2.Knapsack problem using brute force

```
#include <stdio.h>
#include <math.h>

#define MAX_ITEMS 20

struct Item {
   int weight;
   int value;
};

void knapsack_bruteforce(struct Item items[], int n, int capacity) {
   int max_value = 0;

   for (int i = 0; i < (1 << n); i++) {
      int total_weight = 0, total_value = 0;

      for (int j = 0; j < n; j++) {
         if (i & (1 << j)) {
            total_weight += items[j].weight;
            total_value += items[j].value;
         }
      }

      if (total_weight <= capacity && total_value > max_value) {
         max_value = total_value;
      }
   }

   printf("Maximum value in Knapsack = %d\n", max_value);
}
```

```
int main() {
    int n, capacity;

    printf("Enter the number of items: ");
    scanf("%d", &n);

    struct Item items[n];

    printf("Enter the weights and values of the items:\n");
    for (int i = 0; i < n; i++) {
        printf("Item %d - Weight: ", i + 1);
        scanf("%d", &items[i].weight);
        printf("Item %d - Value: ", i + 1);
        scanf("%d", &items[i].value);
    }

    printf("Enter the capacity of the knapsack: ");
    scanf("%d", &capacity);

    knapsack_bruteforce(items, n, capacity);

    return 0;
}
```

Output

```
Enter the number of items: 2
Enter the weights and values of the items:
Item 1 - Weight: 33
Item 1 - Value: 666
Item 2 - Weight: 3
Item 2 - Value: 55
Enter the capacity of the knapsack: 80
Maximum value in Knapsack = 721


=== Code Execution Successful ===
```