Cambridge International AS & A Level

Cambridge International Examinations
Cambridge International Advanced Subsidiary and Advanced Level

COMPUTER SCIENCE **9608/21**

Paper 2 Fundamental Problem-solving and Programming Skills **October/November 2015**

PRE-RELEASE MATERIAL

This material should be given to candidates on receipt by the Centre.

9753524949

READ THESE INSTRUCTIONS FIRST

Candidates should use this material in preparation for the examination. Candidates should attempt the practical programming tasks using their chosen high-level, procedural programming language.

DC (NH/CGW) 95378/4
© UCLES 2015

This material is intended to be read by teachers and candidates prior to the November 2015 examination for 9608 Paper 2.

Reminders

The syllabus states:

- there will be questions on the examination paper which do not relate to this pre-release material
- you must choose a high-level programming language from this list:
    - Visual Basic (console mode)
    - Python
    - Pascal / Delphi (console mode)

Questions on the examination paper may ask the candidate to write:

- structured English
- pseudocode
- program code

A program flowchart should be considered as an alternative to pseudocode for the documenting of an algorithm design.

Candidates should be confident with:

- the presentation of an algorithm using either a program flowchart or pseudocode
- the production of a program flowchart from given pseudocode or the reverse

Preliminary work

You should create a text file `BOOK-FILE` using a text editor.

Key focus: Text file

The file will consist of around 30 book titles with each book title on a new line.

TASK 1

Key focus: Arrays

A program is to be written to:

- read the data values from the text file `BOOK-FILE` into a 1D array `Book`
- output each book title from the array

TASK 1.1

Write the **pseudocode** for this program.

Key focus: Program code

TASK 1.2

Write **program code**, in your chosen language, for the pseudocode design produced in TASK 1.1.

Suggested extension task

Write **program code** to add a main menu to the program. The options are:

1 display the file contents
2 search the array for a particular book
3 end the program

Write **program code** for menu option 2 as follows:

The program will:

- prompt the user for entry of a book title
- input the book title
- carry out a linear search of the Book array to output either:
  - BOOK FOUND
  - BOOK NOT FOUND

TASK 2

TASK 2.1

A second program is to be written to search the file for a particular book.

The program will:

- prompt the user for entry of a book title
- input the book title
- carry out a linear search of the file to output either:
  - BOOK FOUND
  - BOOK NOT FOUND

Key focus: Linear search

The pseudocode for this program is given below.

The pseudocode assumes the exact number of books in the file is not known.

Key focus: Pseudocode

```
OPENFILE "BOOK-FILE" FOR READ
IsFound ← FALSE
OUTPUT "Enter book"
INPUT ThisBook

REPEAT
    READFILE "BOOK-FILE", FileBook
    IF FileBook = ThisBook
      THEN
        IsFound ← TRUE
        OUTPUT "BOOK FOUND"
    ENDIF
UNTIL IsFound = TRUE OR EOF("BOOK-FILE")

IF IsFound = FALSE
  THEN
    OUTPUT "BOOK NOT FOUND"
ENDIF
```

Write **program code** from this pseudocode design.

