SWE20004 -Technical Software Development

Assignment 2

(This assignment is worth 6% of the subject's total assessment marks)

Due Date: Monday 6th May 2019 at 11:59 pm

Submission instructions

Submit a soft copy of required documents through Canvas.

Introduction:

This assignment requires the knowledge of variables, constants, stream input / output, output formatting, assignment statements, expressions, sequence, selection (if/switch statements), loops, arrays, functions etc.

You should complete the **problem**.
Need to submit **Task 5.3 or Task 5.4** from week 5, **Task 7.3 and Task 7.4** from week 7.

Problem

Some credit card companies use Luhn algorithm to detect invalid credit card numbers. The key thing in this algorithm is to find a check digit, which should be the rightmost digit in any credit card number.

In this task, you have to implement a version of Luhn algorithm using the instructions given below.

Step1: Read in credit card number as a series of digits into an array

Ask the user to enter series of one digit positive numbers into a one dimensional array, the number -1 is used to indicate the end of series, no need to read in -1 into the array. The maximum size of the array is set to 20.

Step2: Find sum1 (main should call a function passing array and size)

Ignoring the last (right most digit) `check digit` of the credit card number, and moving left, double the value of every second digit and find the sum of these doubled numbers. If the result of the doubling operation is a two digit number, you should add the digits of the doubled number before finding the sum. You must printout the numbers for sum1, these should be in the order (from left to right) as it appears on credit card.

Step 3: find sum2 (main should call a function passing array and size)

Find the sum of all other numbers (last digit is not included in this sum as well). You must printout the numbers for sum1, these should be in the order (from left to right) as it appears on credit card.

Step 4: Calculate check sum

Compute the total of sum1 and sum2 and multiply the result by 9, `checksum` is found by extracting the rightmost digit.

Image courtesy: http://echeck.org/citi-simplicity-credit-card-review/

5	4	2	4	1	8	0	1	2	3	4	5	6	7	8	9
5+5=10 1+0=1		2+2=4		1+1=2		0		2+2=4		4+4=8		6+6=12 1+2=3		8+8=16 1+6=7	sum1=1+4+2+0+4+8+3+7 = 29
	4		4		8		1		3		5		7		sum2 = 4+4+8+1+3+5+7 = 32
															Total=29+32 = 61

61 × 9 = 549 ← 9 is the check sum digit.

Other requirements:

- This assignment must be written in C++
- Your code must have appropriate header(multiline/block) comments including your name and student number, the name of the .cpp file, the purpose of the program, brief explanations of variables and explanations of any code, which is not obvious to another programmer, summarising the input, output and local variables as well as expressions used in your program and test data.
- Include inline (single-line) comments throughout the program describing important statements.
- Use appropriate and descriptive variable following the naming rules and conventions.
- Write a brief (no more than several pages) report, which illustrates your program design (algorithm or flowchart, identification of variables, constants) and include evidence of testing – screen shots or pasted output text of several tests, and the contents of the .cpp file
- Marks will be allocated depending on the amount of original work submitted. Marks will be deducted for plagiarised and/or un-attributed work.

Assignment submission:

Submissions through **Canvas** must be made on or before the due date/time.

Each submission should have two files.
1. A report (name of the report should be your student number, eg: 1012546.docx)

This report will be used for plagiarism check using turnitin software. No marks will be given if this report is missing for plagiarism check. Report must (***.doc/docx, .pdf*** or ***.rtf*** format – use SWE20004_AssignmentReportTemplate) contain:

