

University of Nottingham Malaysia

FOUNDATION IN SCIENCE

FUNDAMENTALS OF PROGRAMMING (INDIVIDUAL ASSIGNMENT)

Name: Yeo Yi Xin

University ID: 20414841

Lecturer: TS. DR. THAMIL VAANI

TABLE OF CONTENTS

No.	Topic	Pages
1.	Algorithm	3 – 5
2.	Pseudocode	6 – 8
3.	Flowchart	9 – 13
4.	Testing Output	14
5.	Marking Rubric	15

A. Algorithm

STEP 1: START.

STEP 2: Initialise hour, minute and seconds with 0.

STEP 3(a): WHILE (1)

 ,1 means when all the condition is met.

3(b): Clear system output.

3(c): Increase second and check if it is equal to 60 then increase minute and reset seconds to 0.

3(d): Increase minute and check if it is equal to 60 then increase hour and reset minute to 0.

3(e): Increase hour and check if it is equal to 24 then reset hour to 0.

3(f): IF (hour<12),

 THEN FUNCTION CALL decoration() and move to STEP 4 (a).

 Print hour:minute:seconds "AM".

3(g): ELSE

 THEN FUNCTION CALL decoration() and move to STEP 4 (a).

 Print hour:minute:seconds "PM".

 ENDIF

3(h): FUNCTION CALL calendar() and move to STEP 5(a).

3(i): FUNCTION CALL decoration1() and move to STEP (a).

3(j): FUNCTION CALL Sleep(1000).

 Delay the system by 1000 milliseconds

END WHILE

STEP 4(a): Declare FUNCTION DEFINITION decoration with datatype, void.

4(b): Print heading line.

STEP 5(a): Declare FUNCTION DEFINITION calendar with datatype, void.

- 5(b): Initialise variable name t with data type time_t which represents clock time as an integer.
- 5(c): Initialise variable t with time(NULL), which is implementation-defined null pointer constant, to get current time.
- 5(d): Prompt structure pointer with variable tm tdy (stands for today) with localtime function. The localtime function converts the calendar time pointed by t into a broken-down time, expressed as a local time. The struct tm allows the program to describe calendar time. The localtime(&t) function returns the current time to t under variable tdy.
- 5(e): Print escape sequence “\t”.
- 5(f): Initialise w with tdy.tm_wday+1, which gives the current day of the week.
- 5(g)(a): SWITCH (w)
- 5(g)(b): Case 1: Print “Sunday”.
- 5(g)(c): Case 2: Print “Monday”.
- 5(g)(d): Case 3: Print “Tuesday”.
- 5(g)(e): Case 4: Print “Wednesday”.
- 5(g)(g): Case 5: Print “Thursday”.
- 5(g)(h): Case 6: Print “Friday”.
- 5(g)(i): Case 7: Print “Saturday”.
- 5(g)(j): DEFAULT: Print “Wrong week of the day!”.

END SWITCH

5(h): Display current day of the month using `tdy.tm_mday`, one of the components in structure, in datatype integer.

5(i): Initialise `m` with `tdy.tm_mon+1`, one of the components in structure, to get the current month.

5(i)(a): SWITCH (`m`)

5(i)(b): Case 1: Print "January, ".

5(i)(c): Case 2: Print "February, ".

5(i)(d): Case 3: Print "March, ".

5(i)(e): Case 4: Print "April, ".

5(i)(f): Case 5: Print "May, ".

5(i)(g): Case 6: Print "June, ".

5(i)(h): Case 7: Print "July, ".

5(i)(i): Case 8: Print "August, ".

5(i)(j): Case 9: Print "September, ".

5(i)(k): Case 10: Print "October, ".

5(i)(l): Case 11: Print "November, ".

5(i)(m): Case 12: Print "December, ".

5(i)(n): DEFAULT: Print "Wrong month!".

END SWITCH

5(j): Display current year using `tdy.tm_year+1900` one of the components in structure, in datatype integer.

STEP 6(a): Declare FUNCTION DEFINITION `decoration1` with datatype, void.

