Mid Lab Exam [DS]

1. Suppose your friend Robin is a first semester student of a university. In this semester he took 5 courses and got 94, 87,82,85,90 marks. Now which data structure is suitable to store these values? Store these values in appropriate data structure and calculate his CGPA considering all courses as 3 credits.

   For reference:

Marks	Point
>=90	4.00
>=85 and <90	3.75
>=80 and <85	3.50
>=75 and <80	3.25
>=70 and <75	3.00
>=50 and <70	2.75
<50	0.00

2. Suppose Robin's father is a class teacher. He has 20 students in class and they have 5 courses. Now he wants you to make a program, which will take number of each student for 5 courses and show their average marks. Now develop the program applying the appropriate data structure.

   Note: Build the program for 3 students at first.

3. Now an algorithm is given below. Implement this.

   a. Take a string **"s"** valued as **28+96-/**
   b. Take an **empty Stack "stack" of integers** (use your previous developed stack or built in Stack)
   c. **Traverse** the **string** form first to last. (**Either by for loop or while loop**)

```
for(i from 0 to length){
    char c = s[i]
}
```

   d. If the **character** is a **number** then **subtract 48** from the character and **push** it into the **stack**
      i. if (c >= '0' and c <= '9')
      ii. **push(c-48) [c is the variable holding the Character]**
   e. Else do as below
      i. Keep the top value in variable **"b"** and pop
      ii. Again keep the top value in variable **"a"** and pop
      iii. int x = eval(c,a,b)
      [c is for Operator Character; *eval* function definition is given below]
      iv. Push x into stack
   f. After traversing the string [End of loop]
      i. Keep the top value in variable **"r"**
   g. At last print the value of **r** obtained in step f.

Go to next page of eval function definition

Definition of eval function

```
int eval (char x,int op1,int op2)
  {
            if(x=='+')
                        return(op1+op2);
            if(x=='-')
                        return(op1-op2);
            if(x=='*')
                        return(op1*op2);
            if(x=='/')
                        return(op1/op2);
            if(x=='%')
                        return(op1%op2);
  }
```