Structured Programming Language Sessional

Week1:

https://codeforwin.org/2017/08/install-configure-codeblocks-c-compiler.html

Week2: Data Type, Expression and Operators.

Standard I/O, Data Type, and Operators

(1) Write a program to convert temperature from Celsius to Fahrenheit.

Input: Temperature in Celsius

Output: Temperature in Fahrenheit

Sample input: Celsius: 5

Output: In Fahrenheit: 41 degree

Hints: C/5 = (F-32)/9

(2) Write a program that computes the salary of a part-time employee based on the following input:

- number of days worked
- number of hours per day (assume that he/she worked the same hours every day)
- the hourly wage

Sample input: Days: 5
Hours: 3
Wage: 230 BDT

Output: Salary: 3450 BDT

(3) Write a program to convert a given number of days into years, months, and days.

Sample input: 759

Output: 2 Years 0 Months 29 days

(4) Write a program to calculate the area of a circle.

Input: Radius of a circle

Output: Area of a circle

Sample input: 17

Output: 907.92 Unit

(5) Write a program to print the ASCII value of a character and vice versa.

Week3: - Conditional Statements, Looping Statements

1) Write a program to read an integer number from the user and check whether the number is EVEN or ODD using conditional operator (?:)

2) Write a program to find the largest among three numbers using conditional statement.

3) Write a program that will prompt the user for three sides of a triangle. Check and print whether the triangle is EQUILATERAL, ISOSCELES, SCALENE.

4) Write a program to find the sum of individual digits of a given integer number. Sum of the individual digits means adding all the digits of a number. For example, sum of digits of number 586 is 19 (5+8+6).

5) Write a program to read the height of a pattern from the user and display the pattern as below:

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
0 1 2 3 4 5
. . . . . . .
```

Example: for height 4, pattern will be:

```
0
0 1
0 1 2
0 1 2 3
0 1 2 3 4
```

6) Write a program to input integer numbers from the keyboard until the user enters a zero. Also, the program needs to calculate the summation of input numbers except for the last number.

7) Write a program to convert a given decimal number into an equivalent binary number.

<u>Homework-01</u>

1) We already know about basic data types. Declaring a variable of a specific type means allocating a particular range of storage for that variable. As I said, these ranges are actually machine and implementation dependent.

If you want to know the actual size of the allocated space of specific type, you need to use the sizeof() operator. See the following code segment:

```
#include <stdio.h>
int main() {
printf("Size of char is %d byte\n", sizeof(char));
printf("Size of int is %d byte \n", sizeof(int));
printf("Size of float is %d byte \n", sizeof(float));
printf("Size of double is %d byte", sizeof(double));
return 0;
}
```

In my computer output is:

2) Characters are actually represented in C as integer values. Each character is represented by its ASCII code (e.g., A = 65. B=66, etc). The table at the end of this document shows the printable ASCII characters and their corresponding ASCII code.

Printing a char variable using “%c” will print the character but printing it with “%d” will print the ASCII code. Similarly, printing an integer variable with “%c” will also print the character provided the value is within the range of characters. The following example demonstrates this.

```
#include <stdio.h>
int main()
{
char ch = 'A';
int code =65;
printf("The ASCII value of %c is %d\n", ch, ch);
printf("Print %c using its ASCII value %d", code, code);
return 0;
}
```

Output:

3) As I said in the class, there are many built-in math functions in <math.h>. So try to use these functions in your code. See the following simple program where 3 math functions sqrt(), cos() and pow() used.

Write a program that reads the lengths of two sides of a triangle and the angle between them and then calculates the length of the third side using the following formula (law of cosine):

$side_3 = \sqrt{(side_1{}^2 + side_2{}^2 - 2 \times side_1 \times side_2 \times \cos a)}$

```
#include <stdio.h>
#include <math.h>
 #define PI 3.1456
int main() {
double side1, side2, side3, angle;
printf("Enter side1:");
scanf("%lf", &side1);
printf("Enter side2:");
scanf("%lf", &side2);
printf("Enter angle in degrees:");
scanf("%lf", &angle);
angle = angle/180*PI; //convert angle to radians
side3 = sqrt(side1*side1+side2*side2-2*side1*side2*cos(angle));
//or side3 = sqrt(pow(side1,2)+pow(side2,2)-2*side1*side2*cos(angle));
printf("side3 = %.2f",side3);
return 0;
}
```

4) Consider the following mathematical expressions (with assumed initial values of a, b, c and x to be 6, 7, 8 and 9 respectively):

```
x = a + b++ * ++c – 6
x = ++x * b++ + 2 * x % ++b
x = c++ - --b * b++ / a
```

Run the program containing the expressions and confirm that the answers obtained are in line with the operator’s precedence rules studied in class.

