

```
#include <iostream>
#include <conio.h>
#include <stdio.h>
#include <iomanip>

#define CAM 10
#define ETA 5

using namespace std;

int main() {

    // Variables
    int i, j, idCamioneta;
    char indicador;

    // Vectores
    int idCam[CAM]; // vector asociado a fila
    int tiempoEt[ETA]; // vector asociado a columnas
    int contEtFin[CAM]; // vector asociado a fila

    int matRally[CAM][ETA];

    // Inicialización de la matriz y otros vectores
    for (i = 0; i < CAM; i++) {
        contEtFin[i] = 0;
        for (j = 0; j < ETA; j++) {
            matRally[i][j] = 0;
        }
    }

    // Carga de datos ID camioneta
    for (i = 0; i < CAM; i++) {
        cout << "Ingrese el ID de la camioneta (500-1200): ";
        cin >> idCam[i];

        // Validación
        while (idCam[i] < 500 || idCam[i] > 1200) {
            cout << "ERROR! Ingrese un ID válido: ";
            cin >> idCam[i];
        }
    }

    // Carga de datos de tiempos de las etapas
    for (j = 0; j < ETA; j++) {
        cout << "Ingrese el tiempo de la etapa " << j << " (en minutos): ";
        cin >> tiempoEt[j];
    }

    cout << endl;
```

```

    cout << "Ingrese el ID de la camioneta (500-1200) o 0 para finalizar:
";
    cin >> idCamioneta;

    // Bucle para ingresar datos de rallies
    while (idCamioneta != 0) {
        int f = 0;
        while (f < CAM && idCamioneta != idCam[f]) {
            f++;
        }
        if (f < CAM) {
            // Ingreso de datos del rally
            cout << "Ingrese la etapa (0-4): ";
            cin >> j;
            cout << "Ingrese el tiempo (en minutos): ";
            cin >> matRally[f][j];
        }
        cout << endl;
        cout << "Ingrese el ID de la camioneta (500-1200) o 0 para
finalizar: ";
        cin >> idCamioneta;

        // Validación
        while (idCamioneta != 0 && (idCamioneta < 500 || idCamioneta >
1200)) {
            cout << "ERROR! Ingrese un ID válido o 0 para finalizar: ";
            cin >> idCamioneta;
        }
    }

    // 1. Por cada camioneta: tiempo en cada etapa, ID de la camioneta,
impresión matricial o 0 si no cumplió
    cout << endl;
    cout << "ID Camioneta - Etapa 0 - Etapa 1 - Etapa 2 - Etapa 3 - Etapa
4" << endl;
    cout << "-----"
<< endl;

    for (i = 0; i < CAM; i++) {
        cout << setw(11) << idCam[i] << " - ";
        for (j = 0; j < ETA; j++) {
            cout << setw(7) << matRally[i][j] << " - ";
        }
        cout << endl;
    }

    // 2. Por cada camioneta, indicador que informe si tardó (-, =, +) en
cada etapa
    cout << endl;
    cout << "ID Camioneta - Etapa 0 - Etapa 1 - Etapa 2 - Etapa 3 - Etapa
4" << endl;
    cout << "-----"
<< endl;

```

```
for (i = 0; i < CAM; i++) {
    cout << setw(11) << idCam[i] << " - ";
    for (j = 0; j < ETA; j++) {
        if (matRally[i][j] < tiempoEt[j]) {
            indicador = '-';
        } else if (matRally[i][j] > tiempoEt[j]) {
            indicador = '+';
        } else {
            indicador = '=';
        }
        cout << setw(7) << indicador << " - ";
    }
    cout << endl;
}

// 3. Por cada camioneta, la cantidad de etapas finalizadas
cout << endl;
cout << "Cantidad de etapas finalizadas por camioneta: " << endl;
for (i = 0; i < CAM; i++) {
    for (j = 0; j < ETA; j++) {
        if (matRally[i][j] > 0) {
            contEtFin[j]++;
        }
    }
    cout << "ID Camioneta " << idCam[i] << " - Finalizó " <<
contEtFin[j] << " etapas" << endl;
}

// 4. Máximo y mínimo en cada etapa

// Max
for (j = 0; j < ETA; j++) {
    int maxEt = 0;
    for (i = 0; i < CAM; i++) {
        if (matRally[i][j] > matRally[maxEt][j]) {
            maxEt = i;
        }
    }
    cout << "Etapa " << j << ": Máximo tiempo - ID Camioneta " <<
idCam[maxEt] << " - " << matRally[maxEt][j] << " minutos" << endl;
}

// Min
for (j = 0; j < ETA; j++) {
    int minEt = 0;
    for (i = 0; i < CAM; i++) {
        if (matRally[i][j] < matRally[minEt][j]) {
            minEt = i;
        }
    }
    cout << "Etapa " << j << ": Mínimo tiempo - ID Camioneta " <<
idCam[minEt] << " - " << matRally[minEt][j] << " minutos" << endl;
}
```

```
// 5. Promedio de tiempos, solo para vehículos que finalizaron
for (j = 0; j < ETA; j++) {
    int cont = 0;
    int acum = 0;
    for (i = 0; i < CAM; i++) {
        if (matRally[i][j] > 0) {
            cont++;
            acum += matRally[i][j];
        }
    }
    if (cont > 0) {
        cout << "Promedio de tiempos para etapa " << j << ": " <<
(acum / cont) << " minutos" << endl;
    }
}

getch();
return 0;
}
```