



EXAMEN PARCIAL 1

Alyx Claire Rodriguez Cicero

Código en C++:

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>

using namespace std;

const int NUM_PRODUCTOS = 100;
const int NUM_MESES = 12;

const string NOMBRES_MESES[NUM_MESES] = {
    "Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
    "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre", "Diciembre"
};

void calcularVentasProductos(const vector<vector<int>>& ventas, vector<int>&
ventasProductos) {
    for (int i = 0; i < NUM_PRODUCTOS; ++i) {
        ventasProductos[i] = 0;
        for (int j = 0; j < NUM_MESES; ++j) {
            ventasProductos[i] += ventas[i][j];
        }
    }
}

void calcularVentasMeses(const vector<vector<int>>& ventas, vector<int>&
ventasMeses) {
    for (int j = 0; j < NUM_MESES; ++j) {
        ventasMeses[j] = 0;
        for (int i = 0; i < NUM_PRODUCTOS; ++i) {
            ventasMeses[j] += ventas[i][j];
        }
    }
}

void productoMasVendidoPorMes(const vector<vector<int>>& ventas,
vector<int>& productoMasVendido) {
    for (int j = 0; j < NUM_MESES; ++j) {
        int maxVentas = -1;
        for (int i = 0; i < NUM_PRODUCTOS; ++i) {
            if (ventas[i][j] > maxVentas) {
                maxVentas = ventas[i][j];
                productoMasVendido[j] = i;
            }
        }
    }
}
```

```

    }
}

void imprimirResultados(const vector<int>& ventasProductos, const
vector<int>& ventasMeses, const vector<int>& productoMasVendido, const
vector<vector<int>>& ventas) {
    cout << "Programa creado por Alyx Claire" << endl;
    cout << "Tabla de Resultados:" << endl;
    cout << "-----"
<< endl;
    cout << "| Producto | Total Ventas |";
    for (int j = 0; j < NUM_MESES; ++j) {
        cout << " " << NOMBRES_MESES[j] << " |";
    }
    cout << endl;
    cout << "-----"
<< endl;

    // Imprimir datos de cada producto
    for (int i = 0; i < NUM_PRODUCTOS; ++i) {
        cout << "|      " << i + 1 << "      |      " << ventasProductos[i] <<
"
        |";
        for (int j = 0; j < NUM_MESES; ++j) {
            cout << "      " << ventas[i][j] << "      |";
        }
        cout << endl;
    }
    cout << "-----"
<< endl;

    // Imprimir totales de ventas por mes
    cout << "|      Total      |";
    for (int j = 0; j < NUM_MESES; ++j) {
        cout << "      " << ventasMeses[j] << "      |";
    }
    cout << endl;
    cout << "-----"
<< endl;

    // Imprimir productos más vendidos por mes
    cout << "| Más Vend. |";
    for (int j = 0; j < NUM_MESES; ++j) {
        cout << " Prod " << productoMasVendido[j] + 1 << " |";
    }
}

```

```

        cout << endl;
        cout << "-----"
<< endl;
    }

int main() {
    vector<vector<int>> ventas(NUM_PRODUCTOS, vector<int>(NUM_MESES, 0));

    // Simulación de datos de ventas
    for (int i = 0; i < NUM_PRODUCTOS; ++i) {
        for (int j = 0; j < NUM_MESES; ++j) {
            ventas[i][j] = rand() % 100; // Número aleatorio entre 0 y 99
        }
    }

    vector<int> ventasProductos(NUM_PRODUCTOS);
    vector<int> ventasMeses(NUM_MESES);
    vector<int> productoMasVendido(NUM_MESES);

    calcularVentasProductos(ventas, ventasProductos);
    calcularVentasMeses(ventas, ventasMeses);
    productoMasVendidoPorMes(ventas, productoMasVendido);

    imprimirResultados(ventasProductos, ventasMeses, productoMasVendido,
ventas);

