

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI (RAJ.)**  
**CS F111 Computer Programming**  
**LAB SESSION #5**  
(Control Flow: Branching and Loops)

Create a directory “**lab5**” inside “**myprogs**” directory for all your programs in this week’s lab. We will write programs related to branching and looping.

### **Branching Constructs in C:**

We have studied **if... else** and **switch... case** constructs that are available in C for branching. We have also studied the **ternary operator** which is an alternative for if... else in some cases. Please refer to the slides uploaded on Nalanda related to Module 6 to understand the syntax and usage of the above constructs. You can also go through these links in order to get started.

[If..else](#)

[Switch](#)

[Ternary operator](#)

Let us now solve a few problems:

1. Write a C program to input week number (1-7) and print day of week name **using switch case**. For example. If the user inputs 2, the program should output the second day of the week i.e., “Tuesday”. *[Note: for the purpose of this exercise you can consider that a week starts from Monday.]*
2. Write a program that takes the user’s age as an input. Then using the **ternary operator (?:)**, it decides whether the user can vote or not and displays a message accordingly on the screen. **Do not use if statements**. *[Hint: A person’s age must be equal to or above 18 years for him/her to be able to vote]*
3. Write a C program that reads four integers and displays the pair with the largest sum. For example, if the user enters 10, -8, 17 and 5, the program should display  $10 + 17 = 27$ . *[Note: Use nested-if-else]*

### **Iterative Constructs in C:**

C supports three kinds of iterative constructs: while loop, do... while loop and for loop. Please refer to the lecture slides uploaded on Nalanda to understand the usage and syntax of these loops along with their differences. You can also go through this links to get started: [Loops](#)

Now, let us write a few programs:

4. Following is a C program that prints the first N natural numbers, one per line, where N is user input. It uses a while loop.

```
int main(){
    sum = 0;           /* let sum be zero, to begin with */
    num = 1;           /* start adding up from 1 */
    while (num <= N)    /* as long as num has not exceeded N...*/
    {
        printf("%d\n", num); /* print out the number */
        num++;              /* increment it to the next value*/
    }
    printf("Bye\n");
```

```

    return 0;
}

```

You should understand that the condition **num <= N** is checked for every iteration of the loop, and if the condition is TRUE (i.e., the number is smaller than N), then the loop continues; otherwise, control comes out of the loop and then goes to the next statement after the loop (in this case, the **printf("Bye\n");** statement). Please copy the above program into a .c file, compile and execute it.

Now, let us now see how to print only the odd numbers from 1 to N:

```

int main(){
    num = 1;
    while (num <= N) /* as long as num has not exceeded N... */
    {
        (num % 2)? printf("%d\n", num):0; // print num if it is odd
        num++;                          // increment num to next number
    }
    return 0;
}

```

Copy the program into a .c file, compile and execute.

- (a) Would it have made any difference if you had written `++num;` instead of `num++;` above? Please try.
- (b) What would have happened if you missed the statement `num++;` altogether? Please try.
- (c) Write the above program using `do... while` loop
- (d) Write the above program (printing odd numbers) using `for` loop.

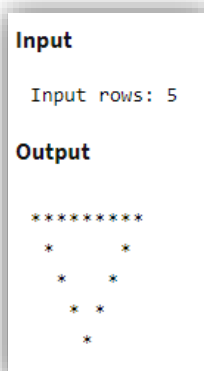
5. Without running the following code, write down what you expect the output to be. Then, run the code and compare it to the output you expected. Note down your observations. What is the problem with this code?

```

int i;
while (i = 2) {
    printf("Some even numbers: %d %d %d\n", i, i+2, i+4);
    i = 0;
}

```

6. Using loops, write a C program to create a hollow star pattern as shown below. You should be writing a generic code that works for any number of rows as input. Try following the code of a similar example on slides to print this pattern.



## Additional Practice Exercises

7. Write a program that simulates a physical calculator. The program should read the symbol of an arithmetic operation (any of the five: +, -, \*, / and %) and two integers and display the result of the arithmetic operation. Remember to incorporate an error check for division by zero. Solve this using (a) if statement, (b) switch statement.
8. Write a program that reads a four-digit positive integer and uses the ?: (ternary) operator to display a message whether it can be read the same in reverse order or not (e.g., the number 7667 can be). Do not use if statements.
9. Write a complete C program that will take one of the following options from the user, and performs jobs accordingly:
  - Option-1: Convert from degree Celsius to deg. Fahrenheit
  - Option-2: Convert from deg. Fahrenheit to deg. Celsius
  - Option-3: Quit the program execution

For deciding which job to execute based on the option number provided by the user, you need to use a switch statement.

You must also ensure the user inputs a temperature value greater than -273.15 deg. C (for option-1) and -459.67 deg. F (for option-2), by repeatedly asking the user to input a valid temperature as long as it not within the limit.

10. Write a program that reads an integer value for n and then sums the integers from n to 2\*n if n is nonnegative, or from 2\*n to 3\*n if n is negative.
11. Until interrupted, the following code prints TRUE FOREVER on the screen repeatedly:

```
while (1)
    printf(" TRUE FOREVER ");
```

Write a simple program that accomplishes the same thing, but use a for statement instead of a while statement. The body of the for statement should contain just the empty statement ";".