Programming Test
Stratbeans Consulting

Details
Institute Name: _____ Date of Interview: _____
Your Name: _____ Branch: _____
B.Tech Score: _____ Class 12th Score & Board: _____
Class 10th Score & Board:_____

PLEASE SPEND 5 MINS TO READ THE INSTRUCTION

Instructions:
1. This is a test of programming capability and problem solving skills. This is a qualifying paper, next test would happen only if you are successful in this.
2. You can use **ANY** language of your choice (C,C++, Java, PHP, etc)
3. Just write the body of the function wherever you are asked to do so. **Please don't waste your time writing headers (#include etc) and C main functions, negative marking may apply in case of non adherence.**
4. Write code neatly so that it is easy for us to understand.
5.There are more marks to writing good code in terms of
    a. Clarity of logic
    b. Comments
    c. Boundary Conditions

Good Luck Do Well!

Programming Test
Stratbeans consulting

Problem #1 – Numerology (20 marks)

Some people believe in the power of numbers. **Power number** is a single digit number generated by doing repeated summing of the number till we get a single digit. E.g. for 132, it is 6 as 1 + 3 + 2= 6.

Stratbeans has a client Numerology Inc, which needs our help in creating a function **powerNumber**(int N), which would return the power number of the given number N. Working as part of Stratbeans team you have been assigned this task, could you please do it in the best possible manner, like commenting your code, efficient logic and taking care of boundary conditions etc !

Examples:
N=132, output = 6
N=1111111, output = 7
N=998, output = 8 // 9 + 9 + 8 = 26 ----> reduced further ---> 2 + 6 = 8

Programming Test
Stratbeans consulting

Problem #2 – Middle Length of a Triangle (25 marks)

Middle length of a triangle is the length of the side which is neither the smallest nor the largest. Say you have a triangle with side lengths 0.3, 0.4, and 0.5, middle length would be 0.4 in this case.

Lets say three co-ordinates of a triangle are (x1, y1), (x2, y2), (x3, y3). Lets write a method with following signature: **middleLength**(x1,y1,x2,y2,x3,y3). This function should return the side which is of the middle length.

Example

middleLength(0,0,3,0,0,4) should return 4
middleLength(0.8,1,0,1,0,1.6) should return 0.8

Programming Test
Stratbeans consulting

Problem #3 – Cheque Amount (30 marks)

Many a times when we write a cheque amount, we are also supposed to write the amount in words. Write a function to help me do that. Function **amountInWord** would take as argument integer N (where {0<= N <=1000}) and would print the amount in word.

Example

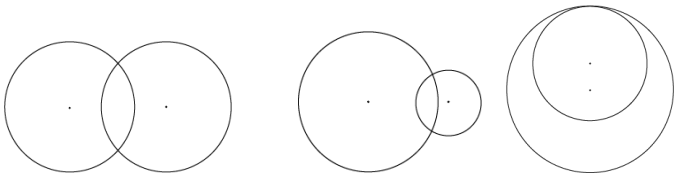
amountInWord(2) would print “Two”
amountInWord(12) would print “Twelve” (and not “One Two” !)
amountInWord(973) would print “Nine Hundred Seventy Three”

Programming Test
Stratbeans consulting

Problem #4 – Circle Collapse Video Game (25 marks)

Using Object Oriented Programming a programmer can create abstract representations. Stratbeans team is creating a video game where there are various circles of different radius on a 2D plane and whenever two circles overlap, it has to be detected. We want to you to define the properties of the class and write a method **isOverlapping**(), which would detect overlap between two circles and would return 1 in case of an overlap or else return 0 when its not. For example after you have implemented the class the following code would be possible:

Here are some visual example of what qualifies as overlap.



```
Circle c1, c2; // two circle objects
if (c1.isOverlapping(c2)) {
        print “Circles are overlapping”
}
else {
        print “Circles are not overlapping”
}

Class Circle
{
// Define properties


// implement method isOverlapping




}
```

Problem #5 – Gimme Largest Number (40 marks)

Write a function **largestNumber(arr)** that given a list of non negative integers, arranges them such that they form the largest possible number.

Example, given array [50, 2, 1, 9], the largest formed number is 95021.

Problem #6 – Where is my Ant (50 marks)

There is an ant at origin (0,0), that moves only in clockwise direction depending on a pathArray. You have to write a function **whereIsMyAnt(pathArray)** that should print the final co-ordinate of the ant and indicate if the path intersected (crossed or touched), in the following format:
ANT : (x, y) intersected : <yes/no>

Example 1 : pathArray is [1, 6, 3, 5, 4], the ant moves 1 step to right, then 6 steps downward, 3 steps left, 5 steps upwards and 4 steps to right. The output of the program would be

ANT : (2, -1) intersected : yes

Example 2 : pathArray is [2, 2, 2, 1] . The output of the program would be

ANT : (0, -1) intersected : no