

Find a subset of a given set $S = \{s_1, s_2, \dots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d . For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.

```
#include <stdio.h>
```

```
#define TRUE 1
```

```
#define FALSE 0
```

```
#define MAX_SIZE 50
```

```
int include[MAX_SIZE], w[MAX_SIZE], sum, n;
```

```
int promising(int i, int wt, int total) {
```

```
    return (((wt + total) >= sum) && ((wt == sum) || (wt + w[i + 1] <= sum)));
```

```
}
```

```
void sumset(int i, int wt, int total);
```

```
void input() {
```

```
    int i, temp, total = 0;
```

```
    printf("\nEnter how many numbers: ");
```

```
    scanf("%d", &n);
```

```
    printf("\nEnter %d numbers: ", n);
```

```
    for (i = 0; i < n; i++) {
```

```
        scanf("%d", &w[i]);
```

```
        total += w[i];
```

```
}
```

```
printf("\nEnter the sum value to create a subset: ");
```

```
scanf("%d", &sum);
```

```
if (total < sum) {
```

```
    printf("\nSubset construction is not possible\n");
```

```
} else {
```

```
    for (i = 0; i < n; i++) {
```

```
        include[i] = 0;
```

```
    }
```

```
printf("\nThe given %d numbers in ascending order: ", n);
```

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

```
    printf("%3d", w[i]);
```

```
printf("\n\nThe solution using backtracking is:\n");
```

```
sumset(-1, 0, total);
```

```
}
```

```
}
```

```
void sumset(int i, int wt, int total) {
```

```
    int j;
```

```
    if (promising(i, wt, total)) {
```

```
        if (wt == sum) {
```

```
            printf("{}");
```

```
            for (j = 0; j <= i; j++)
```

```
                if (include[j])
```

```
                    printf("%3d", w[j]);
```

```
        printf(" }\n");
    } else {
        include[i + 1] = TRUE;
        sumset(i + 1, wt + w[i + 1], total - w[i + 1]);
        include[i + 1] = FALSE;
        sumset(i + 1, wt, total - w[i + 1]);
    }
}
}
```

```
int main() {
    input();
    return 0;
}
```