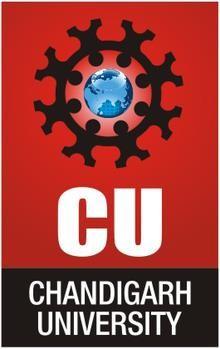
**CHANDIGARH UNIVERSITY**

UNIVERSITY INSTITUTE OF ENGINEERING

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



|  |  |
| --- | --- |
| **Submitted By:                                                                          Submitted To:**  Yash Gupta ER. Monika(E12802) | |
| **Subject Name** | Design Analysis and Algorithm |
| **Subject Code** | 20CSP\_312 |
| **Branch** | CSE |
| **Semester** | 5th |

**LAB -INDEX**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Program** | **Date** | **Evaluation** | | | | **Sign** |
| **LW(12)** | **VV(8)** | **FW(10)** | **Total (30)** |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**Experiment 6**

**1. Aim/Overview of the practical:**

To implement a subset-sum problem using dynamic programming.

**2. Algorithm:**

* 1. We create a boolean subset[][] and fill it in bottom up manner.
  2. The value of subset[i][j] will be true if there is a subset of set[0..j-1] with sum equal to i., otherwise false.
  3. subset[i][j] = true if there is a subset with:
  4. the i-th element as the last element \* sum equal to j
  5. subset[i][0] = true as sum of {} = 0
  6. subset[0][j] = false as with no elements we can get no sum
  7. subset[i][j] = subset[i-1][j-E1]; where E1 = array[i-1]
  8. Finally, we return subset[n][sum].

**3. Steps for experiment/practical/Code:**

#include <iostream>

using namespace std;

bool isSubsetSum(int set[], int n, int sum)

{

bool subset[n + 1][sum + 1];

for (int i = 0; i <= n; i++)

subset[i][0] = true;

for (int i = 1; i <= sum; i++)

subset[0][i] = false;

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= sum; j++) {

if (j < set[i - 1])

subset[i][j] = subset[i - 1][j];

if (j >= set[i - 1])

subset[i][j] = subset[i - 1][j]

|| subset[i - 1][j - set[i - 1]];

}

}

return subset[n][sum];

}

int main()

{

int set[] = { 3, 34, 4, 12, 5, 2 };

int sum = 9;

int n = sizeof(set) / sizeof(set[0]);

if (isSubsetSum(set, n, sum) == true)

cout <<"Found a subset with given sum";

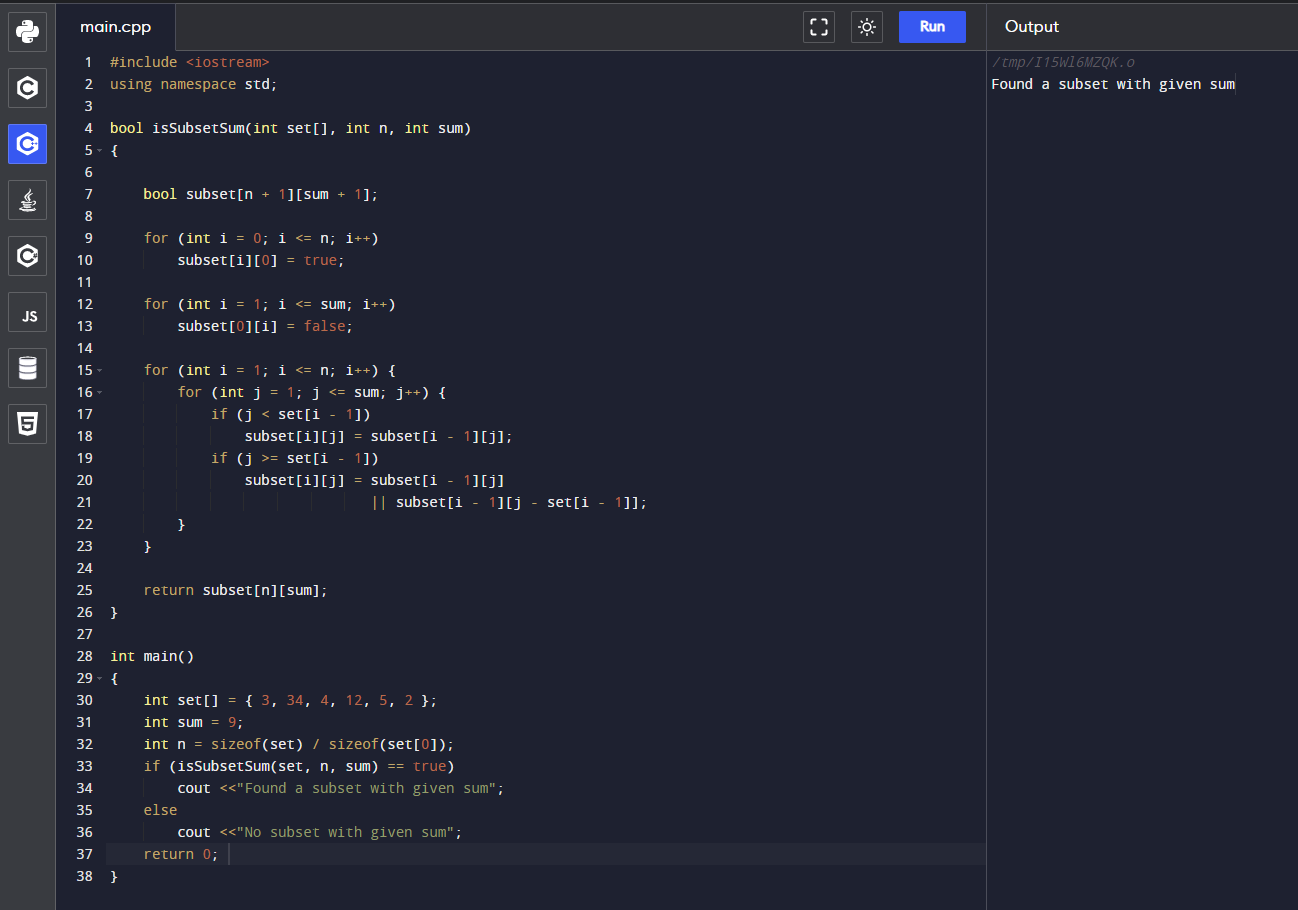
else

cout <<"No subset with given sum";

return 0;

}}

**4. Result/Output/Writing Summary:**



**5. Observations/Discussions/ Complexity Analysis:**

Time complexity – O ( sum\*n ).

6. **Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |