

Criteria B: Design

User Perspective:	1
A/ General Overview of the Command Line Interface (CLI)	1
B/ Function and Design of Each Part of the GUI	2
C/ Program Flow from User Perspective	3
Developer Perspective:	4
A/ Command Line Interface	4
B/ Main Program	5
i/ Page structure: Question Paper	5
ii/ Page structure: Mark Scheme	6
iii/ Program Overview	7
iv/ Program Files	9
v/ Program Functions	11
vi/ Program Variables	12
vii/ Detailed Flowchart of each part of program	14
a/ initiate_program	14
b/ make_folder	16
c/ convert_pdf	17
d/ crop	18
e/ merge	20
f/ find_qs	21
g/ merge_qs	22
h/ crop_final	23
i/ delete_folder	24
j/ mark_scheme	25
viii/ External Libraries Needed	26
ix/ Developer Test Plan	27

A/ General Overview of the Command Line Interface (CLI)

Please choose one of the below input file structure formats: 1) single file 2) multiple file Enter number here: 1
Please type the path of the past paper file you would like to parse: foo
Please choose one of the below output formats: 1) jpg 2) png Enter number here: 2
Great! Please wait while we work through <foo> Processing ██████████ 100%
Done! Please check <foo> again, there should be a new folder containing all the cropped and sorted individual <png> images of questions and accompanying mark scheme!

1

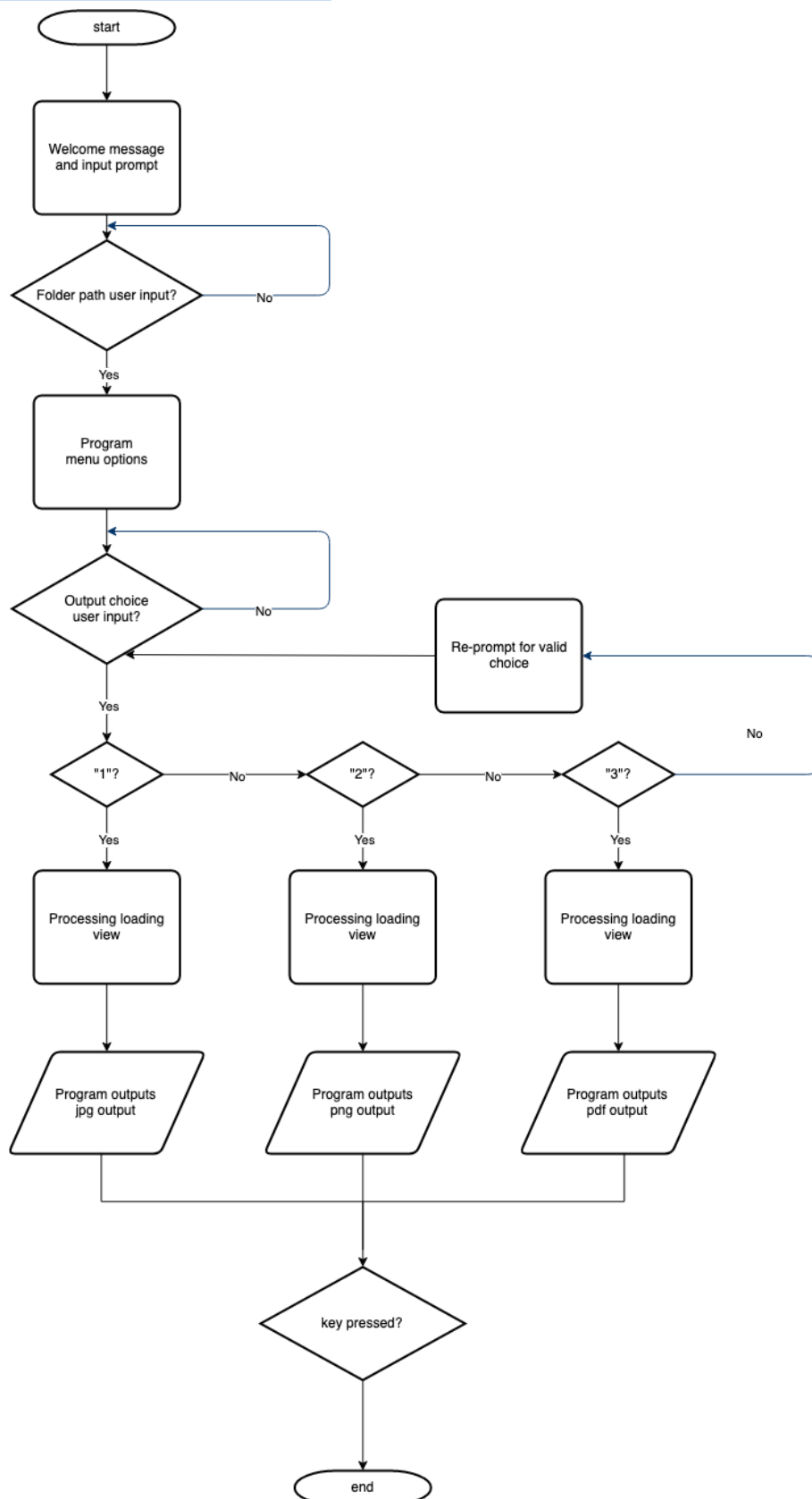
B/ Function and Design of Each Part of the GUI