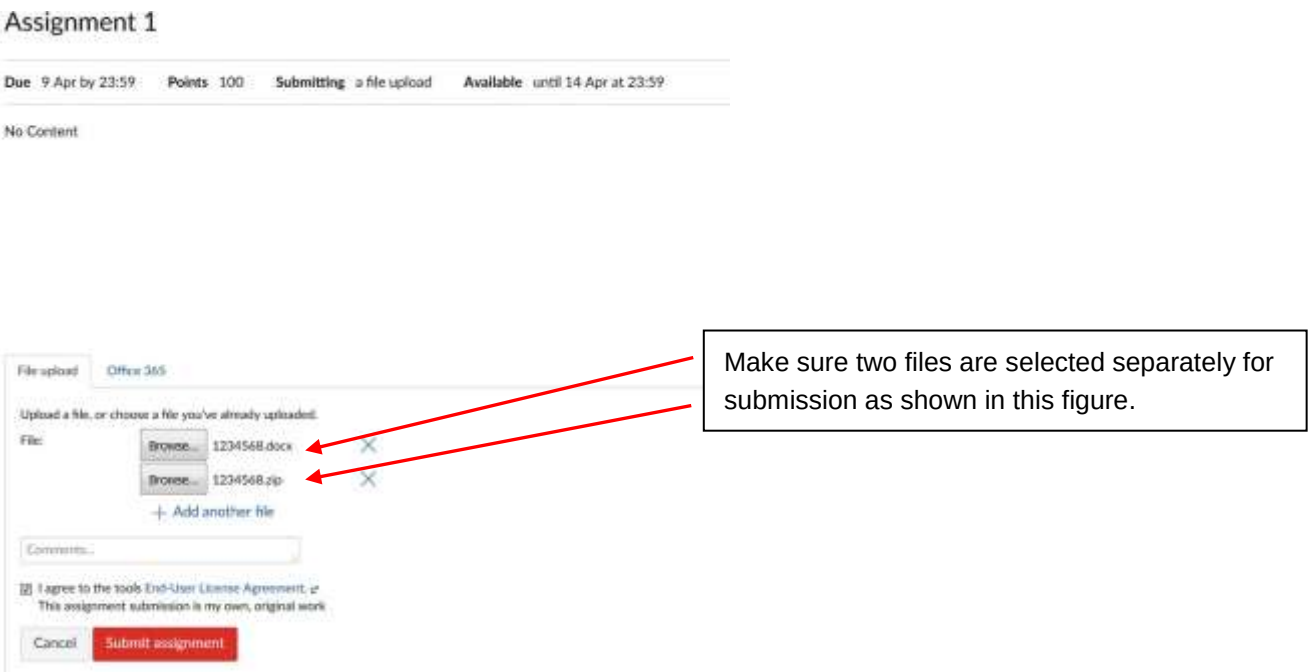
- Description of the problem,
- Description of inputs and outputs,
- A description of the algorithm(s) used in pseudocode or a flowchart
- A copy of the contents of the **.cpp** file
- Pasted **text output** or **screen shots** of the working program resulting from the testing of the program

2. A ***.zip*** file (name of the zip file should be your student number, eg: 1012546.zip) containing:

a) The actual program (.**cpp** source code) with comments.
   - The name of your **.cpp** file must be **your student number.cpp (eg**. 386123.cpp) for the assignment question.

b) Task5.3.cpp or Task 5.4.cpp
c) Task7.3.cpp
d) Task7.4.cpp
   - If you are using Visual Studio for this assignment, do not include the solution files, folders or **exe** files.

Submissions larger than 5 MB will not be accepted.