6(b): Print heading line.

STEP 7: END.

B. Pseudocode

PROGRAM DigitalClock:

Start;

Initialize hour, minute, seconds with 0;

WHILE (1), 1 means that all the condition is met.

 Clear system output;

 seconds++;

 IF (seconds==60)

 THEN minute+=1;

 seconds=0;

 ENDIF;

 IF (minute==60)

 THEN hour+=1;

 minute=0;

 ENDIF;

 IF (hour==24)

 THEN hour=0;

 ENDIF;

 IF (hour<12)

 THEN FUNCTION CALL decoration();

 Print heading line;

 Print hour:minute:seconds "AM";

 ELSE FUNCTION CALL decoration();

 Print heading line;

 Print hour:minute:seconds "PM";

 ENDIF;

```

FUNCTION CALL calendar();

    time_t t;

    Set t = time(NULL);

    Prompt struct tm tdy = *localtime(&t);

    w = tdy.tm_wday+1;

    Print escape sequence "\t";

    SWITCH (w)

        Case 1: Print "Sunday";

        Case 2: Print "Monday";

        Case 3: Print "Tuesday";

        Case 4: Print "Wednesday";

        Case 5: Print "Thursday";

        Case 6: Print "Friday";

        Case 7: Print "Saturday";

        DEFAULT: Print "Wrong week of the day!";

    END SWITCH;

    Print tdy.tm_mday in integer datatype;

    m = tdy.tm_mon+1;

    SWITCH (m)

        Case 1: Print "January, ";

        Case 2: Print "February, ";

        Case 3: Print "March, ";

        Case 4: Print "April, ";

        Case 5: Print "May, ";

```

Case 6: Print "June, ";

Case 7: Print "July, ";

Case 8: Print "August, ";

Case 9: Print "September, ";

Case 10: Print "October, ";

Case 11: Print "November, ";

Case 12: Print "December, ";

DEFAULT: Print "Wrong month!";

END SWITCH;

Print tdy.tm_year+1900;

FUNCTION CALL decoration1();

Print heading line;

FUNCTION CALL Sleep(1000);

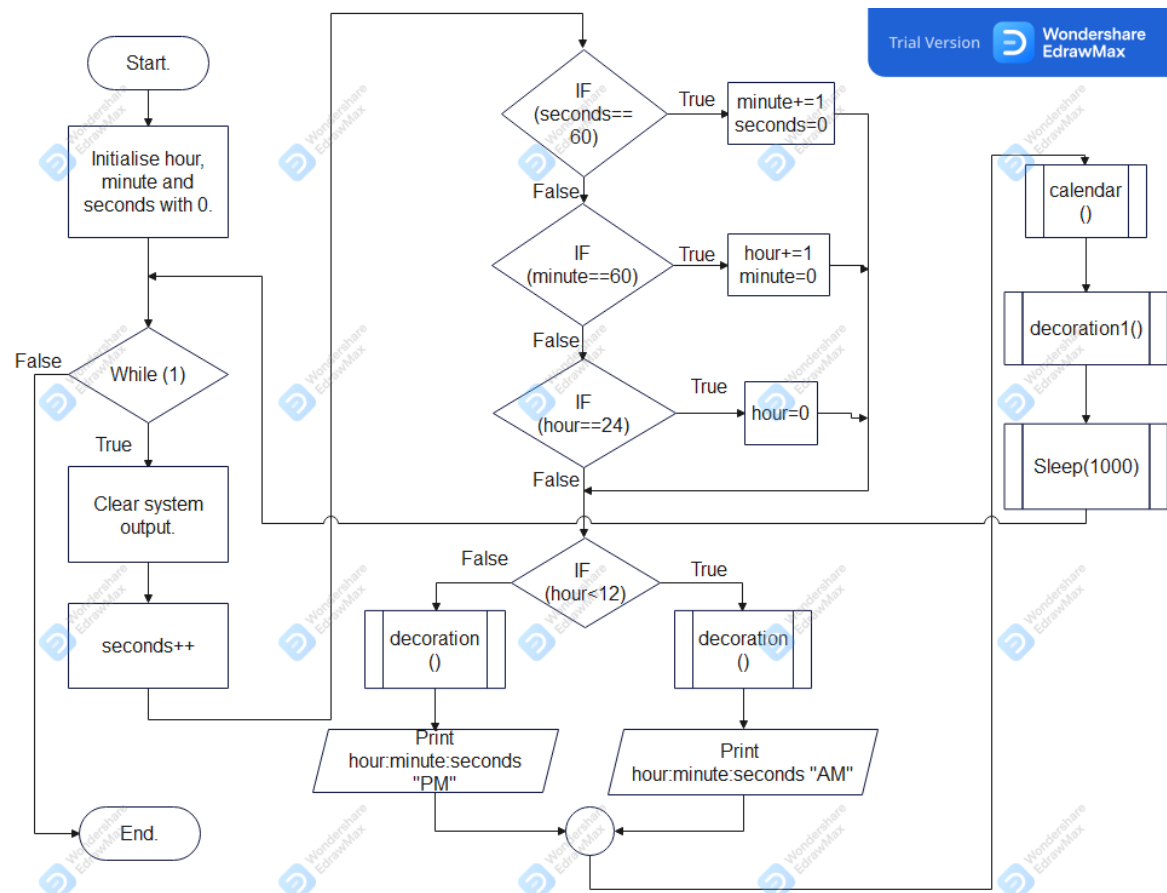
Delay system output by 1000 milliseconds;