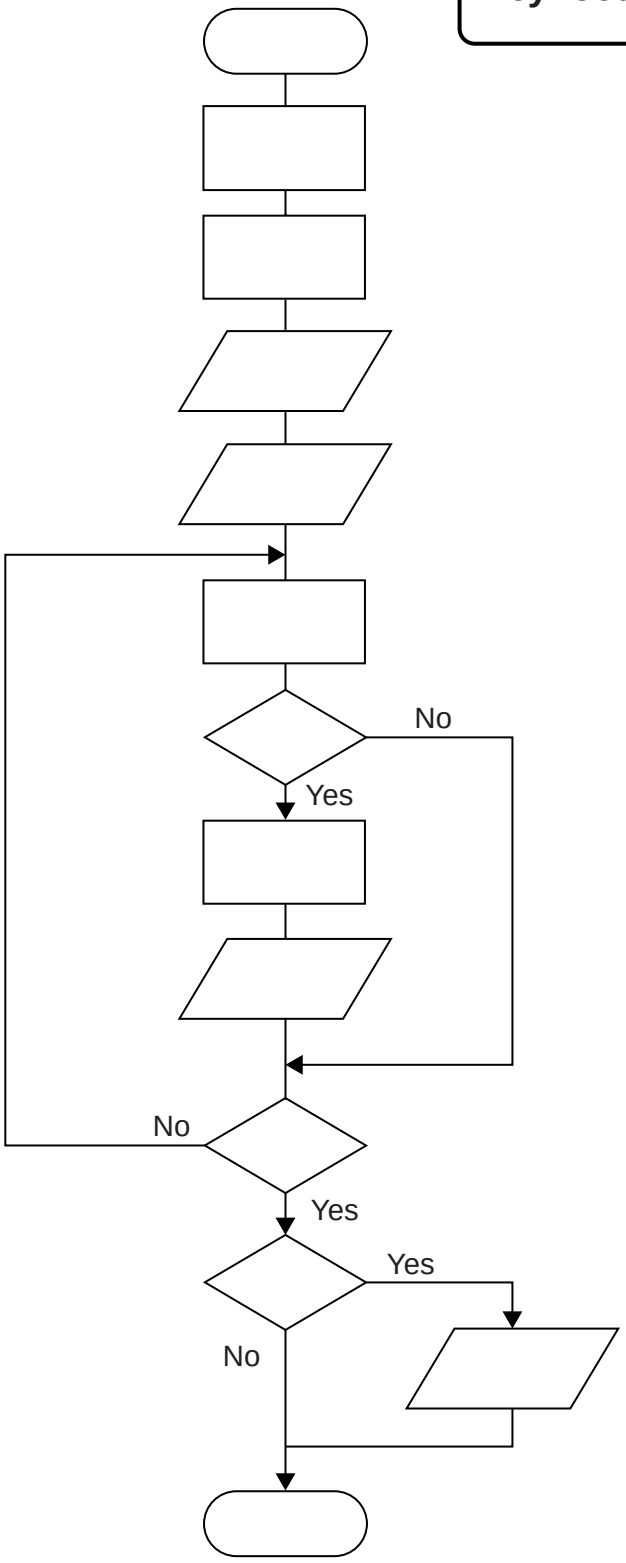
TASK 2.2

The program design for TASK 2.1 could have been given as a program flowchart.

Complete the following **program flowchart** from the given pseudocode, by:

- labelling the boxes
- drawing arrows on the lines

Key focus: Program flowchart



© UCLES 2015

& Operator

The & operator will be used to concatenate two strings.

For example: `"Birthday" & "Party"`
Evaluates to: `"Birthday Party"`

Built-in Functions

Any high-level programming language will have many built-in functions for the programmer to use.

It is appreciated that the three programming languages often implement these functions with very different syntax. Candidates should be familiar with the syntax used in their chosen programming language.

If a built-in function is to be used in pseudocode on the examination paper, the function will be shown and explained. Examples of this follow.

Key focus: Built-in function definitions

String handling functions (Pseudocode)

`ONECHAR(ThisString : STRING, Position : INTEGER) RETURNS CHAR`
returns the character at index position `Position` (counting from the start of the string with value 1) from the string `ThisString`.

For example: `ONECHAR("Hockey", 4)` returns `'k'`

`CHARACTERCOUNT(ThisString : STRING) RETURNS INTEGER`
returns the number of characters in string `ThisString`.

For example: `CHARACTERCOUNT("Real Madrid")` returns 11

`SUBSTR(ThisString : STRING, Value1 : INTEGER, Value2 : INTEGER) RETURNS STRING`
returns a sub-string from within `ThisString`.
`Value1` is the start index position (counting from the left, starting with 1).
`Value2` is the final index position.

For example: `SUBSTR("art nouveau", 5, 11)` returns `"nouveau"`

Conversion between data types (Pseudocode)

```
TONUM(ThisDigit : STRING) RETURNS INTEGER
```
Returns the integer equivalent of the string `ThisDigit`
For example: `TONUM("8")` returns integer 8

```
TOSTRING(ThisNumber : INTEGER or REAL) RETURNS STRING
```
Returns characters which make up `ThisNumber` as a string
For example: `TOSTRING(83)` returns `"83"`

`TOSTRING(704.25)` returns `"704.25"`

Using ASCII character codes (Pseudocode)

```
CHR(ThisInteger : INTEGER) RETURNS CHAR
```
Returns the character with ASCII value `ThisInteger`
For example: `CHR(65)` returns character `'A'`

```
ASC(ThisCharacter : CHAR) RETURNS INTEGER
```
Returns the ASCII value for character `ThisCharacter`
For example: `ASC('A')` returns integer 65

Random Number Generator (Pseudocode)

```
RANDOM(Integer1 : INTEGER, Integer2 : INTEGER) RETURNS INTEGER
```
generates a random integer in the range from `Integer1` to `Integer2` inclusive.

For example: `RANDOM(10, 12)` returns either: 10, 11 or 12

TASK 3

TASK 3.1

Write a program with the following specification:

- the user enters two integers (X and Y)
- the difference between X and Y should be at least 20 and X < Y
- the program generates a sequence of 20 random numbers between X and Y
- the program outputs the sequence of random numbers

Suggested extension task

At present, the program might generate the same number more than once.

Modify the program design so that duplicates are not displayed.

© UCLES 2015

TASK 4

Use the functions given to evaluate the following expressions:

1. `ONECHAR("Tiger", 3)` .....

2. `CHARACTERCOUNT("Great Pyramid of Giza")` .....

3. `TONUM("3") + TONUM("29")` .....

4.
```
p ← "The" & "Titanic"
q ← SUBSTR(p, 8, 4)
```
q .....

5. `IDE ← "Integrated Development Environment"`

   Using one or both of the `SUBSTR` and `ONECHAR` functions, write expressions to return:

   (a) `"Integrated"` .....

   (b) `"Development"` .....

   (c) `"IDE"` .....

6. Use an ASCII code table to find the value for:

   (a) `(ASC('F') + 30) / 50` .....

   (b) `2 * (ASC(<Space>) + ASC('D'))` .....

7. Write the text string output by the following sequence of statements.

```
A ← "The answer is: "
B ← TOSTRING(42)
OUTPUT A & B
```
.....

© UCLES 2015

BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge International Examinations Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cie.org.uk after the live examination series.

Cambridge International Examinations is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of University of Cambridge Local Examinations Syndicate (UCLES), which is itself a department of the University of Cambridge.