```
#include <stdio.h>
int main()
{
int x = 9, a = 6, b = 7, c = 8;
x = a + b++ * ++c - 6;
x = ++x * b++ + 2 * x % ++b;
x = c++ - --b * b++ / a;
printf("%d %d %d %d", a, b, c, x);
return 0;
}
```

Assignment 01

Submission: We will evaluate the assignment before midterm in the lab class

Submission Rules:

- Create a sub-directory in your hard-drive using format “assign01_IIT_student_id” and do all the programming in this sub-directory.
- Name the source code using problem number. For example, p1.c is the source code name of problem 01.
- Use any standard test cases to test your program.
- Bring the zip folder with only source codes using a flash drive before last class of your midterm exam.

1) All of we already familiar with integer numbers. For a positive integer number, you need to solve the following problems

a) Write a function DigitCount that returns the number of digits of a given integer number.

Example: Input: 123, Output: 3

b) Write a function IntegerSum that returns the sum of individual digits of a given integer number.

Example: Input: 123 Output: 1+2+3 = 6

c) Write a function IntegerReverse that returns the reverse of a given integer number.

Example: Input: 123, Output: 321

d) Write a function EvenOddExtract that extracts all the even and odd digits of a given integer number

and print the numbers.

Example: Input: 1234, Output: Even: 2 4, Odd: 1 3

e) Write a function DigitPosition that returns the position of a digit of a given integer number from the right side. Here, you need to consider only the first occurrence of the digit.

Example: Input: 1543, Output: for digit 5, returns 3

f) As we already know how to check whether a number is a prime, you need to write a function called PrimeFactor to find all the prime factors of a given integer number. Here, the prime factor is the factor of a given number which is also a prime number. For example, 2 and 5 are the prime factors of 10.

For this problem, you need to create a menu to perform the operations. Use the following code segment as a help.

```
do {
printf("Press l: Digit Count");
printf("Press 2: Integer Sum");
printf(Press 3: Integer Reverse");
printf("Press 4: Even Odd Extract");
printf("Press 5: Digit Position");
printf("Press 6: Prime Factor");
printf("Press 7: Exit);

switch(choice) {
        case 1:
                DigitCount();
                break;
        case 2:
                IntegerSum();
                break;
        case 3:
                IntegerReverse();
                break;
        case 4:
                EvenOddExtract();
                break;
        case 5:
                DigitPosition():
                break;
        case 5:
                PrimeFactor():
                break;
        case 7: exit(0);
        }
        }while(...);
```

2) Write a function DisplayNumberSystem that prints a table of the binary, octal, hexadecimal equivalents of the decimal numbers in the range 0 through 16 (Table 01). If you are not familiar with these number systems, you can read attached pages on number system.

Decimal Number	Binary Number	Octal Number	Hexadecimal Number
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Table 01: Decimal, binary, octal, and hexadecimal equivalents

3) Write a program to generate a table of currency conversions from US dollars to BDT. Use title and column

headings. Assume the conversion rate 1 US$ = 82.6 BDT.

Allow the user to enter the starting value, ending value and the increment between lines in BDT. The starting value, ending value and the increment are all integer values. Use do_while loop to generate the conversion table.

Test Cases: (1) starting: 1, ending: 5, increment: 1; (2) starting: 0, ending: 50, increment: 5; (3) starting: 50, ending: 0, increment: 5 (treat this case as an error)

4) Write a program to display a pattern like below. Your program should take the height of a pattern from

the keyboard. For example, a pattern of height 5 is shown below.

