector A:\n");
    for(i=0;i<n;i++){
        printf("A[%d]: ",i);
        scanf("%d",&A[i]);
    }
    printf("\nIngrese los elementos del vector B:\n");
    for(i=0;i<n;i++){
        printf("B[%d]: ",i);
        scanf("%d",&B[i]);
    }
    //Proceso
    int prod=0;
    for(i=0;i<n;i++) prod+=(A[i]*B[i]);
    //Impresión del resultado
    printf("\nEl producto escalar es: %d\n",prod);
    return 0;
}
```

- 3) Considere los 100 primeros números naturales. El programa debe devolver una matriz lógica (0=falso, 1=verdadero) donde se indique si el elemento i es o no primo.

```
#include<stdio.h>

int main(){
    int i, j, primo[101];
    for(i=0;i<=100;i++) primo[i]=1;
    //Se usará el metodo de Eratóstenes
    for(i=2;(i*i)<=100;i++){
        if(primo[i]){
            for(j=(i*i);j<=100;j+=i) primo[j]=0;
        }
    }
    //impresion de resultados
    printf("\nLa tabla de numeros primos del 1 al 100 es:\n");
    for(i=0;i<10;i++){
        for(j=0;j<10;j++){
            printf("%d ",primo[i*10+j+1]);
        }
        printf("\n");
    }
    return 0;
}
```

- 4) Se cuenta con una lista L1 de números enteros en el cual existen numerosos valores repetidos. A fin de economizar el espacio de almacenamiento, se desea crear una nueva lista L2 en la cual cada valor diferente aparece una sola vez, sin repetición, pero indicando la cantidad de veces que se repite dicho valor en la lista L1.

Ejemplo:

Lista original L1 (dato para el algoritmo)

23	27	8	14	23	23	8	23	27	23	27	8	27
----	----	---	----	----	----	---	----	----	----	----	---	----

Lista final L2 (resultado del algoritmo)

8	3	14	1	23	5	27	4
---	---	----	---	----	---	----	---

La interpretación de la lista L2 es como sigue: el valor 8 aparece 3 veces en L1, el valor 14 aparece 1 vez en L1, el valor 23 aparece 5 veces en L1, y el valor 27 aparece 4 veces en L1.

Note que los valores de L1 deben aparecer ordenados ascendentemente en L2 (8-14-23-27).

Escriba un algoritmo en C que reciba la lista original L1 y obtenga la lista L2 con las características indicadas.

```
#include<stdio.h>

int main(){
    int i, n, aux, j;
    printf("Ingrese el tamanho del vector: ");
    scanf("%d",&n);
```

```

int vec[n];
printf("Ingrese los elementos del vector:\n");
for(i=0;i<n;i++){
    printf("vec[%d]: ",i);
    scanf("%d",&vec[i]);
}
//Ordenamiento del vector
int band=1;j=1;
while(band){
    band=0;
    for(i=0;i<(n-j);i++){
        if(vec[i]>vec[i+1]){
            band=1;
            aux=vec[i];
            vec[i]=vec[i+1];
            vec[i+1]=aux;
        }
    }
    j++;
}
//Proceso
int sol[(2*n)];
int cant=1, pos=0;
int act = vec[0];
for(i=1; i<n; i++){
    if (vec[i] == act) cant++;
    else {
        sol[pos] = act;
        sol[pos+1] = cant;
        pos += 2;
        act = vec[i];
        cant = 1;
    }
}
sol[pos] = act;
sol[pos+1] = cant;
pos += 2;
printf("\nImpresion del resultado:\n");
for(i=0;i<pos;i++) printf("%d\t",sol[i]);
return 0;
}

```