**Approximation Algorithms**

**BinPacking Algorithm**

1. **#include<stdio.h>**
2. **void binPacking(int \*a, int size, int n) {**
3. **int binCount = 1, i;**
4. **int s = size;**
5. **for (i = 0; i < n; i++) {**
6. **if (s - \*(a + i) > 0) {**
7. **s -= \*(a + i);**
8. **continue;**
9. **} else {**
10. **binCount++;**
11. **s = size;**
12. **i--;**
13. **}**
14. **}**
16. **printf("Number of bins required: %d", binCount);**
17. **}**
19. **int main(int argc, char \*\*argv) {**
20. **printf("Enter the number of items in Set: ");**
21. **int n;**
22. **int a[n], i;**
23. **int size;**
24. **scanf("%d", &n);**
25. **printf("Enter %d items:", n);**
27. **for (i = 0; i < n; i++)**
28. **scanf("%d", &a[i]);**
29. **printf("Enter the bin size: ");**
30. **scanf("%d", &size);**
31. **binPacking(a, size, n);**
32. **return 0;**
33. **}**

[**https://www.sanfoundry.com/c-program-implement-bin-packing-algorithm/**](https://www.sanfoundry.com/c-program-implement-bin-packing-algorithm/)

[**https://www.tutorialspoint.com/bin-packing-problem-minimize-number-of-used-bins-in-cplusplus**](https://www.tutorialspoint.com/bin-packing-problem-minimize-number-of-used-bins-in-cplusplus)

**Vertexcover**

**Travelling Salesman Problem**