Assignment 2 2023

DUE DATE:	**3 July 2023**
CUT-OFF DATE:	**6 July 2023**
TUTORIAL MATTER:	**Chapters 4, 5, 6, 8 and 9 of the Study Guide**
	Chapters 4 (section 4.6), 5 (section 5.5), 6, 8 and 9 (excluding the optional parts of section 9.2) of Savitch
WEIGHT:	**30%**
MARKS:	**70**

Question 1

Write a program to determine the tuition fees for a student. The program should use two overloaded functions, each named `calcFees`, to determine the tuition fees for a student. Students who repeat a module pay a different fee for the modules which are repeated. The program should first ask if the student repeats any modules. If the student repeats, the program should ask for the number of modules which are repeated.

One of the overloaded functions should accept the number of modules enrolled for the first time and the fee for those modules as arguments (parameters), while the other function accepts arguments for the number of modules enrolled for the first time and the fee for those modules as well as the number of modules repeated and the fee for those modules. Both functions should return the tuition fees for the student.

Question 2

Write a program that converts from 24-hour notation to 12-hour notation. For example, it should convert 14:25 to 2:25 PM. The input is given as two integers. Verifies that a legitimate 24-hour notation has been input by using the `assert` statement.

Question 3

Write a program for your local bank to prepare a statement for a customer's checking account at the end of each month. The data is stored in a `.dat` file in the following format:

```
46780976 3750.40
W 250.00
D 1200.00
W 75.00
W 375.00
D 1200.00
I 5.50
W 400.00
W 600.00
D 450.50
W 35.65
```

The first line of code shows the account number followed by the account balance. For each line of code there is a transaction code and the transaction amount. The transaction codes are as follows:

W = Withdrawal

D = Deposit

I = Interests

The program has to display the account statement on the console. It should update the balance after each transaction. During the month, if at any time the balance goes below R 1000, a R25 service fee is charged. The program should print the following information: account number, opening balance at the beginning of the month, each transaction as it occurs, service fees when charged, interest paid by the bank and closing balance at the end of the month. Banking costs (i.e. total of all service fees incurred) are deducted at the end of the month. An example of the output your program should produce for the input file above, is shown below. Note that a deposit is shown as a credit ('Ct') on the statement:

Sample output:

```
Account number: 46780976

Opening balance: R3750.40

Transaction     Amount          Balance         Bank costs
Withdrawal       250.00         4000.00
Deposit         1200.00Ct       2800.00
Withdrawal        75.00         2725.00
Withdrawal      1375.00         1350.00
Deposit         1200.00Ct       1550.00
Interest           5.50         1555.50
Withdrawal       400.00         1155.50
Withdrawal       600.00          555.50              25.00
Deposit          450.00Ct       1005.50
Withdrawal        35.65          969.85
Banking costs     25.00          969.60

Closing balance: R969.60
```

NB: First plan your program on paper (using your computational thinking to do so). **You have to submit your plan for your program as well as the actual program code, input and output.** Planning your program can take the form of a flowchart, pseudocode, or notes to guide you in the development of the program.

Question 4

You have to write a program to read an input file character by character to help Peter solve the following activity in his activity book. The paragraph below is given:

```
We h2pe that 32u e5723ed the acti4it3. A6ter 32u ha4e c2mpleted
the acti4it3, 0e5d 32ur re0ult t2: The Acti4it3 C2mpetiti25,
Betty Da4i0 0treet 99, Auckla5d Park, 8989, a5d 0ta5d a cha5ce
t2 wi5 a hamper c250i0ti51 26 c2l2uri51 a5d acti4it3 b22k0,
c2l2uri51 pe5cil0 a5d pe50.
```

Create an input file `activity.dat` with the paragraph above. The numbers 0 to 7 have to be replaced as follows:

`0` must be replaced by `s`
`1` must be replaced by `g`
`2` must be replaced by `o`
`3` must be replaced by `y`
`4` must be replaced by `v`
`5` must be replaced by `n`
`6` must be replaced by `f`
`7` must be replaced by `j`

Ask the user to input the names of the input and output files. Read the input file character by character, and write the character (if it stays the same) to the output file, or write the changed character to the output file. Call your output file `competition.txt`.
Allow the user to specify the names of the input and output files.

NB: First plan your program on paper (using your computational thinking to do so). **You have to submit your plan for your program as well as the actual program code, input and output.** Planning your program can take the form of a flowchart, pseudocode, or notes to guide you in the development of the program.

Question 5

Write a program that reads a person's name in the following format: first name, then middle name or initial, and then last name. The program then outputs the name in the following format:

last name, first name. middle initial.

For example, the input

```
Mary Average User
```

should produce the output

```
User, Mary A.
```

Your program should work the same and place a full stop after the middle initial even if the input did not contain a full stop. Your program should allow for users who give no middle name or initial. In that case, the output of courses contains no middle name or initial. For example, the input

```
Mary User
```

should produce the output

```
User, Mary
```

Your program should also accept names in lowercase, uppercase or a mix of lowercase and uppercase, and display that in the correct format, e.g. if the input is

```
mArY average USER
```

should produce the output

```
User, Mary A.
```

Use C-strings and assume that each name is at most 20 characters long.

Hint: it may be easier to use 3 C-strings.

Remember to plan your program!

Question 6

Given the following header:

```
vector<string> split(string target, string delimiter);
```

implement the function `split()` so that it returns a vector of the strings in `target` that are separated by the string `delimiter`. For example,

`split("do,re,me,fa,so,la,ti,do", ",")` should return a vector with the strings `"do"`, `"re"`, `"me"`, `"fa"`, `"so"`, `"la"`, `"ti"` and `"do"`.

Test your function `split()` in a driver program that displays the strings in the vector after the `target` has been split.

Question 7

(a) What is a pointer?

(b) What is a dereferencing operator?

(c) What is the difference between assignment statements `p1 = p2;` and `*p1 = *p2;`

(d) What is a dynamic variable?

(e) What is the purpose of the `new` operator?

(f) What is the purpose of the `delete` operator?

(g) What is the freestore (also called the heap)?

(h) What is the difference between dynamic variables and automatic variables?

(i) What is a dynamic array?

(j) What is the advantage of using dynamic arrays?

(k) What is the relationship between pointers and arrays?

(l) Write statements to do the following:

i. Define a pointer type `int_ptr` for pointer variables that contain pointers to `int` variables.

ii. Declare `p1` to be a pointer to an `int`.

iii. Dynamically allocate an integer variable and store its address in `p1`.

iv. Assign the value 23 to the variable that `p1` is pointing to.

v. Declare an `int` variable `a`.

vi. Let `p1` point to `a`.

vii. Free the memory allocated to the variable that `p1` is pointing to.

(m) Write statements to do the following:

i. Define a pointer type `int_ptr` for pointer variables that contain pointers to `int` variables.

ii. Declare `p2` to be a pointer to an `int`.

iii. Obtain an integer value `nrElements` from the user indicating the number of elements to allocate.

iv. Dynamically allocate an array of `nrElements` integers and store its address in `p2`.

v. Declare an `int` array `a` with 500 elements.

vi. Assume `p2` has been initialized and copy the elements of `p2` one by one to the corresponding elements in `a`.

vii. Free the memory allocated to the variable that `p2` is pointing to.

(n) Write a program that asks a user to enter the size of a dynamic array that stores exam marks obtained by students. Create the dynamic array and a loop that allows the user to enter an exam mark into each array element. Loop through the array, find the average mark for the exam and output it. Delete the memory allocated to your dynamic array before exiting your program.

©
UNISA
2023