END WHILE;

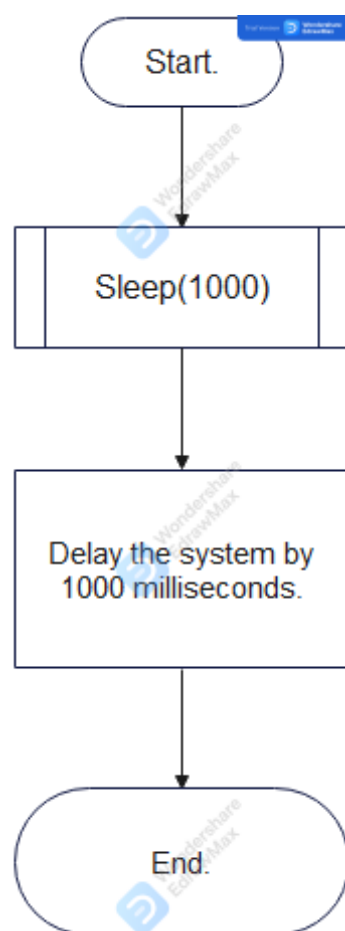
END.

C. Flowchart

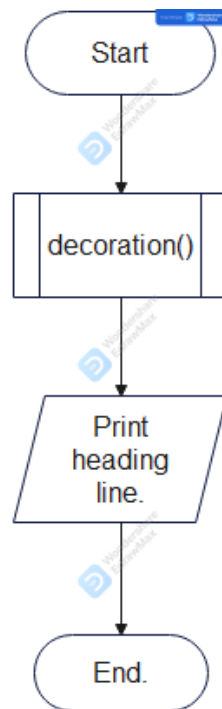
Generic Flowchart



