Two Sum

Program :

```
#include <stdio.h>

#include <stdlib.h>


// Structure to store the result indices

struct Result {

   int first;

   int second;

};

// Function to find the two indices

struct Result twoSum(int nums[], int size, int target) {

   struct Result result = {-1, -1};

   // Create a hash map to store elements and their indices

   int* map = (int*)malloc((target + 1) * sizeof(int));

   for (int i = 0; i <= target; i++) {

      map[i] = -1; // Initialize all indices to -1

   }

   for (int i = 0; i < size; i++) {

      int complement = target - nums[i];

      if (complement >= 0 && map[complement] != -1) {

         result.first = map[complement];

         result.second = i;

         break;

      }

      map[nums[i]] = i;

   }

   free(map); // Clean up memory

   return result;

}
```

```
int main() {
    int nums[] = {2, 7, 11, 15};
    int target = 9;
    int size = sizeof(nums) / sizeof(nums[0]);
    struct Result result = twoSum(nums, size, target);
    if (result.first != -1 && result.second != -1) {
        printf("Indices: %d, %d\n", result.first, result.second);
    } else {
        printf("No valid solution found.\n");
    }
    return 0;
}
```