**Problem Statements**

1. Write a program to take a number from user and print if number is zero, a positive number or a negative number.
2. Write a program to take a number from user and print if number is an Armstrong number or not. A number will be considered an Armstrong number if it satisfies following conditions:
   1. Number is a 3 digits number
   2. Sum of the cubes of all its digits is equal to the number itself

**Examples:**  
371 is an Armstrong number since (3\*3\*3 =27) + (7\*7\*7=343) + (1\*1\*1=1) = (27+343+1) = 371  
156 is an Armstrong number since (1\*1\*1=1) + (5\*5\*5=125) + (3\*3\*3=27) = (1+125+27) = 153

1. Write a program to take students marks for following subject:  
   English, Mathematics, Computer Science, Social Studies, General Science  
     
   Calculate student’s grade & display on screen, as per the following rules:
   1. Mathematics and Computer Science are mandatory subjects
   2. Passing marks for Mathematics & Computer Science are 75% and 80% respectively
   3. Student should be passed in all mandatory and at least in two other subjects in order to secure a grade, otherwise, the student will be considered fail & assigned a grade “F” straightaway
   4. Total marks for each subject are 100. Display in error & stop programs if user inputs invalid marks for any of the subject. Make sure to display message explaining the invalid input

**Grade’s Rules**

|  |  |
| --- | --- |
| **Average Mark Range** | **Grade** |
| 80 – 100 | A |
| 61 – 80 | B |
| 40 – 60 | C |
| 0 – 39 | F |

1. Write a program input the angles of a triangle and check whether the triangle is valid or not. A triangle will be considered valid if none of its angle is less than or equal to zero and the sum of all angles is 180.   
     
   Please also print a message if triangle is a right-angled triangle. A triangle is considered right-angled triangle if it has one angle equal to 90 degrees
2. Write a program to input day number and print weekday. Also, print an error message if user input wrong weekday i.e., below 1 or above 7
3. Write a program to take a number as input and check if it’s even or odd then update even number into its upper 2nd nearest odd number and similarly convert the odd number into its lower even number using if-else statement.  
     
   **Example:** 10 is even and its nearest upper odd number is 11 and 2nd nearest upper odd number is 13  
   **Example:** 29 is odd and its nearest lower number is 28