**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **17-06-2020** | | | | | **Name:** | **Huda Sultana** | |
| **Sem & Sec** | **8 A** | | | | | **USN:** | **4AL16CS039** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **-** | | | | | | |
| **Max. Marks** | | **-** | | **Score** | | | **-** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Understanding CIDR Notation** | | | | | | | |
| **Certificate Provider** | | | **AWS** | | **Duration** | | | **15 mins** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**   1. **Find the smallest positive integer value that cannot be represented as sum of any subset of a given array sorted in ascending order.**. | | | | | | | | |
| **Status: Solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **Yes** | | | |
| **If yes Repository name** | | | | | **Hudasulltana/online\_coding** | | | |
| **Uploaded the report in slack** | | | | | **Yes** | | | |

Online Test Details:

Not conducted

Certification Course Details:



Coding Challenges Details:

**PROGRAM 1 .**

**//Find the smallest positive integer value that cannot be represented as sum of any subset of a given array sorted in ascending order.**

**#include<stdio.h>**

**int findSmallest(int arr[], int n)**

**{**

**int res = 1;**

**for (int i = 0; i < n && arr[i] <= res; i++)**

**res = res + arr[i];**

**return res;**

**}**

**int main()**

**{**

**int arr1[] = {1, 1, 6, 10, 11, 15};**

**int n1 = sizeof(arr1)/sizeof(arr1[0]);**

**printf("%d\n", findSmallest(arr1, n1));**

**int arr2[] = {1, 1, 1, 1};**

**int n2 = sizeof(arr2)/sizeof(arr2[0]);**

**printf("%d\n", findSmallest(arr2, n2));**

**int arr3[] = {1, 1, 3, 4};**

**int n3 = sizeof(arr3)/sizeof(arr3[0]);**

**printf("%d\n",findSmallest(arr3, n3));**

**int arr4[] = {1 , 2, 5, 10, 20, 40};**

**int n4 = sizeof(arr4)/sizeof(arr4[0]);**

**printf("%d\n",findSmallest(arr4, n4));**

**int arr5[] = {1, 2, 3, 4, 5 , 6};**

**int n5 = sizeof(arr5)/sizeof(arr5[0]);**

**printf("%d\n",findSmallest(arr5, n5));**

**return 0;**

**}**