```
AA
BBAA
AABBAA
BBAABBAA
AABBAABBAA
```

5) Given a positive integer N. Your task is to find a smallest positive integer Q so that the product of digits of Q is exactly equal to N. Here, the number Q should have minimum two digits. Suppose for input 10, output is 25.

Assignment 02

In this assignment, you need to solve two (02) programming contest type problem from Light OJ, a Bangladeshi programming contest site. You can choose any two problem from the beginner's category or any other category.

Light OJ: http://lightoj.com

Submission: You need to submit a printed copy of the problem, solution, submitted solution acceptance page, my submissions page, and user statistics page. Don’t forget to attach a cover page mentioning your student id.

Submission Deadline:

Assignment 03

Submission: We will evaluate the assignment on ---------- in lab class.

- Create a sub-directory in your hard-drive using format “assign03_IIT_student_id” and do all the programming in this sub-directory.

- Name the source code using problem number. For example, p1.c is the source code name of problem 01.

- Use any standard test cases to test your program.

- Bring the zip folder with only source codes using a flash drive on -----------.

1) Problem: Matrix Multiplication

Write a program to multiply two given matrix of type integer. Your program should have a function MMultiplication(), and you also need to pass two input matrix (other information if required) as a function parameter. Finally, you need to print the multiplication result on screen from main() function.

2) Problem: String Sort

Write a program to sort an array of strings. Your program should have a function StringCompare(str1, str2) to compare two string during the sorting process. You are not allowed to use any string related built-in function to solve this problem.

3) Problem: Big Number Summation

Write a program to calculate the summation of two large number. These numbers are as large as it is not possible to perform the sum operation using int/long int/long long int number. So, you have to use string to take two large number as input and show the result of the summation. You are not allowed to use any string related built-in function to solve this problem.

4) Problem: Binary Palindrome

The decimal number, 58510 = 10010010012 (binary), is palindromic (same number from left-to-right and from right-to-left) in its binary form. Write a program that will take decimal number <n> as input and determine whether its binary form is palindromic.

5) Problem: Converge to one

The following iterative sequence is defined for the set of positive integers:

n -> n/2 (n is even)

n -> 3n + 1 (n is odd)

Write a computer program that will take one input <n> and determine the length of the sequence before converging to 1. Also, output the sequence.

Input: 13

Output: length 10 (13 -> 40 -> 20 -> 10 -> 5 -> 16 -> 8 -> 4 -> 2 -> 1)

6) Problem: Entropy of a Sentence!

Suppose, we define the entropy of a sentence as the sum of ASCII values of all the alphanumeric characters in the string. For example, the entropy of string "I am Adam" is 650 (73 + 97 + 109 + 65 + 100 + 97 + 109). Note that the spaces and punctuations are not counted for computing the entropy. Write a program that will take a string S as input, and compute the value of entropy of S.

7) Problem: Concatenate

Write a program, that will take two strings S1, and S2 as input. The program will also ask the user for an integer number N. Finally, the program will do the following: (You are not allowed to use any string related built-in function to solve this problem)

- if N <= 0, then the program will concatenate S2, S1 to produce another string.

- if N >= length(S1), then the program will concatenate S1, S2 to produce another string.

- Otherwise, the program will insert S2 after the N-th character of S1 to produce another string.

Sample Input/Output:

```
S1 = "MADAM"
S2 = "mam"
N = -2, output: "mamMADAM"
N = 10, output: "MADAMmam"
N = 3, output: "MADmamAM"
N = 1, output: "MmamDAM"
```

8) Problem: Ranking Student

Design a student ranking program. A student has 4 subjects, which are Math, English, Physics, and Computer. Your program should prompt the user to enter marks of each subject for each of the students and then sort these students according to the total points they get in descending order. The skeleton code should be like the following:

```
struct student {
char name[40];
int course[3];
int total;
};
```

```
int main(){
struct student students[3];
.......
 return 0;
}
```

The structure student keeps the information of a student, which includes student name, 4 Subject marks, and total points the student gets. Since many students' information will be entered, we have to declare a student structure array at the beginning of the main function to keep the information.

Sample Input:

```
Enter how many students> 3
Student 1's name> Mary
Physics> 62
Math> 99
English> 63
Computer> 87
```

```
Student 2's name> Alice
Physics > 98
Math> 87
English> 84
Computer> 99
```

```
Student 3's name> Bob
Physics > 92
Math> 60
English> 58
Computer> 62
```

Output:

```
Rank    Physics  Math    English  Computer    Total   Name
1       98       87      84       99          368     Alice
2       62       99      63       87          311     Mary
3       92       60      58       62          272     Bob
```

9) Problem: Frequency of Letters

A file named "text.txt" contains lines of text. Write a program to open the file and read the characters one after another and output the frequency of vowels and consonants.

Test Data: Suppose, the file "text.txt" contains the following text:

We love Bangladesh. This is our homeland.

Expected Output:

```
No. of Vowels: 13
No. of Consonants: 20
```

10) Problem: GCD(a, b)

The greatest common divisor (gcd) of two non-negative integers a and b (which are not both equal to 0) is the greatest integer d that divides both a and b. You need to write a program to find the gcd(a, b) of two integers.

Input: First line of the input contains an integer N (≤ 50), denoting the number of test cases. Rest of the lines includes test cases, where each case contains two non-negative integers a and b separated by space.

Output: For each case, print the case number and gcd of two integers.

Sample Input

```
3
8 12
54 24
45 54
```

Output

```
Case 1: 4
Case 2: 6
Case 3: 9
```