

Nama : Muhammad Naufal Ramadhani

Nim : 1203230036

Kelas : IF 03-03

1. Source Code

```
1. #include <bits/stdc++.h>
2.
3. using namespace std;
4.
5. string ltrim(const string &);
6. string rtrim(const string &);
7. vector<string> split(const string &);
8.
9. /*
10.  * Complete the 'twoStacks' function below.
11.  *
12.  * The function is expected to return an INTEGER.
13.  * The function accepts following parameters:
14.  * 1. INTEGER maxSum
15.  * 2. INTEGER_ARRAY a
16.  * 3. INTEGER_ARRAY b
17.  */
18.
19. int twoStacks(int maxSum, vector<int> a, vector<int> b) {
20.     int n = a.size();
21.     int m = b.size();
22.
23.     // Hitung berapa banyak elemen yang bisa diambil dari
    stack a
24.     int takenFromA = 0, sum = 0;
25.     while (takenFromA < n && sum + a[takenFromA] <= maxSu
    m) {
26.         sum += a[takenFromA];
27.         takenFromA++;
28.     }
29.
30.     int maxScore = takenFromA;
31.
32.     // Coba tambahkan elemen dari stack b satu per satu,
    dan kurangi elemen dari stack a jika diperlukan
```

```

33.     int takenFromB = 0;
34.     while (takenFromB < m && takenFromA >= 0) {
35.         sum += b[takenFromB];
36.         takenFromB++;
37.
38.         // Kurangi elemen dari stack a jika jumlah sudah
        melebihi maksimum
39.         while (sum > maxSum && takenFromA > 0) {
40.             takenFromA--;
41.             sum -= a[takenFromA];
42.         }
43.
44.         // Periksa apakah skor saat ini lebih besar dari
        skor maksimum yang ditemukan sejauh ini
45.         if (sum <= maxSum) {
46.             maxScore = max(maxScore, takenFromA + takenFr
omB);
47.         }
48.     }
49.
50.     return maxScore;
51. }
52.
53. int main()
54. {
55.     // ofstream fout(getenv("OUTPUT_PATH"));
56.     // Buang output file karena output akan ditampilkan d
i terminal
57.     // string g_temp;
58.     // getline(cin, g_temp);
59.     // int g = stoi(ltrim(rtrim(g_temp)));
60.
61.     // for (int g_itr = 0; g_itr < g; g_itr++) {
62.     //     string first_multiple_input_temp;
63.     //     getline(cin, first_multiple_input_temp);
64.     //     vector<string> first_multiple_input = split(rtr
im(first_multiple_input_temp));
65.     //     int n = stoi(first_multiple_input[0]);
66.     //     int m = stoi(first_multiple_input[1]);
67.     //     int maxSum = stoi(first_multiple_input[2]);
68.     //     string a_temp_temp;
69.     //     getline(cin, a_temp_temp);

```

```

70.         //      vector<string> a_temp = split(rtrim(a_temp_tem
           p));
71.         //      vector<int> a(n);
72.         //      for (int i = 0; i < n; i++) {
73.         //          int a_item = stoi(a_temp[i]);
74.         //          a[i] = a_item;
75.         //      }
76.         //      string b_temp_temp;
77.         //      getline(cin, b_temp_temp);
78.         //      vector<string> b_temp = split(rtrim(b_temp_tem
           p));
79.         //      vector<int> b(m);
80.         //      for (int i = 0; i < m; i++) {
81.         //          int b_item = stoi(b_temp[i]);
82.         //          b[i] = b_item;
83.         //      }
84.         //      int result = twoStacks(maxSum, a, b);
85.         //      fout << result << "\n";
86.         //  }
87.
88.         // Buat input secara langsung untuk testing
89.         vector<int> a = {4, 5, 2, 1, 1};
90.         vector<int> b = {3, 1, 1, 2};
91.         int maxSum = 11;
92.         int result = twoStacks(maxSum, a, b);
93.         cout << result << endl;    // Tampilkan output langsung
           di terminal
94.
95.         // fout.close();
96.
97.         return 0;
98.     }
99.
100. string ltrim(const string &str) {
101.     string s(str);
102.
103.     s.erase(
104.         s.begin(),
105.         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>
           >(isspace)))
106.     );
107.
108.     return s;

```

```

109. }
110.
111. string rtrim(const string &str) {
112.     string s(str);
113.
114.     s.erase(
115.         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, i
116.             nt>(isspace))).base(),
117.         s.end()
118.     );
119.     return s;
120. }
121.
122. vector<string> split(const string &str) {
123.     vector<string> tokens;
124.
125.     string::size_type start = 0;
126.     string::size_type end = 0;
127.
128.     while ((end = str.find(" ", start)) != string::npos)
129.     {
130.         tokens.push_back(str.substr(start, end - start));
131.         start = end + 1;
132.     }
133.
134.     tokens.push_back(str.substr(start));
135.
136.     return tokens;
137. }
138.
139.

