

```

1 // Approccio top-down
2
3 const int MAXN = 5000, MAXK = 5000, INF = 1000000000;
4
5 // La matrice memo contiene le soluzioni dei sottoproblemi
6 int memo[MAXN + 1][MAXK + 1];
7
8 /* Questa funzione calcola ricorsivamente le soluzioni
9  * che verranno memoizzate in memo
10  */
11 int mangia_rec(int i, int j, int P[]) {
12     // Restituisce 0 se il limite j è negativo
13     if (j < 0) return 0;
14
15     /* Se la soluzione del sottoproblema è già stata calcolata,
16      * la restituisce
17      */
18     if (memo[i][j] != -1)
19         return memo[i][j];
20
21     // Calcola ricorsivamente la soluzione e la memoizza in memo[i][j]
22     return memo[i][j] = min(mangia_rec(i - 1, j, P),
23                             mangia_rec(i - 1, j - P[i - 1], P) + P[i - 1]);
24 }
25
26 int mangia(int N, int K, int P[]) {
27     // Casi base
28     for (int i = 1; i <= K; i++)
29         memo[0][i] = INF;
30
31     /* Tutta la matrice, salvo la prima riga,
32      * viene riempita con -1
33      */
34     for (int i = 1; i <= N; i++)
35         for (int j = 0; j <= K; j++)
36             memo[i][j] = -1;
37
38     /* Chiama la funzione mangia_rec() sui parametri
39      * iniziali e restituisce la soluzione
40      */
41     return mangia_rec(N, K, P);
42 }

```