No	Content	Description
(1)	Input file structure selection	Lists available input file structure formats, then prompts user to choose one from the 'menu list'
(2)	Input selection	Prompts user for input path
(3)	Output format selection	Lists available output formats, then prompts user to choose one from the 'menu list'
(4)	Progress indicator	Shows a graphical representation of the processing progress
(5)	Output result	General description and location of output, then prompts user to terminate program

Corresponding parameters:

No	Input type	Purpose	Possible values
(1)	int	Indicates user choice of input folder structure type	'1', '2', '3'
(2)	str	Indicates pathway for program to read file	Mac/Linux: /{group}/{user}/{directory}/{file_name} Windows: {volume}:\\{directory}\\{file_name}
(3)	int	Indicates user choice of output format	'1', '2', '3'
(5)	key	Terminates window	Any key

C/ Program Flow from User Perspective



Developer Perspective:

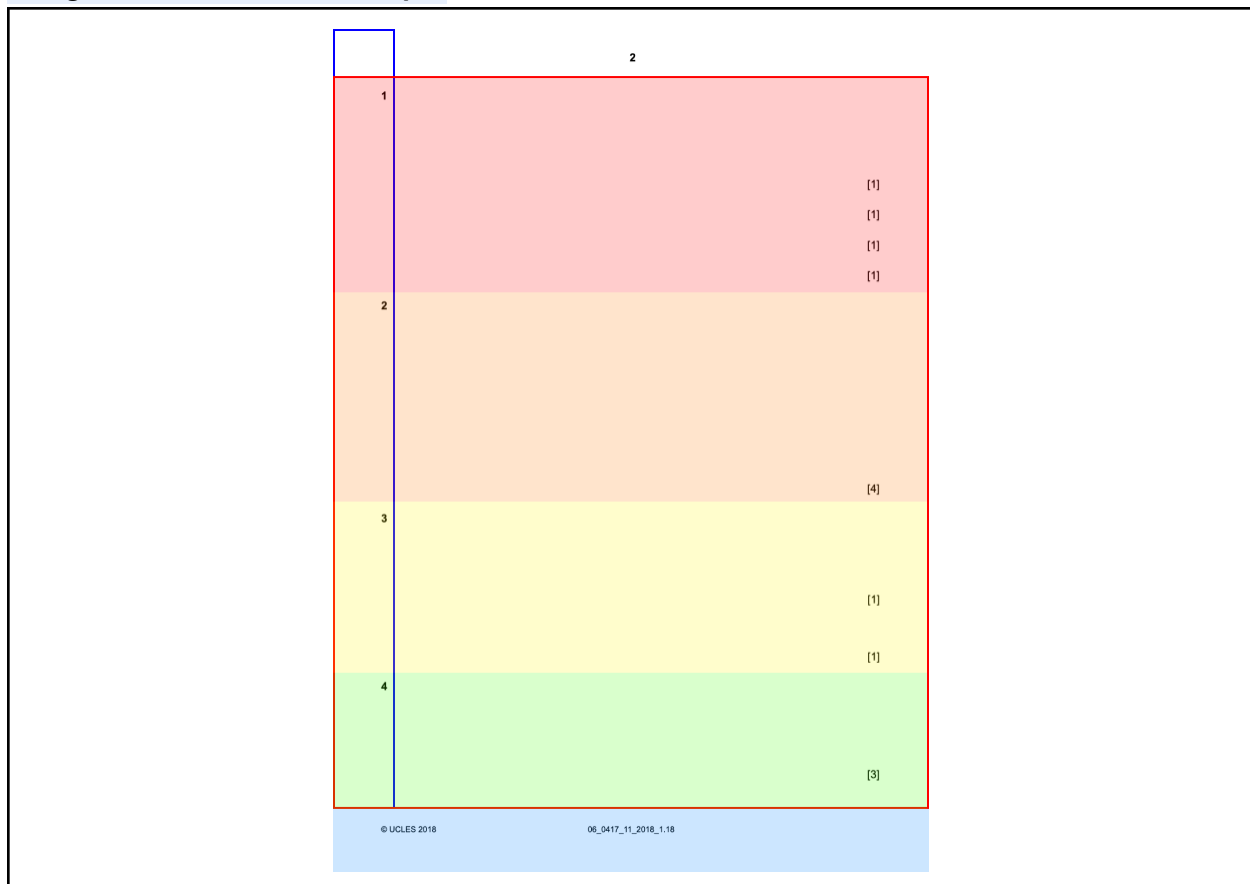
A/ Command Line Interface

Libraries I plan to use for the CLI are:

Library Name	Functionality								
Progress	Shows a progress bar for processing period								
EXAMPLE:									
Processing ██████████ 100% [00:02/00:00]									
Fig 2. Example progress bar									
Rich	Allows rich text formatting for CLI in order to make it more user friendly and to segment different sections of the CLI								
EXAMPLE:									
Please choose one of the below output formats: output choices									
<table border="1"><thead><tr><th>Number</th><th>Output format</th></tr></thead><tbody><tr><td>1</td><td>jpg</td></tr><tr><td>2</td><td>png</td></tr><tr><td>3</td><td>pdf</td></tr></tbody></table>		Number	Output format	1	jpg	2	png	3	pdf
Number	Output format								
1	jpg								
2	png								
3	pdf								
Enter Number here: 3									
Great! Please wait while we work through /{group}/{user}/{dir}/{file}									
Fig 3. Example Rich text									

B/ Main Program

i/ Page structure: Question Paper



Colour Code	Section Description
Blue outline	Left margin of the question paper. This margin is consistent throughout all question papers post-2016, which makes it useful for identifying when questions start.
Red outline	Main body of the question paper. This is cropped into a new image first in order to get rid of unnecessary information in the final version (ie. page number, footer)
Red fill	First question with 4 sub-questions
Orange fill	Second question
Yellow fill	Third question with 2 sub-questions
Green fill	Fourth question
Blue fill	Footer contains the paper code of the paper. This gives information on Year, session, course code and paper number.