    return 0;
}

```

Salida de escritorio:

● Programa creado por Alyx Claire

Tabla de Resultados:

Producto	Total Ventas	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre
1	547	41	67	34	0	69	24	78	58	62	64	5	45
2	610	81	27	61	91	95	42	27	36	91	4	2	53
3	587	92	82	21	16	18	95	47	26	71	38	69	12
4	553	67	99	35	94	3	11	22	33	73	64	41	11
5	598	53	68	47	44	62	57	37	59	23	41	29	78
6	522	16	35	90	42	88	6	40	42	64	48	46	5
7	577	90	29	70	50	6	1	93	48	29	23	84	54
8	484	56	40	66	76	31	8	44	39	26	23	37	38
9	432	18	82	29	41	33	15	39	58	4	30	77	6
10	679	73	86	21	45	24	72	70	29	77	73	97	12
11	707	86	90	61	36	55	67	55	74	31	52	50	50
12	583	41	24	66	30	7	91	7	37	57	87	53	83
13	415	45	9	9	58	21	88	22	46	6	30	13	68
14	605	0	91	62	55	10	59	24	37	48	83	95	41
15	689	2	50	91	36	74	20	96	21	48	99	68	84
16	626	81	34	53	99	18	38	0	88	27	67	28	93
17	324	48	83	7	21	10	17	13	14	9	16	35	51
18	401	0	49	19	56	98	3	24	8	44	9	89	2
19	567	95	85	93	43	23	87	14	3	48	0	58	18
20	832	80	96	98	81	89	98	9	57	72	22	38	92
21	619	38	79	90	57	58	91	15	88	56	11	2	34
22	628	72	55	28	46	62	86	75	33	69	42	44	16
23	773	81	98	22	51	21	99	57	76	92	89	75	12
24	486	0	10	3	69	61	88	1	89	55	23	2	85
25	652	82	85	88	26	17	57	32	32	69	54	21	89
26	586	76	29	68	92	25	55	34	49	41	12	45	60
27	625	18	53	39	23	79	96	87	29	49	37	66	49
28	666	93	95	97	16	86	5	88	82	55	34	14	1
29	539	16	71	86	63	13	55	85	53	12	8	32	45
30	609	13	56	21	58	46	82	81	44	96	22	29	61
31	599	35	50	73	66	44	59	92	39	53	24	54	10
32	544	41	67	34	0	69	24	78	58	62	64	5	45
33	610	81	27	61	91	95	42	27	36	91	4	2	53
34	587	92	82	21	16	18	95	47	26	71	38	69	12
35	553	67	99	35	94	3	11	22	33	73	64	41	11
36	598	53	68	47	44	62	57	37	59	23	41	29	78
37	522	16	35	90	42	88	6	40	42	64	48	46	5
38	577	90	29	70	50	6	1	93	48	29	23	84	54
39	484	56	40	66	76	31	8	44	39	26	23	37	38
40	432	18	82	29	41	33	15	39	58	4	30	77	6
41	679	73	86	21	45	24	72	70	29	77	73	97	12
42	707	86	90	61	36	55	67	55	74	31	52	50	50
43	583	41	24	66	30	7	91	7	37	57	87	53	83
44	415	45	9	9	58	21	88	22	46	6	30	13	68
45	605	0	91	62	55	10	59	24	37	48	83	95	41
46	689	2	50	91	36	74	20	96	21	48	99	68	84
47	626	81	34	53	99	18	38	0	88	27	67	28	93
48	324	48	83	7	21	10	17	13	14	9	16	35	51
49	401	0	49	19	56	98	3	24	8	44	9	89	2
50	567	95	85	93	43	23	87	14	3	48	0	58	18
51	832	80	96	98	81	89	98	9	57	72	22	38	92
52	619	38	79	90	57	58	91	15	88	56	11	2	34
53	628	72	55	28	46	62	86	75	33	69	42	44	16
54	773	81	98	22	51	21	99	57	76	92	89	75	12
55	486	0	10	3	69	61	88	1	89	55	23	2	85
56	652	82	85	88	26	17	57	32	32	69	54	21	89
57	586	76	29	68	92	25	55	34	49	41	12	45	60
58	625	18	53	39	23	79	96	87	29	49	37	66	49
59	666	93	95	97	16	86	5	88	82	55	34	14	1
60	539	16	71	86	63	13	55	85	53	12	8	32	45
61	609	13	56	21	58	46	82	81	44	96	22	29	61
62	599	35	50	73	66	44	59	92	39	53	24	54	10
63	544	41	67	34	0	69	24	78	58	62	64	5	45
64	610	81	27	61	91	95	42	27	36	91	4	2	53
65	587	92	82	21	16	18	95	47	26	71	38	69	12
66	553	67	99	35	94	3	11	22	33	73	64	41	11
67	598	53	68	47	44	62	57	37	59	23	41	29	78
68	522	16	35	90	42	88	6	40	42	64	48	46	5
69	577	90	29	70	50	6	1	93	48	29	23	84	54
70	484	56	40	66	76	31	8	44	39	26	23	37	38
71	432	18	82	29	41	33	15	39	58	4	30	77	6
72	679	73	86	21	45	24	72	70	29	77	73	97	12
