|  |  |
| --- | --- |
| **PowerSet in C++** | |
| #include <iostream>  using namespace std;  void generatePowerSet(char set[], int n) {  for (int i = 0; i < (1 << n); i++) {  cout << "{ ";  for (int j = 0; j < n; j++) {  if (i & (1 << j)) {  cout << set[j] << " ";  }  }  cout << "}" << endl;  }  }  int main() {  char set[] = {'a', 'b', 'c'};  int n = sizeof(set) / sizeof(set[0]);  generatePowerSet(set, n);    return 0;  } | **Dry Run Example:**  Let's dry run this with the set {'a', 'b', 'c'}. The set has 3 elements, so n = 3 and the total number of subsets will be 2^3 = 8.   | **i** | **i in binary** | **Subset representation (bits set)** | **Subset generated** | | --- | --- | --- | --- | | 0 | 000 | None | { } | | 1 | 001 | 3rd bit set (only c) | { c } | | 2 | 010 | 2nd bit set (only b) | { b } | | 3 | 011 | 2nd & 3rd bits set (b, c) | { b c } | | 4 | 100 | 1st bit set (only a) | { a } | | 5 | 101 | 1st & 3rd bits set (a, c) | { a c } | | 6 | 110 | 1st & 2nd bits set (a, b) | { a b } | | 7 | 111 | All bits set (a, b, c) | { a b c } |   **Output:**  { }  { c }  { b }  { b c }  { a }  { a c }  { a b }  { a b c } |
| { }  { a }  { b }  { a b }  { c }  { a c }  { b c }  { a b c } | |