Department of Computer Science & Engineering
I. I. T. Kharagpur
Programming and Data Structure (*Laboratory*) : CS19001
1st Year : 2nd Semester
Laboratory Test I (**Odd Machine Numbers**)
Section : 4/D *13th February, 2014* (**1445 - 1645 hrs**) *Marks:* [**6+6+7+6**]

Write a C program to solve the given problems. There are four parts. Write the first part and test it. Then write other parts and test. All the parts together form a single program. **Do not forget to print the input data**. Your file name should be '**Dmm1.c**', where '**mm**' is your machine number and '**1**' is for the *laboratory test one.* Send the file by **ftp** to **10.5.17.186** under the subdirectory **odd1** (of the remote machine). Write your **machine number**, **roll number**, **section** and **name** in the program header.

1. Read an integer n. Print the largest integer m less than or equal to n, and divisible by 17.
2. Print the digits (decimal) of n. If $n = -123$, the output is $3, 2, 1$.
3. Print the largest Fibonacci number f less than or equal to the absolute value of n, $|n|$. Do not use any function or array.

Input	Output
0	0
± 5	5
± 127	89

4. Let $f(x)$ be a real valued function and x_n is close to a real root of $f(x) = 0$. A better approximation of the root x_{n+1} is obtained by $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$. The process starts with x_0, a '*reasonable*' initial guess of the root and iterates to get better value of it.
   Write a C function `float pow2p5(float x)` that computes $x^{2.5}$ using the method mentioned above (Newton-Raphson) and returns the value.
   Call the function from `main()` with $|n|$ as the argument, and print the value of $|n|^{2.5}$ from `main()`. Do not use any mathematical library function. The *error* should be within 0.0001 per cent. As an example, $15^{2.5} = 871.421$
5. After you are satisfied, send the C program file (no output) to the remote machine (10.5.17.186) under the correct subdirectory (**odd1**).

Do not change name or type of the specified function.