73	707	86	90	61	36	55	67	55	74	31	52	50	50
74	583	41	24	66	30	7	91	7	37	57	87	53	83
75	415	45	9	9	58	21	88	22	46	6	30	13	68
76	605	0	91	62	55	10	59	24	37	48	83	95	41
77	689	2	50	91	36	74	20	96	21	48	99	68	84
78	626	81	34	53	99	18	38	0	88	27	67	28	93
79	324	48	83	7	21	10	17	13	14	9	16	35	51
80	401	0	49	19	56	98	3	24	8	44	9	89	2
81	567	95	85	93	43	23	87	14	3	48	0	58	18
82	832	80	96	98	81	89	98	9	57	72	22	38	92
83	619	38	79	90	57	58	91	15	88	56	11	2	34
84	628	72	55	28	46	62	86	75	33	69	42	44	16
85	773	81	98	22	51	21	99	57	76	92	89	75	12
86	486	0	10	3	69	61	88	1	89	55	23	2	85
87	652	82	85	88	26	17	57	32	32	69	54	21	89
88	586	76	29	68	92	25	55	34	49	41	12	45	60
89	625	18	53	39	23	79	96	87	29	49	37	66	49
90	666	93	95	97	16	86	5	88	82	55	34	14	1
91	539	16	71	86	63	13	55	85	53	12	8	32	45
92	609	13	56	21	58	46	82	81	44	96	22	29	61
93	599	35	50	73	66	44	59	92	39	53	24	54	10
94	544	41	67	34	0	69	24	78	58	62	64	5	45
95	610	81	27	61	91	95	42	27	36	91	4	2	53
96	587	92	82	21	16	18	95	47	26	71	38	69	12
97	553	67	99	35	94	3	11	22	33	73	64	41	11
98	598	53	68	47	44	62	57	37	59	23	41	29	78
99	522	16	35	90	42	88	6	40	42	64	48	46	5
100	577	90	29	70	50	6	1	93	48	29	23	84	54
101	484	56	40	66	76	31	8	44	39	26	23	37	38
102	432	18	82	29	41	33	15	39	58	4	30	77	6
103	679	73	86	21	45	24	72	70	29	77	73	97	12
104	707	86	90	61	36	55	67	55	74	31	52	50	50
105	583	41	24	66	30	7	91	7	37	57	87	53	83
106	415	45	9	9	58	21	88	22	46	6	30	13	68
107	605	0	91	62	55	10	59	24	37	48	83	95	41
108	689	2	50	91	36	74	20	96	21	48	99	68	84
109	626	81	34	53	99	18	38	0	88	27	67	28	93
110	324	48	83	7	21	10	17	13	14	9	16	35	51
111	401	0	49	19	56	98	3	24	8	44	9	89	2
112	567	95	85	93	43	23	87	14	3	48	0	58	18
113	832	80	96	98	81	89	98	9	57	72	22	38	92
114	619	38	79	90	57	58	91	15	88	56	11	2	34
115	628	72	55	28	46	62	86	75	33	69	42	44	16
116	773	81	98	22	51	21	99	57	76	92	89	75	12
117	486	0	10	3	69	61	88	1	89	55	23	2	85
118	652	82	85	88	26	17	57	32	32	69	54	21	89
119	586	76	29	68	92	25	55	34	49	41	12	45	60
120	625	18	53	39	23	79	96	87	29	49</			

80	676	12	88	82	54	85	10	84	74	80	15	51	41
81	638	15	79	10	98	73	88	77	32	56	89	13	8
82	708	41	90	23	63	28	84	78	0	71	85	74	71
83	696	33	67	53	95	68	25	76	29	50	98	9	93
84	537	86	80	16	49	67	28	79	64	21	5	26	16
85	634	16	26	66	87	81	64	40	86	21	62	21	64
86	399	9	15	2	73	24	41	45	62	23	31	6	68
87	356	18	2	7	7	81	12	36	30	14	9	84	56
88	634	90	93	96	52	54	45	8	48	91	12	31	14
89	563	39	58	22	4	95	52	69	79	38	23	18	66
90	714	59	98	86	96	62	33	58	22	46	92	37	25
91	611	47	58	2	7	98	30	92	0	78	99	52	48
92	668	82	40	15	75	62	67	36	97	18	97	28	51
93	491	16	30	49	25	58	29	20	40	60	47	62	55
94	744	75	92	61	54	98	46	14	46	88	69	38	63
95	521	75	15	21	75	15	28	34	70	5	64	57	62
96	447	61	24	49	69	30	23	50	33	25	10	37	36
97	543	37	78	93	36	14	64	91	49	35	5	37	4
98	707	37	23	64	70	8	68	81	85	52	73	52	94
99	636	76	26	96	72	49	40	74	19	43	11	41	89
100	561	19	65	5	85	16	50	15	9	64	66	93	74

Total	4722	5297	4685	5240	4884	4754	4502	4729	4779	4956	4461	4781	

Mis Vend.	Prod 48	Prod 4	Prod 20	Prod 16	Prod 18	Prod 23	Prod 15	Prod 57	Prod 30	Prod 15	Prod 59	Prod 98	

& "C:\Users\ashle\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugadapters\vsdbg\bin\vsdebugconsole.exe" '\\.\p													