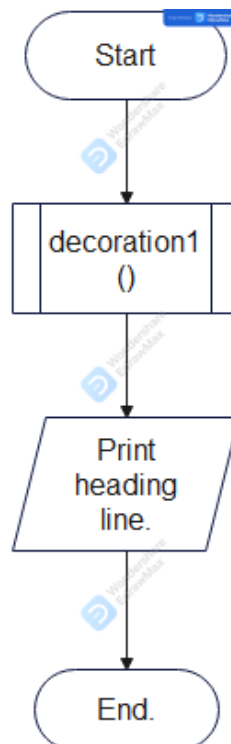
Specific Flowchart for Sleep() Function



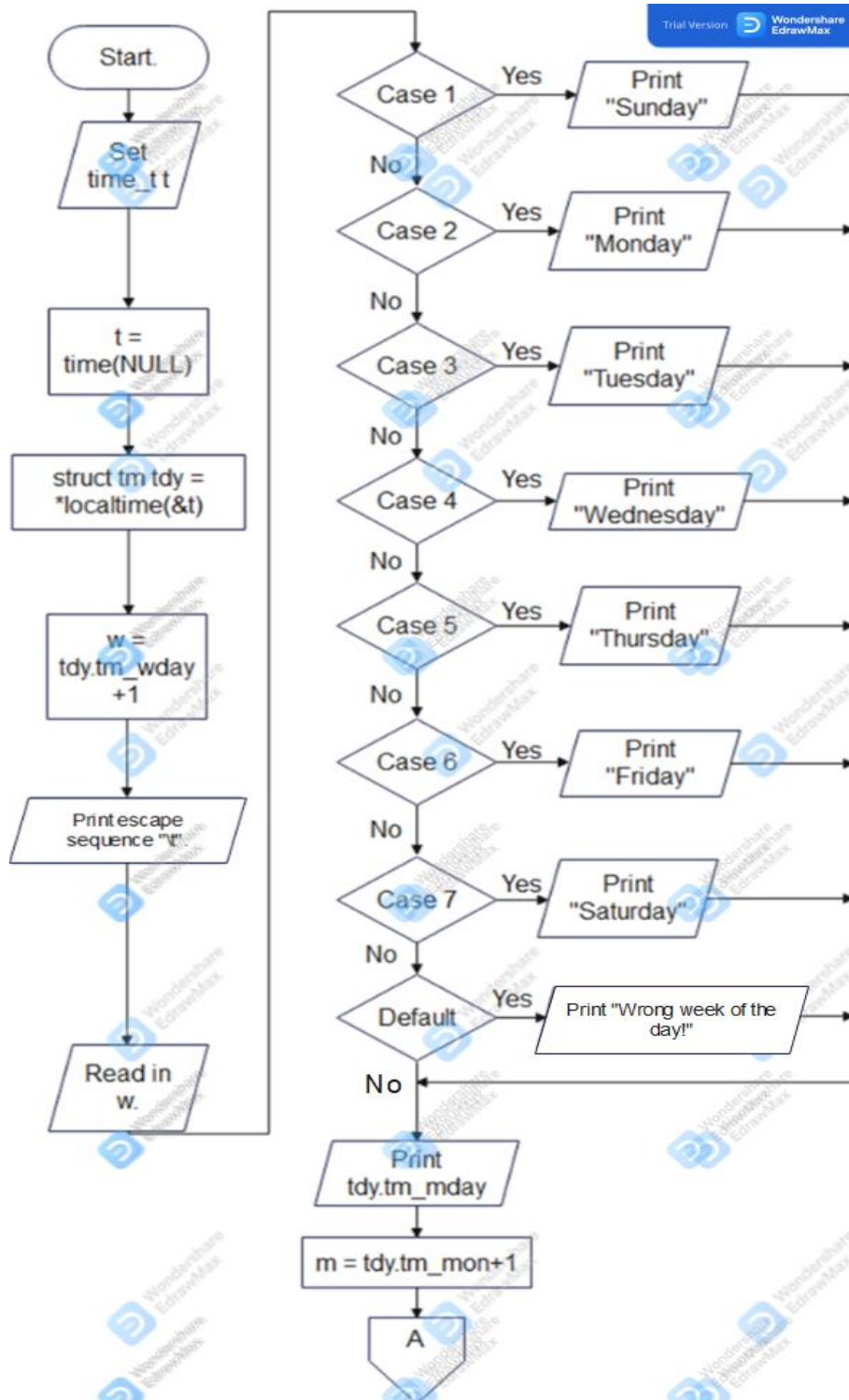
Specific Flowchart for decoration() Function

