# ITEC113 Algorithms and Programming Techniques

# Lab 3 Loops

## Pre lab Questions

**Task 1 :** **Design an algorithm that prints all even numbers between 2 and 100 on screen.**

1. The following is an incomplete flowchart designed for task 1. Complete the missing sections of the flowchart.



1. Convert the flowchart of part (a) to pseudocode

|  |
| --- |
|  |

1. The following pseudocode is a variation of the previous answer for task 1. Complete the missing sections according to the task definition.

|  |
| --- |
| num 🡨 …..WHILE ……………………. ………………………… DISPLAY numENDWHILE |

1. Convert the pseudocode given in section (c) to flowchart

|  |
| --- |
|  |

1. Draw a flowchart for task1 using DO WHILE Loop

|  |
| --- |
|  |

**Task 2 :** Write a program that calculates the average mark of all students in a class. Assume that the last value read is the sentinel -1. An example run is given below:

Please enter a mark or -1 to stop : 90

Please enter a mark or -1 to stop : 20

Please enter a mark or -1 to stop : 80

Please enter a mark or -1 to stop : 50

Please enter a mark or -1 to stop : 60

Please enter a mark or -1 to stop : -1

The average is 60

Answer :

We need to read a number of values from keyboard, therefore we need to use a loop for this question. We do not know how many times the loop will be executed. We say the loop will be executed an indefinite number of times determined by the sentinel value -1. As soon as the user enters -1, we will exit the loop. Such loops are not counter-controlled loops that we have been using so far. A loop whose exit condition is determined by a special value is called a sentinel-controlled (or indefinite) loop and the special value that is used to cause the loop to exit is called the sentinel value.

Solution :This is a very common solution used for sentinel controlled loops. Please notice that we read the mark twice, once inside the loop and once outside the loop. The exit condition of the loop is mark =-1. As soon as the user enters -1 for mark, the loop is exited.

Ans:

Notes:

In C, we have a short hand notation for some common operations

Computing running sums/subtractions/divisions/multiplications is also very frequent.

sum=sum+mark; can be written as sum+=mark;

sum=sum-mark; can be written as sum-=mark;

sum=sum\*mark; can be written as sum\*=mark;

sum=sum/mark; can be written as sum/=mark;;

Incrementing or decrementing by 1 is a very frequent operation:

count=count+1; can be written as count+=1; or better yet count++ or ++count

count=count-1; can be written as count+=1; or better yet count++ or ++count