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

// Constant definitions
const int MAX = 100;

// class definitions
class SubSet
{
    int stk[MAX], set[MAX];
    int size, top, count;
    public:
    SubSet()
    {
        top = -1;
        count = 0;
    }
    void getInfo(void);
    void push(int data);
    void pop(void);
    void display(void);
    int fnFindSubset(int pos, int sum);
};

/
**********************************************************************
**********
*Function   : getInfo
*Description: Function to read input
*Input parameters: no parameters
*RETURNS    : no value
**********************************************************************
**********/

void SubSet :: getInfo(void)
{
    int i;
    cout << "Enter the maximum number of elements : ";
    cin >> size;

    cout << "Enter the weights of the elements : \n";
    for (i=1; i<=size; i++)
        cin >> set[i];

}

/
**********************************************************************
**********
*Function   : push
*Description: Function to push an element on to the stack
*Input parameters:
*int data   - value to be pushed on to the stack
*RETURNS    : no value
```

```
**********************************************************************
**********/
void SubSet :: push(int data)
{
    stk[++top] = data;
}

/
**********************************************************************
**********
*Function   : pop
*Description: Function to pop an element from the stack
*Input parameters: no parameters
*RETURNS    : no value
**********************************************************************
**********/

void SubSet :: pop(void)
{
    top--;
}

/
**********************************************************************
**********
*Function   : display
*Description: Function to display solution to sub set sum problem
*Input parameters: no parameters
*RETURNS    : no value
**********************************************************************
**********/
void SubSet :: display()
{
    int i;
    cout << "\nSOLUTION #"<< ++count <<" IS\n{ ";
    for (i=0; i<=top; i++)
        cout << stk[i] << " ";

    cout << "}" << endl;
}

/
**********************************************************************
**********
*Function   : fnFindSubset
*Description    : Function to solve Subset sum problem.
*Input parameters:
*   int pos - position
*   int sum - sum of elements
*RETURNS    : returns 1 if solution exists or zero otherwise
**********************************************************************
**********/
int SubSet :: fnFindSubset(int pos, int sum)
{
```

```
    int i;
    static int foundSoln = 0;

    if (sum>0)
    {
        for (i=pos; i<=size; i++)
        {
            push(set[i]);
            fnFindSubset(i+1, sum - set[i]);
            pop();
        }
    }

    if (sum == 0)
    {
        display();
        foundSoln = 1;
    }

    return foundSoln;
}

/
**********************************************************************
**********
*Function   : main
*Input parameters: no parameters
*RETURNS    :   0 on success
**********************************************************************
**********/
int main(void)
{
    int i,sum;

    SubSet set1;

    set1.getInfo();
    cout << "Enter the total required weight : ";
    cin >> sum;

    cout << endl;

    if (!set1.fnFindSubset(1, sum))
        cout << "\n\nThe given problem instance doesnt have any
solution." << endl;
    else
        cout << "\n\nThe above-mentioned sets are the required
solution to the given instance." << endl;

    return 0;
}
```

OUTPUT

SAMPLE 1

Enter the maximum number of elements : 5

Enter the weights of the elements :

1 2 3 4 5

Enter the total required weight : 5

SOLUTION #1 IS

{ 1 4 }

SOLUTION #2 IS

{ 2 3 }

SOLUTION #3 IS

{ 5 }

The above-mentioned sets are the required solution to the given instance.

SAMPLE 2

Enter the maximum number of elements : 4

Enter the weights of the elements :

1 2 3 4

Enter the total required weight : 11

The given problem instance doesnt have any solution.