C-Codes

Aditi Dure

May 13, 2024

1 Two-Sum-Advanced

1.1 Question

Given a target number and an array, along with the number of elements, output the indices of numbers that add up to the given target.

1.2 Solution

1. **Methods:**

   This question can be approached by iterating through the array and finding all possible pairs that sum up to the target.

2. **Approach:**

   (a) **Main Function:**

   - Variables:
     - n: number of elements in the array
     - $array$[]: array that stores the elements
     - $target$: the target number that the elements should add up to
   - Method:
     - Prompt the user to enter the number of elements and the elements themselves.
     - Validate that all entered numbers are natural numbers.
     - Ask for the target number.
     - Iterate through the array to find pairs that sum up to the target.

   (b) **generateCombinations Function:**

   - Variables:
     - arr[]: array containing the elements
     - $data$[]: array to store the combination of indices

    - *start*, *end*: indices to iterate through the array
    - *index*: index at which the new index has to be stored
    - *r*: size of combinations to generate
    - *tar*: target number
  - Method:
    - Initialize an array with the given size.
    - Iterate through the array to generate combinations using recursion.

(c) **printCombinations Function:**

  - Method:
    - Recursively generate combinations and print those which sum up to the target.

3. **Sample C Code:**

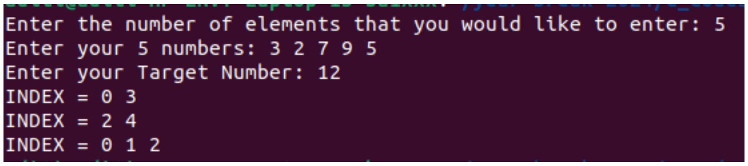
```c
#include<stdio.h>
void printCombination(int arr[], int data[], int start, int end, int index, int r, int tar)
{
    if (index == r)
    {
      int sum = 0;
        for (int i = 0; i < r; ++i)
          sum = sum + arr[data[i]];
        if(sum == tar)
        {
          printf("INDEX = ");
      for(int j = 0; j < r; j++)
        printf("%d ", data[j]);
      printf("\n");
        }
        return;
    }
    for (int i = start; i < end && end - i >= r - index; ++i)
    {
        data[index] = i;
        printCombination(arr, data, i+1, end, index+1, r, tar);
    }
}

void generateCombinations(int arr[], int n, int r, int tar)
{
    int data[r];
    printCombination(arr, data, 0, n, 0, r, tar);
}



int main()
{
    int n, target, status;
  printf("Enter the number of elements that you would like to enter: ");
    scanf("%d", &n);
    int array[n];
    printf("Enter your %d numbers: ", n);
    do
    {
      status = 1;
      for (int i = 0; i < n; i++)
      scanf("%d", &array[i]);
      for (int i = 0; i < n; i++)
      {
        if(array[i] <= 0)
        {
          printf("Enter another set of values, because they should be natural numbers: ");
          status = 0;
          break;
        }
      }
    }
    while (!status);
    printf("Enter your Target Number: ");
    scanf("%d",&target);
    for(int i = 1; i < n; i++)
    generateCombinations(array, n, i, target);
    return 0;
}
```

4. **Output:**



```
Enter the number of elements that you would like to enter: 5
Enter your 5 numbers: 3 2 7 9 5
Enter your Target Number: 12
INDEX = 0 3
INDEX = 2 4
INDEX = 0 1 2
```