PROGRAMMING

TRINH THI DIEU HUYEN

GDD18606

ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 1: Programming		
Submission date	11/8/2019	**Date Received 1st submission**	11/8/2019
Re-submission Date		**Date Received 2nd submission**	
Student Name	Trinh Thi Dieu Huyen	**Student ID**	GDD18606
Class	GCD0826	**Assessor name**	Tran Trong Minh
Student declaration I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.			
		Student's signature	

Grading grid

P1	M1	D1

☼ **Summative Feedback:**

☼ **Resubmission Feedback:**

Grade:

Assessor Signature:

Date:

Lecturer Signature:

Table of Contents

I. Definition of algorithm and outline the process in building an application.

1. Definition of algorithm.

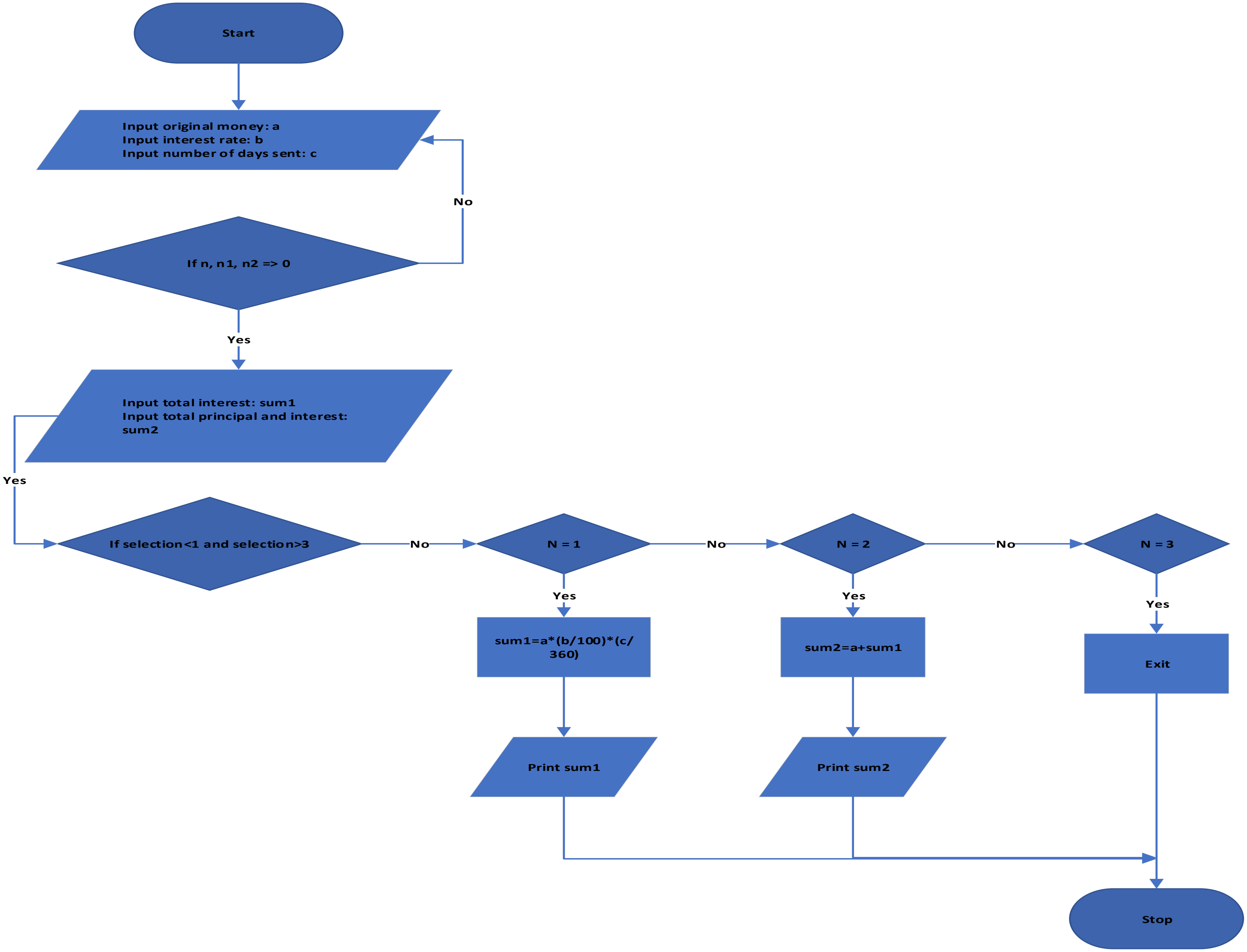
- Algorithm is a step by step method of solving a problem. It is commonly used for data processing, calculation and other related computer and mathematical operations.
- Algorithm is also used to manipulate data in various ways, such as inserting a new data item, searching for a particular item or sorting an item.
- Algorithm is a detailed series of instructions for carrying out an operation or solving a problem. In a non-technical approach, algorithms is use in everyday tasks.
- Technically, computers use algorithms to list the detailed instructions for carrying out an operation. For example, to compute an employee’s paycheck, the computer uses an algorithm. To accomplish this task, appropriate data must be entered into the system. In terms of efficiency, various algorithms are able to accomplish operations or problem solving easily and quickly.

