**Q.1 Problem Description :**

The function accepts two positive integers ‘r’ and ‘unit’ and a positive integer array ‘arr’ of size ‘n’ as its argument ‘r’ represents the number of rats present in an area, ‘unit’ is the amount of food each rat consumes and each ith element of array ‘arr’ represents the amount of food present in ‘i+1’ house number, where 0 <= i

**Note:**

* Return -1 if the array is null
* Return 0 if the total amount of food from all houses is not sufficient for all the rats.
* Computed values lie within the integer range.

**Example:**

**Input:**

* r: 7
* unit: 2
* n: 8
* arr: 2 8 3 5 7 4 1 2

**Output:**

4

**Q.2 You are given a function.**  
int CheckPassword(char str[], int n);  
The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0.  
str is a valid password if it satisfies the below conditions.

* – At least 4 characters
* – At least one numeric digit
* – At Least one Capital Letter
* – Must not have space or slash (/)
* – Starting character must not be a number

**Assumption:**  
Input string will not be empty.

**Example:**

**Input:**  
aA1\_67  
**Output:**  
1

**Sample Input:**  
a987 abC012  
**Output:**  
0

**Q.3 You are given a function,**int findCount(int arr[], int length, int num, int diff);

The function accepts an integer array ‘arr’, its length and two integer variables ‘num’ and ‘diff’. Implement this function to find and return the number of elements of ‘arr’ having an absolute difference of less than or equal to ‘diff’ with ‘num’.  
Note: In case there is no element in ‘arr’ whose absolute difference with ‘num’ is less than or equal to ‘diff’, return -1.

**Example:**  
**Input:**

* arr: 12 3 14 56 77 13
* num: 13
* diff: 2

**Output:**  
3

**Q.4 Implement the following Function**

def differenceofSum(n. m)

The function accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between sum of integers not divisible by n with sum of numbers divisible by n.

**Assumption:**

* n>0 and m>0
* Sum lies between integral range

**Example**

**Input**n:4  
m:20  
**Output**90

Q.5 **You are required to implement the following Function**

def LargeSmallSum(arr)

The function accepts an integers arr of size ’length’ as its arguments you are required to return the sum of second largest  element from the even positions and second smallest from the odd position of given ‘arr’

**Assumption:**

* All array elements are unique
* Treat the 0th position as even

**NOTE**

* Return 0 if array is empty
* Return 0, if array length is 3 or less than 3

**Example**

**Input**

arr:3 2 1 7 5 4

**Output**

7

**Q.6 Implement the following Function**

def ProductSmallestPair(sum, arr)

The function accepts an integers sum and an integer array arr of size n. Implement the function to find the pair, (arr[j], arr[k]) where j!=k, Such that arr[j] and arr[k] are the least two elements of array (arr[j] + arr[k] <= sum) and return the product of element of this pair

**NOTE**

* Return -1 if array is empty or if n<2
* Return 0, if no such pairs found
* All computed values lie within integer range

**Example**

**Input**

sum:9

Arr:5 2 4 3 9 7 1

**Output**

2

**Q.7**

N-base notation is a system for writing numbers that uses only n different symbols, This symbols are the first n symbols from the given notation list(Including the symbol for o) Decimal to n base notation are (0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 10:A,11:B and so on upto 35:Z)

Implement the following function

Char\* DectoNBase(int n, int num):

The function accept positive integer n and num Implement the function to calculate the n-base equivalent of num and return the same as a string

Steps:

1. Divide the decimal number by n,Treat the division as the integer division
2. Write the the remainder (in  n-base notation)
3. Divide the quotient again by n, Treat the division as integer division
4. Repeat step 2 and 3 until the quotient is 0
5. The n-base value is the sequence of the remainders from last to first

**Assumption:**

1 < n < = 36

**Example**

**Input**

n: 12

num: 718

**Output**

4BA

**Q.8 Problem Statement**

A carry is a digit that is transferred to left if sum of digits exceeds 9 while adding two numbers from right-to-left one digit at a time

You are required to implement the following function.

Int NumberOfCarries(int num1 , int num2);

The functions accepts two numbers ‘num1’ and ‘num2’ as its arguments. You are required to calculate and return  the total number of carries generated while adding digits of two numbers ‘num1’ and ‘ num2’.

**Assumption:** num1, num2>=0

**Example:**

* **Input**
  + Num 1: 451
  + Num 2: 349
* **Output**
  + 2