Week 5 Lab

Dhaka University of Engineering & Technology, Gazipur

Computer Science and Engineering Department

CSE 1122 (Structured Programming Language Sessional)

These Programs illustrates on *Functions and Recursion* in C Language.

1. You may use the program template in Figure 1 to test your functions developed in this lab. The program contains a main ( ) which includes a switch statement so that the following functions can be tested by the user. Write the code for each function and use the suggested test cases to test your code for correctness. Use the test cases suggested to test your program and understand why these test cases are used. You do not need to do (extra) error checking on input.

```
#include<stdio.h>
void main(){
int choic e;
do {
printf("Perform the following functions:   \ n");
printf ("1: multiplication test \n");
printf ("2: quotient using division by subtraction    \ n");
printf ("3:  count  the  number of digits   \ n");
printf ("4: position of a digit \ n");
printf ("5: extract a ll odd digits \ n");
printf ("6: quit \n");
scanf("%d", &choice);
switch (choice) {
   case 1: /* add mulTest() call */
         break;
   case 2: /* add divide() call */
         break;
   case 3: /* add countDigits() call*/
         break;
   case 4: /* add position() call */
         break;
   case 5: /* add extractOddDigits() call */
         break;
   case 6: printf("Program terminating ….");
}
} while (choice < 6);
  }
  /* add function code */
}
```

Figure 1: Program template for Lab 05

Write a function that is to test student's ability to do multiplication. The function will ask a student 05 (Five) multiplication questions one by one and checks the answers. The function prints out the number of correct answers given by the student. The function *random ()* from the *standard C library* can be used to produce two positive one-digit integers (i.e. 1,2,3,4 …) in each question. A sample screen display when the function is called is given below:

How much is 6 times 7? **42**

How much is 2 times 9? **18**

How much is 9 times 4? **36**

……..

4 answers out of 5 are correct.

The typing in **bold** is the student's answer to a question. The function header is: **void mulTest ()**

Test cases: (1) give 5 wrong answers; (2) give 1 correct answer; (3) give more than 1 correct answer.

Write the function divide () which does division by subtraction and returns the quotient of dividing m by n. Both m and n are positive integers (i.e.1,2,3,4,…). Division by subtraction means that the division operation is achieved using the subtraction function. For example, divide(12,4) will be performed as follows: 12-4=8, 8-4=4, and then 4-4=0, and it ends and returns the result of 3 as it performs three times in the subtraction operation. No error checking on the parameters is required in the function.

The function header is: **int divide (int m, int n)**

Test cases: (1) m = 4, n = 7 (2) m = 7, n = 7 (3) m = 25, n = 7

Write a function to count the number of digits for a positive integer (i.e.1,2,3,4,…). For example, 1234 has 4 digits. The function countDigits ( ) returns the result.

The function header is: **int countDigits(int n)**

Test cases: (1) n: -12 (give an error message); (2) n : 123 (3) n : 121456;

Write the function position () which returns the position of the first appearance of a specified digit in a positive number n. The position of the digit is counted from the right and starts from 1. If the required digit is not in the number, the function should return -1.

For example, position (12415, 4) returns 3 and position (12, 3) returns -1. No error checking on the parameters is required in the function. The function header is: **int position(int n, int digit)**

Test cases: (1) n: 12345, digit: 3; (2) n: 123, digit: 4; (3) n: 12145, digit: 1;

Write a function extractOddDigits() which extracts the odd digits from a positive number n, and combines the odd digits sequentially into a new number. The new number is returned back to the calling function. If the input number n does not contain any odd digits, then returns -1. For examples, if n=1234567, then 1357 is returned; and if n=28, then –1 is returned. The function header is: **long extractOddDigits(long n)**

Test cases: (1) n : 12345; (2) n : 54123; (3) n : 246; (4) n : -12 (give an error message)

2. Write a function ***IntegerPower(base, power*)** that returns the value of $base^{exponent}$. For example, IntegerPower(3, 4) = 3 * 3 * 3 * 3. Assume that exponent is a positive, nonzero integer and base is an integer. Function IntegerPower(base, power) should use while loop to control the calculations. Do not use any math functions.

3. Write a function ***Reverse*** that takes an integer value from user and returns the number with its digits reversed. For example, if the given number is 5678, then function should return 8765.

4. The Fibonacci series

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Begins with the term 0 and 1 and has the property that each succeeding term is the sum of the two preceding terms. i.e.

Fibonacci(0) = 0 Fibonacci(1) = 1

Fibonacci(n) = Fibonacci(n-1) + Fibonacci(n-2)

Write a recursive function Fibonacci that calculates the nth Fibonacci number.

5. Edit the problem 2 and write a recursive function to solve the problem.

6. Write a program to swap the values of two numbers. Do this using call by reference method of function.