2. Outline the process in building an application.

Purpose: This program helps you to calculate quickly and conveniently.

Calculate bank interest rates:

- ❖ Input variables:
  - ➢ Original money.
  - ➢ Interest rate.
  - ➢ Number of days sent.
- ❖ Input selections:
  - ➢ Total interest.
  - ➢ Total principal and interest.
  - ➢ Exit programer.
- ❖ Flowchart:

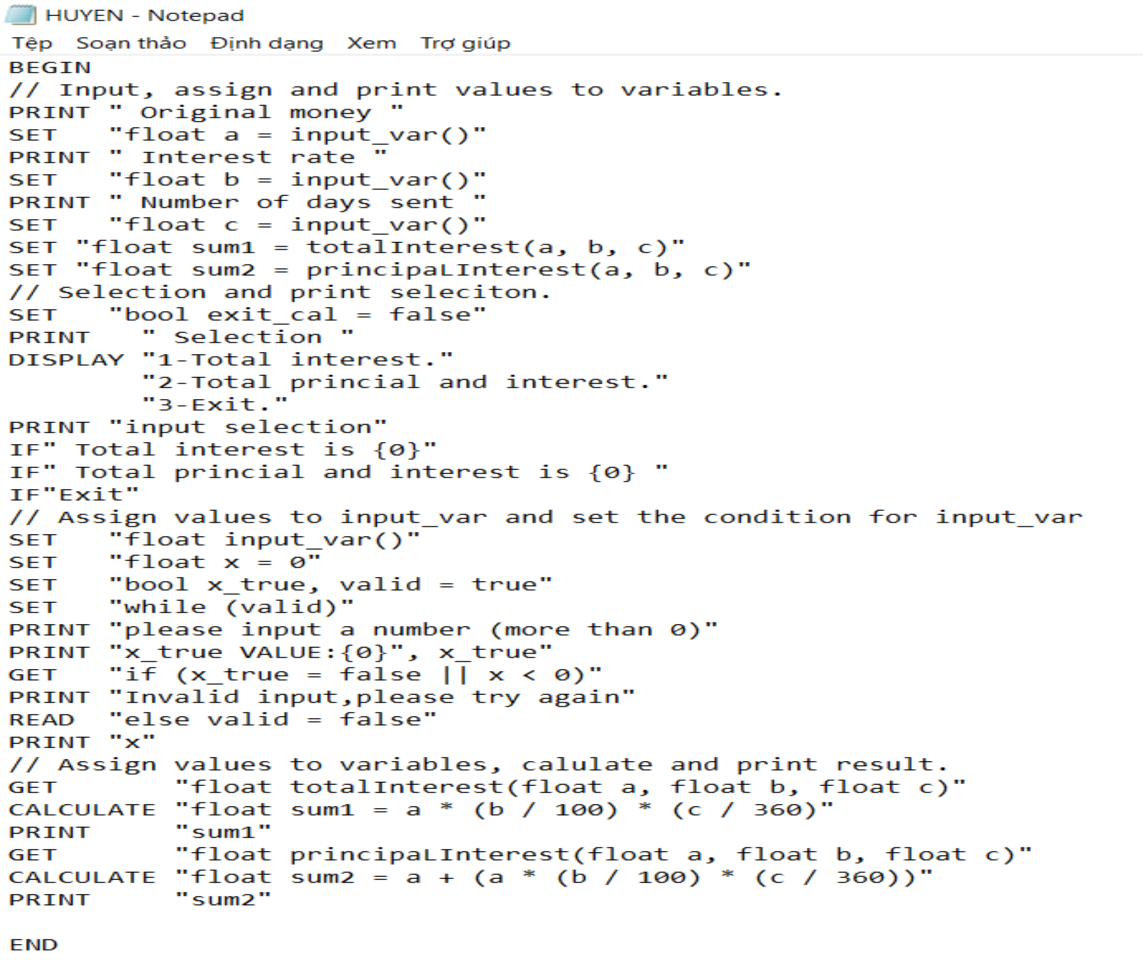

Start
Input original money: a
Input interest rate: b
Input number of days sent: c
No
If n, n1, n2 => 0
Yes
Input total interest: sum1
Input total principal and interest: sum2
Yes
If selection<1 and selection>3
No
N = 1
No
N = 2
No
N = 3
Yes
Yes
Yes
sum1=a*(b/100)*(c/360)
sum2=a+sum1
Exit
Print sum1
Print sum2
Stop

II. The steps taken from writing code to execution (Pseudocode).

1. What is Pseudocode.

   Pseudocode is an informal high-level description of a computer program or algorithm. It is written in symbolic code which must be translated into a programming language before it can be executed.

