

/* Un archivo de texto con tipo (ventas.txt) contiene las ventas realizadas por todos los vendedores de una empresa. Escriba una función por cada inciso para:

a) Calcular el total vendido. */

```
float totalVtas(FILE *arch)
{
    float vta, total=0;

    while(fscanf(arch, "%f", &vta) > 0)
        total += vta;

    return(total);
}
```

/* b) Calcular el promedio de ventas */

```
float promVtas(FILE *arch)
{
    float vta, prom, total=0;
    int cont=0;

    while(fscanf(arch, "%f", &vta) > 0)
    {
        cont++;
        total += vta;
    }

    prom= total / cont;

    return(prom);
}
```

/* c) Encontrar el monto de la mayor venta */

```
float mayorVtas(FILE *arch)
{
    float vta, sMayor=0;

    while(fscanf(arch, "%f", &vta) > 0)
        if(vta > sMayor)
            sMayor=vta;

    return(sMayor);
}
```

/* Un archivo de texto, almacena base y altura de un conjunto de triángulos. Escriba una función para generar un segundo archivo de texto que contenga las áreas de cada triángulo. */

```
void archAreas(FILE *arch, FILE *archArea)
{
    float base, altura, area;

    while(fscanf(arch, "%f%f", &base, &altura) > 0)
    {
        area = base * altura / 2;
        fprintf(archArea, "%.2f ", area);
    }
}
```

```
/* Dos archivos de texto almacenan base y altura de un conjunto de triángulos.
```

```
    Escriba una función que genere un 3er. archivo de texto que contenga base, altura y área de cada triángulo. */
```

```
void archAreas(FILE *archBase, FILE *archAltura, FILE *archArea)
{
    float base, altura, area;

    while(fscanf(archBase, "%f", &base) > 0)
    {
        fscanf(archAltura, "%f", &altura);
        area = base * altura / 2;
        fprintf(archArea, "%.2f %.2f %.2f\n", base, altura, area);
    }
}
```

```
/* Un archivo de texto con formato contiene nombre y 3 calificaciones para un grupo
```

```
    de alumnos. Escriba una función para generar un archivo de texto que contenga nombre, promedio y estatus (aprobado/reprobado) para cada alumno.
*/
```

```
void alumnos(FILE *arch, FILE *nwArch)
{
    char nom[25], status[10];
    float call, cal2, cal3, prom;

    while(fgets(nom, 25, arch) > 0)
    {
        fscanf(arch, "%f%f%f", &call, &cal2, &cal3);
        prom = (call+cal2+cal3)/3;
        if(prom<6)
            strcpy(status, "Reprobado");
        else
            strcpy(status, "Aprobado");

        fprintf(nwArch, "%s\n%.2f%s\n", nom, prom, status);
    }
}
```

```
/* Escriba una función para agregar un dato entero a un archivo.
```

```
Nota: La función
```

```
    recibe el nombre del archivo (una cadena) no el id. */
```

```
int agregaDato(char *nomArch, int dato)
{
    int res;
    FILE *arch;

    res = abreArch(&arch, nomArch, "a");
    if(res)
        fprintf(arch, "%d ", dato);

    fclose(arch);

    return(res);
}
```