

20221205

```
/*PARTE TEORICA
int queQuiereDecir (int x[], int tam)
{if (tam==1)
    return x[0];
else
    return x[tam-1] + queQuiereDecir (x, tam-1);};
int main ()
{ const int tamArreglo=10;
  int a[tamArreglo]= {2,3,4,5,6,5,4,3,2};
  int resultado= queQuiereDecir (a, tamArreglo);
  cout << resultado
  return 0; }
```

1. Dado el siguiente programa, ¿que hace?
2. Una función puede retornar más de un valor en su nombre
3. Un ciclo de repetición for puede reemplazar un ciclo do...while en cualquier contexto*/

1. El programa anterior devuelve la suma total de los numeros del vector.
En el caso de que sea un vector de una posición devuelve el único valor.
2. Falso. Por declaración devuelve una función.
3. Falso. En el do while la acción se ejecuta antes de evaluar la condición, en cambio en el for primero se evalúa y luego se ejecuta.

/*PARTE PRÁCTICA

La materia AyED dispone para cada curso una matriz de enteros, de dos dimensiones, con sesenta filas (máxima cantidad de alumnos por curso) y siete columnas con las notas de los estudiantes inscriptos en el curso en el orden siguiente (1er Parcial, 2do Parcial, 1er Rec 1er Parcial, 1er Rec 2do parcial, 2do Rec 1er parcial, 2do Rec 2do Parcial, Trabajo practico).

Con las notas de cada instancia ya cargadas.

Ademas dispone de un vector paralelo (las posiciones se corresponden una a una con las filas de la matriz) con los datos personales de los inscriptos (Legajo un entero, apellido y nombre cadena de caracteres) , ordenado por legajo.

1. Se pide desarrollar una función que muestre por pantalla el listado de la situación académica de cada estudiante, ordenado por apellido y nombre del estudiante, por bloques máximos de 20 por hoja:
Orden Apellido situación académica
1 xxxxxxxxxxxx promocionado
2 xxxxxxxxxxxx regularizado
3 xxxxxxxxxxxx no aprobado
Promociona aquel estudiante con notas >= 8 en ambos parciales
(o en 1er recuperatorio de uno de ellos y en TP
Regulariza con Nota >= 6 en un parcial, nota 6 o 7 en otro y >6 en TP,
en cualquier instancia de evaluación, incluido TP
No aprobado con Nota 0 (cero) en TP (no entregado)
u otra situación no definida anteriormente.
2. Declare el prototipo y una invocación completa de la función del punto anterior (1 punto)
3. Declare las estructuras de datos necesarias para la solución de la propuesta (1 punto)

```
struct Alumno //Ordenado por legajo
{int legajo;
 char apellido[20];
 char nombre[20]}

void emitirListado (int m[][70], Alumno v[])
{int i=0;
 int p=0;
 cout << "Orden" << "Apellido" << "Situación académica" << endl;
 cout << "-PAGINA Nro." << p+1 << ":" << endl;
 while (p<4)
 {
 for (i=0; i<20; i++)
 {cout << i+(20*p) << v[i+(20*p)].apellido;
  if((m[0][i+(20*p)]>=8 && m[1][i+(20*p)]>=8)
      || (m[0][i+(20*p)]>=8 && m[3][i+(20*p)]>=8)
```

```

        || (m[1][i+(20*p)]>=8 && m[2][i+(20*p)]>=8)
        && (m[7][i+(20*p)] >= 8))
    cout << "promocionado" << endl;
else
    {if ((6 < m[0][i+(20*p)] && m[0][i+(20*p)] < 8) || (6 < m[2][i+(20*p)] && m[2][i+(20*p)] < 8) || m[4][i+(20*p)]>6)
        && ((6 < m[1][i+(20*p)] && m[1][i+(20*p)]< 8)|| ( 6 < m[3][i+(20*p)] && m[3][i+(20*p)]< 8 )|| m[6][i+(20*p)]>6)
        && (m[7][i+(20*p)] > 6))
        cout << "aprobado" << endl;
    else
        cout <<"no aprobado" << endl;}}
p++;}
return;}

int main ()
{int matriz[7][70];
  Alumno alumnos[70];
emitirlistado (matriz, alumnos);
return; }

```