THE PAPUA NEW GUINEA UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMMUNICATIONS ENGINEERING

FINAL EXAMINATION (2021)
SEMESTER 1

EE212 INTRODUCTION TO PROGRAMMING IN C

SECOND YEAR (ELECTRICAL) BEEL 2

TIME ALLOWED: 3 HOURS

INFORMATION FOR STUDENTS

1. You have **TEN (10) MINUTES** to read the paper.
   You must not begin writing during this time.

2. **Answer all questions.**

3. All answers must be written in the **ANSWER BOOK** supplied.

4. **COMPLETE THE DETAILS REQUIRED ON THE FRONT COVER OF YOUR ANSWER BOOK – DO THIS NOW.**

5. Textbooks and laptops **ARE** permitted.

6. If you are found cheating in the Examination, the penalties specified by the University shall apply.

7. TURN OFF all Mobile Phone and place them on the floor under your seat before the start of Examination

QUESTION ONE Basic Statements and Expression [10 Marks]

Assuming that **A** and **B** are global variables such that **A is an array that holds 4 integers** and **B** is an integer pointer.

A) [3 mark] Write down the declaration of these variables such that A is initialised to 1,2,3, and 4

B) [2 mark] Write down the statement that points **B** to the 3rd element of **A.**

C) [1 mark] After that, If the statement `*B=0;` was executed, what shall be the output for `printf("%d",*(B-1));`

D) [3 mark] The list of statement(s) that will print all the current value of **A** on the same line such that each integer is separated by a space.

E) [1 mark] What the expected output for question D?

QUESTION TWO: Loops and Recursion [10 mark]

The mathematical series for the approximation of π (i.e. 3.1415...) is defined as:

$$\pi = 4\sum_{n=0}^{k}\frac{(-1)^n}{2n+1}$$

Implement the following function:

```
double pi_function(int k)
{
    // implement your code here:
}
```

A) [5 marks] Using loop statement(s).

B) [5 marks] Using recursion only.

QUESTION THREE String processing [10 Marks]

The following function prints multiple lines about alphabet symbol-counts occurred in a string:

```
void char_stats(char* input);
```

Assuming `input` points to a string containing only spaces and lower case words, implement the functionality using your own idea or via the following steps:

1. [3 marks] Declare an integer array variable (say `alpha_count`) of 26 elements, such that each element tracks the tally of a lower case character.
2. [4 marks] Go through each character of the string from `input` and update the tally of a character corresponding to an `alpha_count` element accordingly.
3. [3 marks] Go through each element of `alpha_count` and print the count value only of those with count of 1 or greater.

For example, the following code:

```
int main()
{
    char_stats("hello hae");
}
```

Will output to console:

```
a occur(s) 1 time(s)
e occur(s) 2 time(s)
h occur(s) 2 time(s)
l occur(s) 2 time(s)
o occur(s) 1 time(s)
```

QUESTION FOUR Line Graphics [10 Marks]

The following function draws a timer (using the `DrawLineEx` function from raylib)

```
void DrawTimer(Vector2 pos, double secs);
```

For example, if the following statement where executed (set timer to 130 seconds at screen position: [120, 120]):

```
Vector2 pos = {120, 120};
DrawTimer(pos, 130.0);
```

... then the following timer will be drawn, where the center of the clock is located at screen coordinate (120, 120) (i.e. the value of `pos`). Note: the long handle represents "seconds" whilst the other for "minutes".

Assuming `secs` is always between 0 to 3600 seconds, implement the function according to the following specification:

- Within a radius of 100 pixels, draw 60 evenly spaced marker-lines on the timer's edge, such that:
  - [2 mark] Every 15th line is 25 pixels long and 3 pixels in thickness
  - [2 mark] Every 5th line is 12.5 pixels long and 2 pixels in thickness
  - [2 mark] Every other line is 6.25 pixels long and 1 pixel in thickness
- [2 marks] Draw both handles pointing correctly at the mark-line according to the `sec` parameter variable such that:
  - [1 marks] long handle (seconds) 70 pixels long and 4 pixels in thickness
  - [1 marks] short handle (minutes) 50 pixels long and 8 pixels in thickness.

You may use the following function to generate a screen coordinate from a polar coordinate:

```
Vector2 from_polar(double radius, double theta)
{
    return (Vector2){cos(theta) * radius, -sin(theta) * radius};
}
```

Note that the following raylib function:

```
DrawLineEx(Vector2 startPos, Vector2 endPos, float thick, Color color);
```

Draws a line with a thickness set by the `thick` parameter. E.g. if it is set to 3, then the line will be 3 pixels thick.

QUESTION FIVE Animation [10 Marks]

The following is a partially completed code for animating the timer function implemented in Question 4.

```
typedef struct timerstate_s
{
    double start, current;
} timerstate_t;

timerstate_t gTimer;

void setup_animation(){ /*Part [A] code here*/ }
void timer_ani(animation_t *ani){ /*Part [B] code here*/ }

int main(void)
{
    InitWindow(800, 450, "Timer");
    SetTargetFPS(60);
    animation_init();
    setup_animation();
    while (!WindowShouldClose())
    {
        BeginDrawing();
            ClearBackground(WHITE);
            DrawTimer((Vector2){400,275}, gTimer.current);
        EndDrawing();
        animation_update();
    }
    CloseWindow();
    return 0;
}
```

Assume, that `DrawTimer` was implemented correctly in question 4. Complete the code for animating the timer arms smoothly via the `animation.h` library by:

A) [5 marks] implement `void setup_animation()`, which starts and schedule the `timer_ani` animation function to play for 1 second duration.

B) [5 marks] Implement `timer_ani` to animate `gTimer.current` field smoothly and repeat itself for the next 1 second each time it ends. Note: you may use `gTimer.start` to track the current second being animated.

The following are functions available from the `animation.c` library.

```
enum animation_state_e {ANIME_END,ANIME_START,ANIME_PLAYING};

void animation_init();
animation_t *animation_allocate();
void animation_schedule(animation_t *ani, double duration_seconds, double delay_seconds);
double animation_get_value(animation_t *);
void *animation_get_user_data(animation_t *, unsigned int idx);
void animation_set_user_data(animation_t *, unsigned int idx, void *);
void animation_set_event_listener(animation_t *ani, enum animation_state_e state,
animation_evnt_func_t listener);
void animation_update();
```