2. Pseudocode of calculate bank interest rates.



```
BEGIN
// Input, assign and print values to variables.
PRINT " Original money "
SET   "float a = input_var()"
PRINT " Interest rate "
SET   "float b = input_var()"
PRINT " Number of days sent "
SET   "float c = input_var()"
SET "float sum1 = totalInterest(a, b, c)"
SET "float sum2 = principalInterest(a, b, c)"
// Selection and print seleciton.
SET   "bool exit_cal = false"
PRINT   " Selection "
DISPLAY "1-Total interest."
        "2-Total princial and interest."
        "3-Exit."
PRINT "input selection"
IF" Total interest is {0}"
IF" Total princial and interest is {0} "
IF"Exit"
// Assign values to input_var and set the condition for input_var
SET   "float input_var()"
SET   "float x = 0"
SET   "bool x_true, valid = true"
SET   "while (valid)"
PRINT "please input a number (more than 0)"
PRINT "x_true VALUE:{0}", x_true"
GET   "if (x_true = false || x < 0)"
PRINT "Invalid input,please try again"
READ  "else valid = false"
PRINT "x"
// Assign values to variables, calulate and print result.
GET       "float totalInterest(float a, float b, float c)"
CALCULATE "float sum1 = a * (b / 100) * (c / 360)"
PRINT     "sum1"
GET       "float principalInterest(float a, float b, float c)"
CALCULATE "float sum2 = a + (a * (b / 100) * (c / 360))"
PRINT     "sum2"

END
```

III. Examine the implementation of an algorithm in a suitable language (C#). Evaluate the relationship between the written algorithm and the code variant.

1. Examine the implementation of an algorithm in a suitable language (C#).

   ❖ Code.

