

**Department of Computer Science and Engineering**  
**Bangladesh University of Engineering and Technology**  
**CSE 105 : Structured Programming Language**

**Instructor : Md. Shahrear Iqbal**

## **Assignment 1**

### **Basics:**

1. Input two numbers and work out their sum, average and sum of the squares of the numbers.
2. Input and output your name, address and age and output to the screen.
3. Write a program that finds out the largest and smallest values from a set of 10 inputted numbers.
4. Write a program to read a "float" representing a number of degrees Celsius, and print as a "float" the equivalent temperature in degrees Fahrenheit. Print your results in a form such as  
  
100.0 degrees Celsius converts to 212.0 degrees Fahrenheit.
5. Write a program to print several lines (such as your name and address). You may use either several printf instructions, each with a newline character in it, or one printf with several newlines in the string.
6. Write a program to read a number of units of length (a float) and print out the area of a circle of that radius. Assume that the value of pi is 3.14159 (an appropriate declaration will be given you by ceildh - select setup).

Your output should take the form: The area of a circle of radius ... units is .... units.

If you want to be clever, and have looked ahead in the notes, print the message Error: Negative values not permitted. if the input value is negative.

7. Given as input an integer number of seconds, print as output the equivalent time in hours, minutes and seconds. Recommended output format is something like

7322 seconds is equivalent to 2 hours 2 minutes 2 seconds.

# Conditionals

1. Write a program to read two characters, and print their value when interpreted as a 2-digit hexadecimal number. Accept upper case letters for values from 10 to 15.
2. Read an integer value. Assume it is the number of a month of the year; print out the name of that month.
3. Given as input three integers representing a date as day, month, year, print out the number day, month and year for the following day's date.  
Typical input: 28 2 1992 Typical output: Date following 28:02:1992 is 29:02:1992
4. Write a program which reads two integer values. If the first is less than the second, print the message up. If the second is less than the first, print the message down. If the numbers are equal, print the message equal. If there is an error reading the data, print a message containing the word Error and perform `exit( 0 )`;

# Looping and Iteration

1. Write a program to read in 10 numbers and compute the average, maximum and minimum values.
2. Write a program to read in numbers until the number -999 is encountered. The sum of all number read until this point should be printed out.
3. Write a program which will read an integer value for a base, then read a positive integer written to that base and print its value.  
Read the second integer a character at a time; skip over any leading non-valid (i.e. not a digit between zero and ``base-1'') characters, then read valid characters until an invalid one is encountered.

Input	Output
10 1234	1234
8 77	63 (the value of 77 in base 8, octal)
2 1111	15 (the value of 1111 in base 2, binary)

The base will be less than or equal to 10.

4. Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed.  
Typical output might be:  
Initial value is 9  
Next value is 28  
Next value is 14

```

Next value is 7
Next value is 22
Next value is 11
Next value is 34
Next value is 17
Next value is 52
Next value is 26
Next value is 13
Next value is 40
Next value is 20
Next value is 10
Next value is 5
Next value is 16
Next value is 8
Next value is 4
Next value is 2
Final value 1, number of steps 19
If the input value is less than 1, print a message containing the word
    Error
and perform an
    exit( 0 );

```

5. Write a program to count the vowels and letters in free text given as standard input. Read text a character at a time until you encounter end-of-data.

Then print out the number of occurrences of each of the vowels a, e, i, o and u in the text, the total number of letters, and each of the vowels as an integer percentage of the letter total.

Suggested output format is:

```

Numbers of characters:
a 3 ; e 2 ; i 0 ; o 1 ; u 0 ; rest 17
Percentages of total:
a 13% ; e 8% ; i 0% ; o 4% ; u 0% ; rest 73%

```

Read characters to end of data using a construct such as

```

char ch;
while(
    ( ch = getchar() ) >= 0
) {
    /* ch is the next character */ ....
}

```

to read characters one at a time using `getchar()` until a negative value is returned.