5.1 Flow Controls

6.1 Arrays

7.1 Strings

Write following C++ programs as per given instructions.

1. There is a requirement to create a new **Student Evaluation System** for a school to verify students' marks by themselves. First, system will ask **username (**String**)** & **student ID (**number**)**. Then it asks you to enter marks for 3 subjects. Then system should print some messages according to entered vales. If student scores less than 50 for any subject, then he is repeated all 3 subjects (Means that he has got failed the exam). Therefore, print whether he should do repeat exam or not. Then, if average mark is higher than 75 then print "**Very Good**", 50 ~ 74 print "**Good**", 35~49 print "**average**" otherwise, print "**Fail**". Also, if he scored above 90 for all subjects, he would get a **first class**. System should handle errors, like if user enter invalid marks (-10, 120) for a given subject, system will show error message and ask to enter again. But if the 2nd time also user entered an invalid number, then system shows an error message & value will be set to zero for that subject automatically. Finally print the details in "nice" way.
    a. You may or may not use array support
    b. (Hint: You may create several test cases to verify the system.)

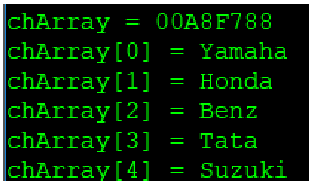
2. Write a simple program to perform below tasks. (Note: You should not use arrays)
    a. Request user to enter only positive numbers as inputs.
    b. Once he enters 0, program should stop requesting more values.
    c. If any negative value is entered, just ignore that value & display a message
    d. print maximum number & minimum number of entered values.
    e. Print the count, sum of entered numbers

3. Find the lucky number by taking user's birthday as an input. Use following format to input birthday YYYYMMDD -19850217. ➔ lucky number is 6.
(19850217 ➔ 1 + 9 + 8 +5+0+2+1+7= 33 → 3+3 = 6)

4. Write a C++ program to perform following task.
    a. Create a 2D **integer** array of 3 columns & 4 rows and Initially, set all values to 0
    b. User is able to enter integer values only and the maximum no. of elements is 12.
        i. But, whenever user enters (-1) before filling the entire 2D array then program should stop taking inputs.
        ii. inform user how many values already inserted in each turn
    c. Once data entry process is over, print all values of the array
    d. Find the maximum & minimum values and print them
        i. (-1) & (0) are not to be consider as minimum values
    e. Calculate the average of only the entered values

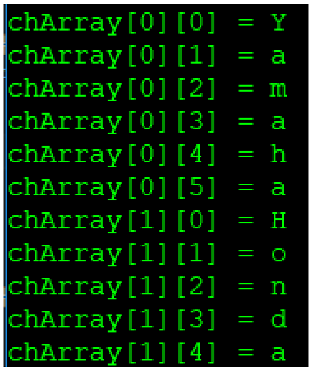
5. Write a program to count number of words in a user entedred string.
    a. (Hint: use C-Style String to store user input value.)

6. Create a 2D character array (C-Style String type) (5 rows & 10 columns)
    a. Store "Yamaha", "Honda", "Benz", "Tata", "Suzuki" strings
    b. Print base memory address without using address (&) operator (See image below)
    c. Print each row values using a for loop (See image below)



```
chArray = 00A8F788
chArray[0] = Yamaha
chArray[1] = Honda
chArray[2] = Benz
chArray[3] = Tata
chArray[4] = Suzuki
```

    d. Print each element using two for loops, & you must print **only** not null values (See image below)



```
chArray[0][0]  = Y
chArray[0][1]  = a
chArray[0][2]  = m
chArray[0][3]  = a
chArray[0][4]  = h
chArray[0][5]  = a
chArray[1][0]  = H
chArray[1][1]  = o
chArray[1][2]  = n
chArray[1][3]  = d
chArray[1][4]  = a
```

....
....
...

7. User will enter long text with **colon** separated. *example Book:Pen:Pencil:Table:Desk*
    a. Use String object to store user input value (string str)
    b. You must print the values one after another, after splitting the string using delimiter (:)
        Book
        Pen
        Pencil
        …
    c. All letters should be **UPPERCASE**
    d. Hint:
        i. You can use substring () & find () string functions support
        ii. You can use while loop support