

ESPOSITO CLAUDIA ANTONELLA

CODICE

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

int programmazioneDinamica(const vector<int>& monete) {
    int M = monete.size();
    int totale = 0;

    for (int i = 0; i < M; i++) {
        totale += monete[i];
    }

    int N = totale / 2;
    vector<vector<int>> dp(M + 1, vector<int>(N + 1, 0));
    for (int i = 1; i <= M; i++) {
        for (int j = 1; j <= N; j++) {

            if (monete[i - 1] <= j) {
                dp[i][j] = std::max(dp[i - 1][j], monete[i - 1] + dp[i - 1][j -
monete[i - 1]]);
            } else {
                dp[i][j] = dp[i - 1][j];
            }
        }
    }
    return totale - 2 * dp[M][N];
}

int main() {
    int N;
    cin >> N;

    while (N-- > 0) {
        int M;
        cin >> M;
        vector<int> monete(M);
        for (int i = 0; i < M; i++) {
            cin >> monete[i];
        }

        int differenza = programmazioneDinamica(monete);
        cout<<endl;
        cout << differenza << std::endl;
    }

    return 0;
}
```

SPIEGAZIONE CODICE

Il programma usa un approccio bottom up dove **dp** è una tabella usata per conservare i risultati parziali: ogni cella è aggiornata con i valori dei sottoproblemi più piccoli.

Simile al problema dello zaino: entrambi devono ottimizzare un insieme di elementi, ciascuno con un valore associato, in modo da massimizzare o minimizzare una funzione obiettivo, soggetta a determinati vincoli.

- **N** è il numero di casi di test
- **M** è il numero di monete
- **Monete[]** è un vettore contenente i valori delle monete
- La funzione **programmazione dinamica** prende in ingresso il vettore **monete**
- Calcoliamo prima il totale di tutte le monete che abbiamo ottenendo la variabile **totale** e la variabile locale alla funzione **N** che è il valore totale diviso 2.
- Iniziamo una matrice **dp** che ha dimensione **M+1** e **mid+1**, **dp[i][j]** è la somma massima di valore ottenibile utilizzando le prime monete **i** e avendo a disposizione uno spazio massimo **j**
- I due cicli for servono a percorrere ogni cella della matrice **dp**
- **dp[0][0]** è inizializzata a 0
- Se il valore corrente della moneta è minore di **j** allora **dp** sarà il valore massimo tra $\max(dp[i-1][j], \text{monete}[i-1] + dp[i-1][j - \text{monete}[i-1]])$ ovvero la moneta corrente può essere inclusa o no, se invece è maggiore di **j** aggiorniamo **dp** al valore della riga precedente
- Alla fine **dp[M][N]** ha la massima somma usando le **M** monete con **N** come spazio disponibile perciò la differenza minima è il totale meno il doppio di questo valore ottenuto

Esempio

Monete[] = 1 4 7 4 8

M è 5 perciò la prima dimensione è 6

N sarà 12

dp[0][0] inserisco 0

per **i=1 j=1**

dp[1][1]

se $\text{monete}[0] \leq 1 \rightarrow \text{monete}[0] = 1$

primo caso : inserisco in **dp[i][j]** il massimo tra $\max(dp[0][1], \text{monete}[0] + dp[0][1-1]) = (0, 1+0) = 1$
dp[1][1] = 1

per **i = 1, j = 2**

dp[1][2]

se $\text{monete}[0] \leq 2 \rightarrow \text{monete}[0] = 1$

primo caso inserisco il massimo tra $\max(dp[0][2], \text{monete}[0] + dp[0][2-1]) = (0, 1+0) = 1$

dp[1][2] = 1

.....

Per **i=2 j=1**

dp[2][1]

se $\text{monete}[1] \leq 1 \rightarrow \text{monete}[1] = 4$

secondo caso $dp[i][j] = dp[i-1][j]$;

dp[2][1] = dp[1][1]

per **i=2 j=4**

se $\text{monete}[1] \leq 4 \rightarrow \text{monete}[1] = 4$

primo caso $\max(1, 4 + dp[1][4-4]) = (1, 4) = 4$

dp[2][4] = 4

per **i=2 j = 5**

se $\text{monete}[1] \leq 5 \rightarrow \text{monete}[1] = 4$

primo caso $\max(1, 4 + dp[1][5-4]) = (1, 5) = 5$

dp[2][5] = 5

...

Tabella dp

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	4	5	5	5	5	5	5	5	5
3	0	1	1	1	4	5	5	7	8	8	8	11	12
4	0	1	1	1	4	5	5	7	8	9	9	11	12
5	0	1	1	1	4	5	5	7	8	9	9	11	12

La cella[M][mid] = 12

La **differenza** è = $24 - 12 * 2 = 0$

La funzione programmazione dinamica ha complessità temporale $O(M * N)$ ovvero il valore dei due cicli for

Input

6
5
1 4 7 4 8
4
5 4 9 1
1
50
3
5 6 7
2
10 10
4
2 5 8 3

OUTPUT

0

1

50

4

0

2