```

2. Penjelasan

1. `#include <bits/stdc++.h>`

Baris ini mengimpor semua pustaka standar yang diperlukan dalam pemrograman C++. Ini akan mengimpor semua pustaka dasar yang sering digunakan, sehingga kita tidak perlu menyebutkan pustaka-pustaka tersebut satu per satu.

2. `using namespace std;`

Baris ini mendeklarasikan penggunaan namespace `'std'`, yang merupakan namespace standar untuk seluruh pustaka standar C++.

3. `string ltrim(const string &);`
`string rtrim(const string &);`
`<string> split(const string &);`

Baris-baris ini mendeklarasikan prototipe dari tiga fungsi yang digunakan untuk membersihkan dan membagi string, yaitu `'ltrim'`, `'rtrim'`, dan `'split'`.

4. `int twoStacks(int maxSum, vector<int> a, vector<int> b) {`

Baris ini mendefinisikan fungsi `'twoStacks'`, yang menerima tiga parameter: `'maxSum'` (batas jumlah), `'a'` (vektor yang merepresentasikan tumpukan pertama), dan `'b'` (vektor yang merepresentasikan tumpukan kedua).

5. `int n = a.size();`
`int m = b.size();`

Baris ini menghitung ukuran tumpukan pertama (`'n'`) dan tumpukan kedua (`'m'`) dengan menggunakan fungsi `'size()'` pada vektor.

```

6.  int takenFromA = 0, sum = 0;
while (takenFromA < n && sum + a[takenFromA] <= maxSum) {
    sum += a[takenFromA];
    takenFromA++;
}

```

Baris-baris ini mencoba untuk mengambil sebanyak mungkin elemen dari tumpukan pertama (`a`) tanpa melebihi batas jumlah (`maxSum`). Ini dilakukan dengan menggunakan loop `while` untuk mengambil elemen dari tumpukan pertama secara berurutan hingga jumlah elemennya melebihi batas atau sampai seluruh elemen telah diambil.

```

7.  int maxScore = takenFromA;

```

Baris ini menginisialisasi `maxScore` dengan jumlah elemen yang telah berhasil diambil dari tumpukan pertama.

```

8.  int takenFromB = 0;
while (takenFromB < m && takenFromA >= 0) {
    sum += b[takenFromB];
    takenFromB++;
}

```

Baris-baris ini menambahkan elemen dari tumpukan kedua satu per satu, sambil memeriksa apakah jumlah total dari elemen dari kedua tumpukan masih berada di bawah atau sama dengan batas jumlah (`maxSum`). Jika melebihi, maka program akan mencoba untuk mengurangi elemen dari tumpukan pertama sebelum menambahkan elemen baru dari tumpukan kedua.

```

9.  while (sum > maxSum && takenFromA > 0) {
    takenFromA--;
    sum -= a[takenFromA];
}

```

Baris-baris ini memeriksa dan mengurangi elemen dari tumpukan pertama jika jumlahnya sudah melebihi batas jumlah.

```

10. if (sum <= maxSum) {
    maxScore = max(maxScore, takenFromA + takenFromB);
}
}

```

Baris-baris ini memeriksa apakah skor saat ini lebih besar dari skor maksimum yang ditemukan sejauh ini. Jika ya, maka `maxScore` akan diperbarui dengan nilai yang lebih besar.

```

11. return maxScore;
}

```

Baris ini mengembalikan jumlah maksimum elemen yang berhasil diambil dari kedua tumpukan.

```

12. int main() {

```

Baris ini mendefinisikan fungsi `main`, yang merupakan titik masuk utama untuk eksekusi program.

```

// Buat input secara langsung untuk testing

```

```

13. vector<int> a = {4, 5, 2, 1, 1};
    vector<int> b = {3, 1, 1, 2};
    int maxSum = 11;

```

Baris-baris ini memberikan input secara langsung untuk tumpukan pertama (`a`), tumpukan kedua (`b`), dan batas jumlah (`maxSum`) untuk tujuan pengujian.

```

14. int result = twoStacks(maxSum, a, b);
    cout << result << endl; // Tampilkan output langsung di terminal

```

Baris ini memanggil fungsi `twoStacks` dengan input tersebut dan menampilkan hasilnya di terminal.

```

15. return 0;
}

```

Baris ini mengakhiri fungsi `main` dan menandakan akhir dari program.

3. SS Input & Output

Input (stdin)

[Download](#)

```
1 1
2 5 4 11
3 4 5 2 1 1
4 3 1 1 2
```

Your Output (stdout)

```
1 5
```