Q1.

```
 #include <iostream>
#include <vector>
#include <queue>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int t;
    cin >> t;
    while (t--) {
        int m, x;
        cin >> m >> x;
        vector<int> costs(m);
        for (int i = 0; i < m; ++i) {
            cin >> costs[i];
        }

        priority_queue<int, vector<int>, greater<int>> minHeap;
        int availableMoney = 0;
        int happinessUnits = 0;
        for (int month = 0; month < m; ++month) {
            // Add this month's cost to the priority queue
            minHeap.push(costs[month]);

            // Earn the salary for this month
            availableMoney += x;
            // Buy the cheapest available happiness if possible

            if (!minHeap.empty() && availableMoney >= minHeap.top()) {

                availableMoney -= minHeap.top();

                minHeap.pop();

                happinessUnits += 1;

            }

        }

        cout << happinessUnits << '\n';

    }

    return 0;

}
```

Explanation:

- Used a priority queue to store the happiness costs.
- For each month, add the current month's cost to the priority queue.
- Increment the available money by the monthly salary.
- If the available money is sufficient to buy the cheapest happiness unit (top of the heap), buy it and increase the happiness counter.