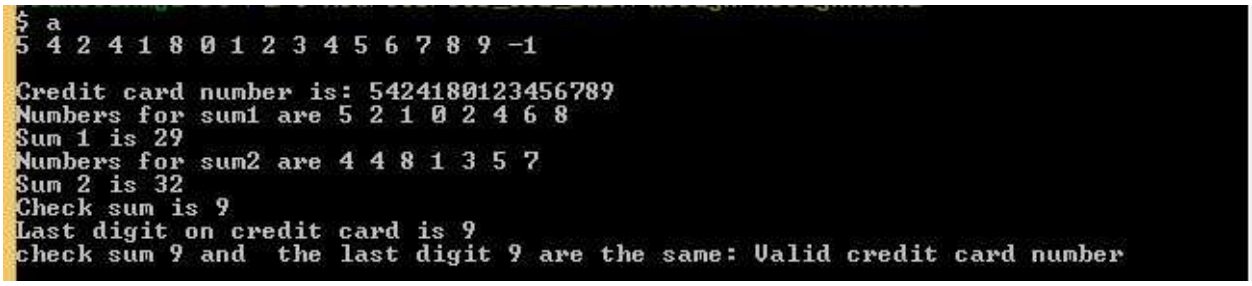


Rubrics

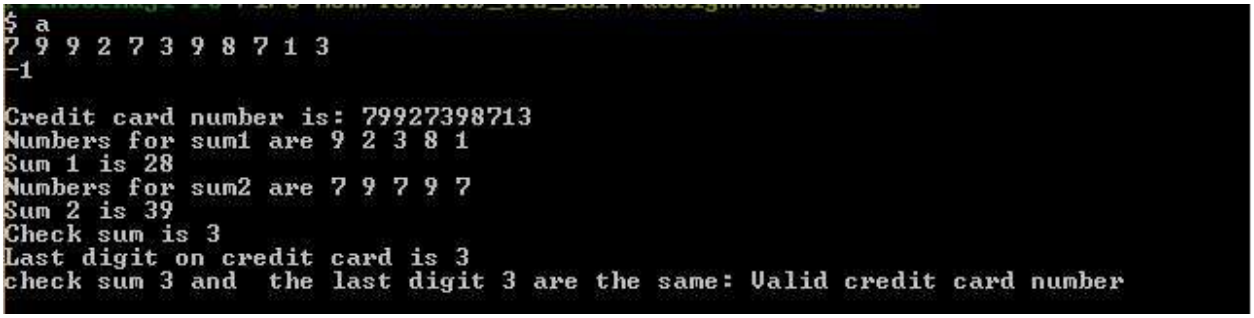
Requirement	Weight (%)	Mark awarded
Program specification and design: (1) Specification of what the purpose and functionality of the program is (2) Design presented as pseudocode or flowchart	5	
Documentation: (1) Header comments describing (a) the purpose and function of the program (b) Subject and assignment details (c) personal details of the author. (2) Inline comments where appropriate to describe crucial program statements	5	
Coding: Program includes the following elements in order to meet the requirements - Necessary preprocessing directives - Namespace specification - Correct main() function header - Opening and closing braces for the body of main() - Return statement to terminate main() and program - Variable and constant declarations - Input and Output statements - Proper use of arrays and functions - Appropriate use of formatting - Processing including assignment statements, expressions, formulae and calculations as necessary - Appropriate use of sequence, selection - Correct syntax - Correct logic - No runtime errors - Appropriate use of identifier naming rules and conventions - Use of appropriate indentation - overall coding structure and efficiency	40	
Lab tasks: Task 5.3.cpp or Task 5.4	10	
Task 7.3	10	
Task 7.4	20	
Testing: A sufficient number of appropriate test cases performed and corresponding screen shots provided as evidence	10	
Deductions: Marks will be deducted accordingly for invalid submission of required documents such as missing files, corrupt files, incorrect file formats, use of programming language(s) other than C++ and late or non-submission		
Total Assignment mark (out of 100)	100	
Contribution to unit mark (out of 6)	6	

Screenshots showing working program:

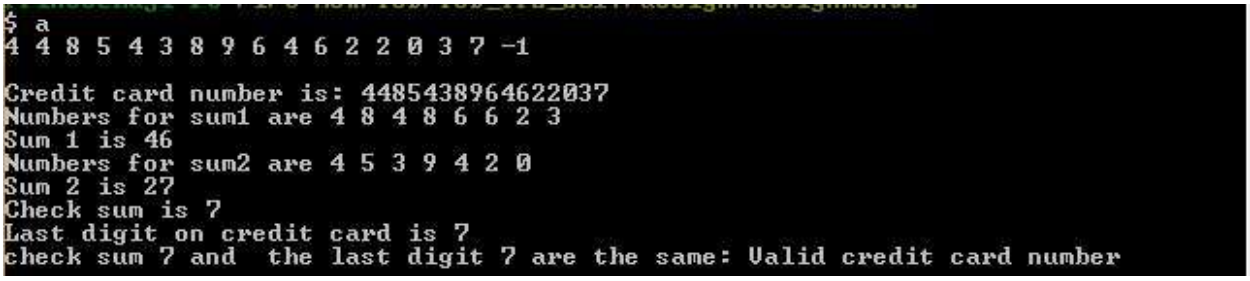
Screen shot for the card number shown above

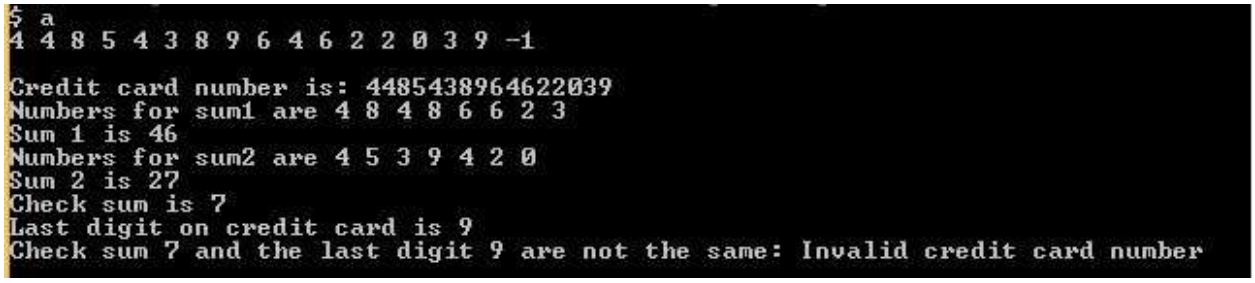


Screen shot for the card number shown on wiki page



eg 3:





Need help?

Talk to your tutor, visit programming Help Desk in ATC620 (8.30 am – 7 pm Monday to Friday)

TSD Tutors - Help Desk Times:

Syeda: Tuesday 12.30 -1.30 and Friday 10.30 - 1.30 pm

Srikanth: Friday 10.30 -12.30 pm

Sharon: Monday 8:30 - 10:30 am

Kai: Tuesday: 12:30 - 4:30 pm

Michael: Tuesday 2:30 - 4:30, Wednesday 3:30 - 5:30 pm

Gavin: Thursday 12.30 – 2.30 pm

Rida: Thursday 12 - 2 pm