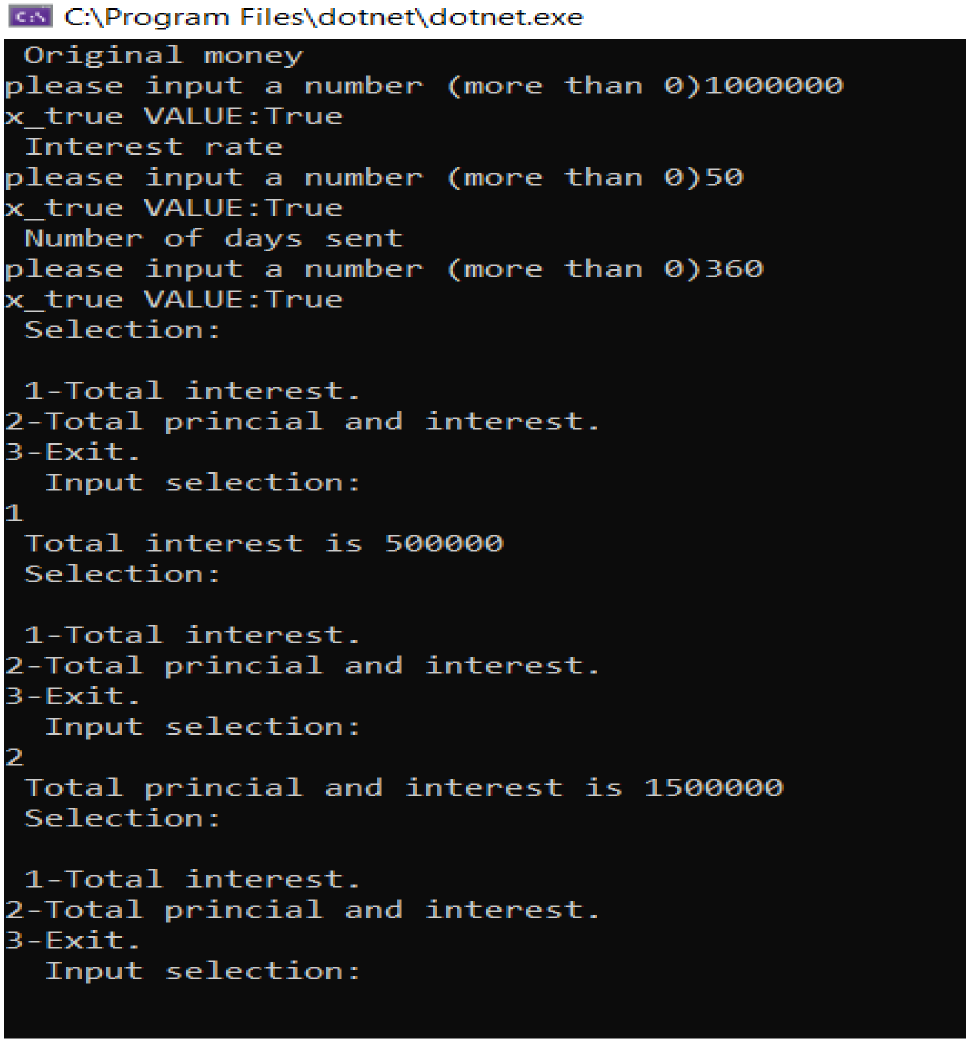
```
using System;

namespace calculateBankInterestRates
{
    class calculateBankInterestRates
    {
        static void Main(string[] args)
        {

            Console.WriteLine(" Original money ");
            float a = input_var();
            Console.WriteLine(" Interest rate ");
            float b = input_var();
            Console.WriteLine(" Number of days sent ");
            float c = input_var();
            float sum1 = totalInterest(a, b, c);
            float sum2 = principalInterest(a, b, c);

            bool exit_cal = false;
            while (!exit_cal)
            {
                Console.WriteLine(" Selection:\n ");
                Console.Write(" 1-Total interest.\n2-Total principal and interest.\n3-Exit.\n ");
                Console.WriteLine(" Input selection: ");
                string selection = Console.ReadLine();
                switch (selection)
                {
                    case "1":
                        Console.WriteLine(" Total interest is {0}", sum1);
                        break;
                    case "2":
                        Console.WriteLine(" Total principal and interest is {0} ", sum2);
                        break;
                    case "3":
                        break;
                    default:
                        Console.WriteLine("please enter again");
                        break;
                }
            }
        }
        static float input_var()
        {
            float x = 0;
            bool x_true, valid = true;
            while (valid)
            {
                Console.Write("please input a number (more than 0)");
                x_true = float.TryParse(Console.ReadLine(), out x);
                Console.WriteLine("x_true VALUE:{0}", x_true);
                if (x_true = false || x < 0)
                {
                    Console.WriteLine("Invalid input,please try again");
                }
                else valid = false;
            }
            return x;
        }
        static float totalInterest(float a, float b, float c)
        {
            float sum1 = a * (b / 100) * (c / 360);
            return sum1;
        }
        static float principalInterest(float a, float b, float c)
        {
            float sum2 = a + (a * (b / 100) * (c / 360));
            return sum2;
        }
    }
}
```

- Debug.



```
 Original money
please input a number (more than 0)1000000
x_true VALUE:True
 Interest rate
please input a number (more than 0)50
x_true VALUE:True
 Number of days sent
please input a number (more than 0)360
x_true VALUE:True
 Selection:

 1-Total interest.
2-Total princial and interest.
3-Exit.
  Input selection:
1
 Total interest is 500000
 Selection:

 1-Total interest.
2-Total princial and interest.
3-Exit.
  Input selection:
2
 Total princial and interest is 1500000
 Selection:

 1-Total interest.
2-Total princial and interest.
3-Exit.
  Input selection:
```

Debug

2. Evaluate the relationship between the written algorithm and the code variant.

	Algorithm	Code
Definition	A well designed series od steps for solving a big problem.	Instructions for machine to execute.

- An algorithm is a series of steps for solving a problem, completing a task or performing a caculation. Algorithms are usually executed by computer programs but the term can also apply to steps in domains such as mathematics for human problem solving.
- Code is a series of steps that machines can execute. In many cases, code is composed in a high level language that is then automatically translated into instruction that machines understand.
- The term algorithm is traditionally used to indicate code that has a highly optimized design such as a well accepted solution to a complex problem in computer science. The term suggests code that has a rigorous design that solves a daunting problem.
- Code is offen low complexity, repetitive or non-critical. For example, code that displays a user interface, validates input, performs a transaction or calculates a value is usually straightforward to implement. Algorithms are at another level of comlexityand may begin life as a research project or similarly intensive effort. Any code that is composed by a developer on the fly that doesn’t solve a big problem isn’t typicall considered an algorithm.
- It should be noted that it is common for firms to use the term algorithm simply because it sounds good. As such, the term is beginning to lose its meaning and is becoming increasingly synonymous with code.

IV. Reference.

Definition of algorithm [online]
Available at: https://www.techopedia.com/definition/3739/algorithm
[Accessed 11 8 2019].

Evaluate the relationship between the written algorithm and the code variant [online]
Available at: https://simplicable.com/new/algorithm-vs-code
[Accessed 11 8 2019].