Lab instructions

Week 06

Introduction to Programming
ECS 102, 2018-19 Semester II

IISER Bhopal

use_getchar.c

(a) Write a program using **getchar** to check whether the input is an upper case character. Check also lower case characters and digits too. Use **putchar** to print the character. Hint: Use ASCII table.

(b) Include header file **ctype.h**. Read a character *c* and print isalpha(c) and isdigit(c) to check alphabet/digit. Check a character whether that is alphanumeric using isalnum(c) function. Use the functions islower, isupper, tolower, toupper.

(c) Write a program to demonstrate the process of multiplication as shown below. Use defensive programming style while taking inputs of digits only.

```
                    45
              x     37
              ----------
7x45 is           315
3x45 is          135
              ----------
Add              1665
```

use_scanf.c

(a) Define a string of characters of maximum allowable length 15 as **char name[15]**. Use scanf("%15c", name), scanf("%s", name), and scanf("%15s", name) to read user inputs. Print them.
   (a) Check whether space and newline are allowed in the inputs.
   (b) What happens if you input more or less than 15 characters.

(b) You can use %[characters] to allow only allowable characters. Allow a-z, A-Z and space.

(c) You can use %[^characters] not to allow the characters. Don't allow newline.

(d) Use scanf("%2d %5d", &a, &b);
   (a) Input 12 34567. Print a and b.
   (b) Input 34567 12. Print a and b. Use another scanf statement scanf("%d", &c) followed by a printf statement to print c. Explain your answer.

(e) See what happens if you scan integer but input float e.g., 12.2, 089, 0.9, .9.

(f) See what happens if you scan float but input multiple dots.

(g) User inputs 3_5_1. Write a scanf statement to read 3, 5, and 1 in three variables separately.

use_printf.c

(a) Print an integer 1234 using the following formats: %d, %6d, %2d, %-6d, %06d, %+6d, %+-6d.

(b) Use %-10.2e to print 98.7654. %g prints the number in the shortest of %f and %e representations. Use 321.65 to justify it.

(c) Print a string “NEW DELHI 11001” using the following format: %s, %20s, %20.10s, %.5s, %-20.10s, %5s.

(d) See the output using printf("%*c\n", 2, 'a’); Generate the following output using a for loop.

```
 a
  a
   a
    a
     a
```

(e) See the output using printf("%*.*s\n", 5, 1, "ABCDE"); Generate the following using a for loop.

```
A
AB
ABC
ABC
ABCDE
```

(f) Print “Well done!” **including** the double inverted commas.

inventory_report.c

Generate the following inventory report. Take each row from user input at a time and use the return value of scanf statement to verify the successful read of the inputs. In case of mistake, user is allowed to input again. Value = Quantity*Rate and Total in the end is the sum of all the values. Align the data in the columns appropriately.

Code	Quantity	Rate	Value
F105	275	237.00	
H220	1	535.50	
I109	52	5.30	
M331	5	1000.00	

Total

use_bool.c

For using **Boolean** data type you need to use the header file **stdbool.h**. You can declare such data type as **bool** b1, b2;

You can define b1 = true; b2 = false;

(a) Print b1 and b2 using printf as integers.

(b) Define b1 as zero, any positive number and any negative number. Use printf to print.

(c) Define an integer i and assign $i = b1 + 90$ for different cases in part (b). Print i.

(d) Operate b1 and b2 with different bitwise operators: AND (&), OR (|), XOR (^).

(e) Define two integers and perform bitwise operations: AND (&), OR (|), XOR (^).