Random File

Papers Dock

COMPUTER SCIENCE 9618 PAPER 3

File Handling

```
TYPE Book
    DECLARE ISBN : Integer
    DECLARE Title : String
    DECLARE Genre : String
ENDTYPE
```

What is a record ?

Types Of Files

Binary Files

- .dat is the extension
- data is stored in binary format directly
- can be accessed directly.

Text Files

- .txt is the extension
- accessed sequentially.

Random Files

Read/Write at the same tim

Data can be accessed directly

- Random files contains a collection of data
- Normally as records of fixed length
- They can be thought of as having a file pointer which can be moved to any location or address in a file
- The record at that location can then be read or written

Pseudocode For Random Files

OPENFILE <filename> FOR RANDOM ⇒ to open a random file

The SEEK command moves the file pointer to a given location
SEEK <filename>, <address> ⇒ goes to the given address

The command GETRECORD should be used to read the record at the file pointer
GETRECORD <filename>, <identifier> ⇒ get the particular record

When this command is executed, the variable is assigned to the record that is read

The command PUTRECORD is used to write a record into the file at the file pointer
PUTRECORD <filename>, <identifier> ⇒ put the particular record

Pseudocode Example

```
TYPE Book
    DECLARE BookID : INTEGER
    DECLARE BookName : STRING
    DECLARE Author : STRING
    DECLARE PublishedDate : DATE
ENDTYPE
```

Question : Make A Variable novel Declared as Book

```
DECLARE novel : Book

novel.BookID ← 235
novel.BookName ← " Topical Booklet "
novel.Author ← "Dr Taha Ali"
novel.PublishedDate ← 2002 / 01 / 21
```

There is a file "Papersdock.dat" — put the novel record in 48th location (address)

```
OPENFILE "Papersdock.dat" FOR RANDOM
SEEK "Papersdock.dat" , 48
PUTRECORD "Papersdock.dat" , novel
CLOSEFILE "Papersdock.dat"
```

There is a file "Papersdock.dat", read the record on 10th location and output the author name.

```
DECLARE Temp  : Book
OPENFILE  “Papersdock.dat”  FOR RANDOM
SEEK “Papersdock.dat” ,  10
GETRECORD  “Papersdock.dat” ,  Temp
CLOSEFILE   “Papersdock.dat”
OUTPUT Temp.AuthorName
```

PastPaper Question

6 (a) Write **pseudocode** statements to declare the composite data type, `TAppointments`, to hold data about patients for a dental clinic. It will include for each patient:

- name (first name and last name)
- date of birth
- telephone number
- date of last appointment
- date of next appointment
- all treatments are complete (yes or no).

```
TYPE TAppointments
    DECLARE Name : STRING
    DECLARE DateOfBirth : DATE
    DECLARE Telephone : STRING
    DECLARE LastAppointment : DATE
    DECLARE NextAppointment : DATE
    DECLAR TreatmentComplete : BOOLEAN
ENDTYPE
```

[4]

(b) This pseudocode algorithm reads dental records stored in a random file using the user-defined data type `TAppointments` and prints the contents of the file, one record at a time.

Complete this file handling pseudocode:

```
DECLARE DentalRecord : ARRAY[1:250] OF TAppointments
DECLARE DentalFile : STRING
DECLARE Count : INTEGER
DentalFile ← "DentalFile.dat"
OUTPUT "The file ", DentalFile, " contains these records:"

OPENFILE DentalFile FOR RANDOM

    Count ← 1

REPEAT
    SEEK DentalFile, Count
    GETRECORD DentalFile , DentalRecord[count]
    OUTPUT DentalRecord[Count]
    Count ← Count + 1

UNTL EOF (DentalFile)

CLOSEFILE DentalFile
```

[5]

(b) A pseudocode algorithm searches for a customer record in a random file `AccountRecord.dat`. A user inputs the name of the customer.

The records are stored using the user-defined data type `TAccount`.

```
TYPE TAccount
    DECLARE AccountNumber : INTEGER
    DECLARE Name : STRING
    DECLARE Address : STRING
    DECLARE Telephone : STRING
ENDTYPE
```

If the record is found, it is output, otherwise an error message is displayed.

Complete the file handling pseudocode.

```
DECLARE Customer : TAccount
DECLARE Location : INTEGER
DECLARE MaxSize : INTEGER
DECLARE FoundFlag : BOOLEAN
DECLARE SearchCustomer : STRING
MaxSize ← 1000

OPENFILE .....
Location ← 1

..... ← FALSE
OUTPUT "Enter the customer's name"

.....

..... AND Location <= MaxSize

    ..... "AccountRecord.dat", .....
    GETRECORD "AccountRecord.dat", Customer
    IF SearchCustomer = Customer.Name THEN
        OUTPUT "Customer found: "
        OUTPUT Customer            // output customer record
        FoundFlag ← TRUE
    ENDIF
    Location ← Location + 1
ENDWHILE
IF NOT FoundFlag THEN

    OUTPUT "....."
ENDIF
```

[7]

```
DECLARE Customer : TAccount
DECLARE Location : INTEGER
DECLARE MaxSize : INTEGER
DECLARE FoundFlag : BOOLEAN
DECLARE SearchCustomer : STRING
MaxSize ← 1000
OPENFILE "AccountRecord.dat" FOR RANDOM
Location ← 1
FoundFlag ← FALSE
OUTPUT "Enter the customer's name"
INPUT SearchCustomer
WHILE NOT FoundFlag AND Location <= MaxSize
    SEEK "AccountRecord.dat", Location
    GETRECORD "AccountRecord.dat", Customer
    IF SearchCustomer = Customer.Name THEN
        OUTPUT "Customer found: "
        OUTPUT Customer            // output customer record
        FoundFlag ← TRUE
    ENDIF
    Location ← Location + 1
ENDWHILE
IF NOT FoundFlag THEN
    OUTPUT "Customer does not exist."
ENDIF
```