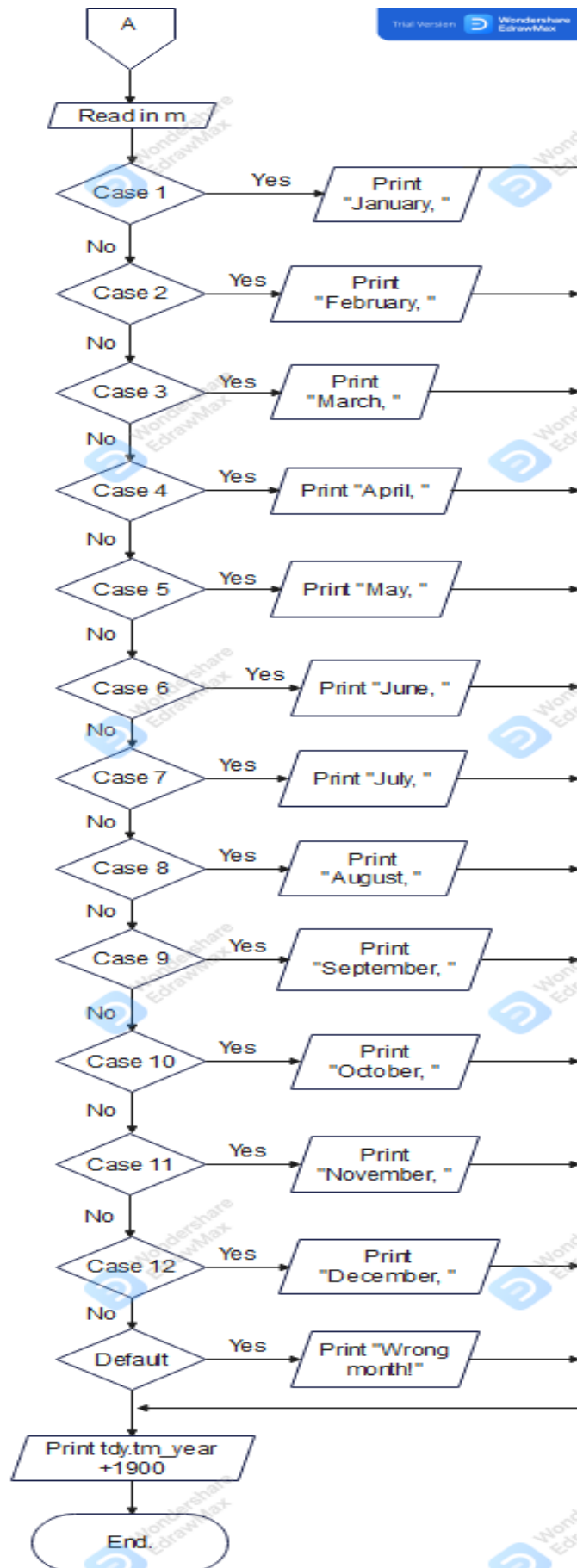


Specific Flowchart for decoration1() Function

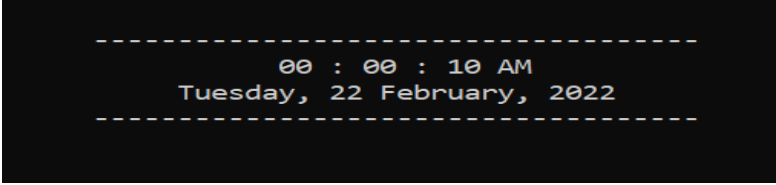
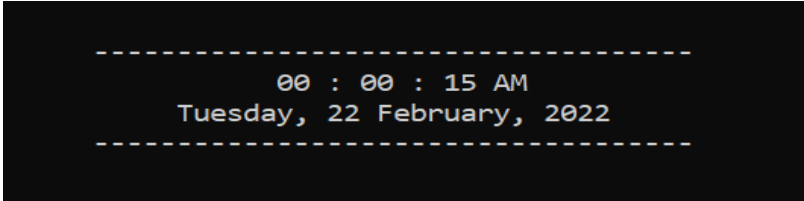
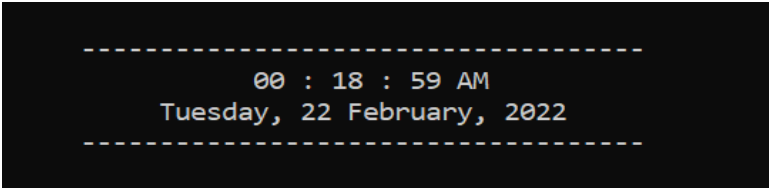
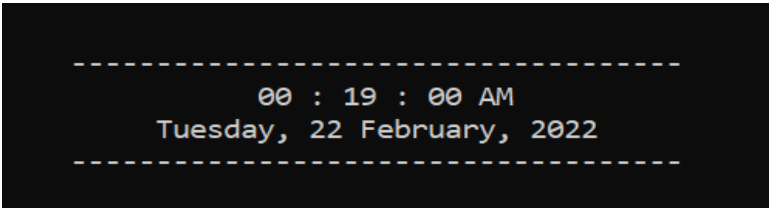
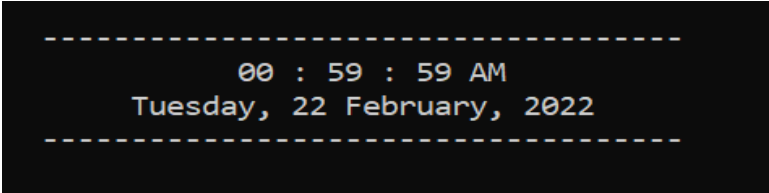
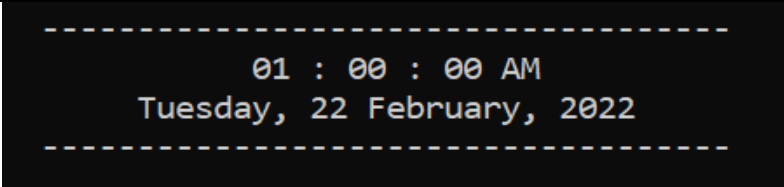


Specific Flowchart for calendar() Function





Testing output

NO	Test data/ test condition	Expected output	Actual output
1	Seconds	From 00:00:10 AM to 00:00:15 AM.	
			
2	Minutes	From 00:18:59 AM to 00:19:00 AM.	
			
3	Hours	From 00:59:59 AM to 1:00:00 AM.	
			

Marking Rubric

	CRITERIA	EXCELLENT	GOOD	SATISFACTORY	UNSATISFACTORY	TOTAL
		10	8-9	6-7	5	
Algorithm		Complete with start/stop of the programme	Less than 2 mistakes of steps	More than 2 mistakes of steps	Not complete	
Pseudocode	Start/stop of the program	Complete with start/stop of the programme		No start/stop of program and no capital letter (keywords) and no indentation.		
	Appropriate capital letter (keywords)	Complete use of appropriate capital letter (keywords) throughout the programme	Use of appropriate capital letter (keywords) in majority part of the programme	Use of few capital letter (keywords) within the programme	Not using capital letter (keywords) in a program	
	Appropriate indentation	Appropriate indentation throughout programme	Appropriate Indentation in majority part of the programme	Few appropriate Indentation in the programme	No indentation in the programme	
Flowchart	Symbol	Correct symbols throughout the chart	Less than 2 mistakes of symbols	More than 2 mistakes of symbols	Inappropriate symbol	
	Logic on control structure in the chart	Correct logic presented in the chart	Less than 2 mistakes of Logic in the chart	More than 2 mistakes of logic in the chart	Inappropriate logic presented in the chart	
Program code	Implementation of code	Accurate and extra ordinary output	Accurate and can execute	Able to run by meeting basic requirements	Inaccurate or cannot produce output	
Output	Accuracy of output based on range of test data	Accurate and Covering all the possible pathway	Accurate and covering majority of pathway	Accurate and covering average number of pathway based on your program	Acceptable output and covering any random pathway only	
Feedback TOTAL						/80