ii/ Page structure: Mark Scheme

0417/11

Cambridge IGCSE – Mark Scheme
PUBLISHED

May/June 2017

Question	Answer	Marks
1(a)		1
1(b)		1
1(c)		1
1(d)		1

Question	Answer	Marks
2		2

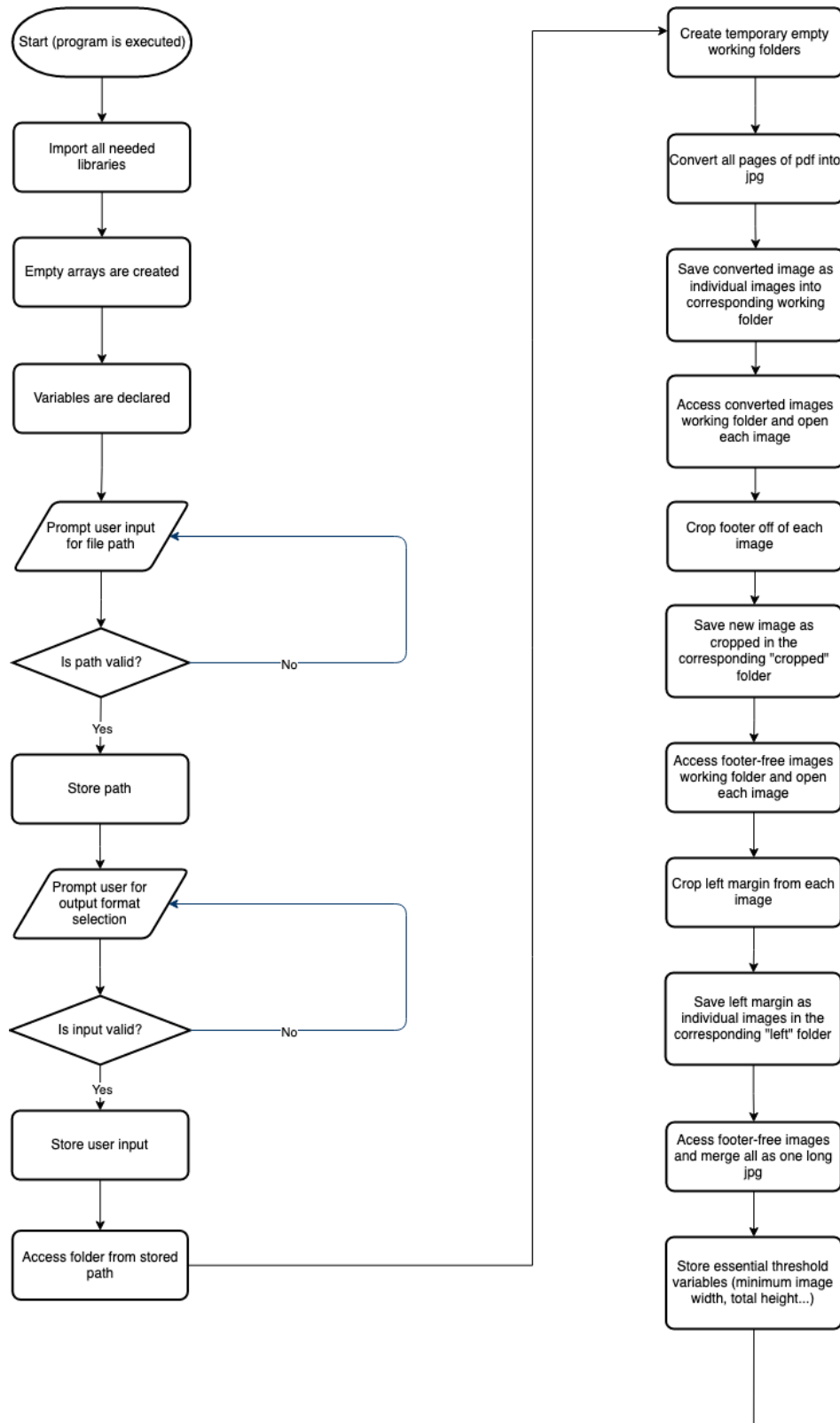
Question	Answer	Marks
3		2

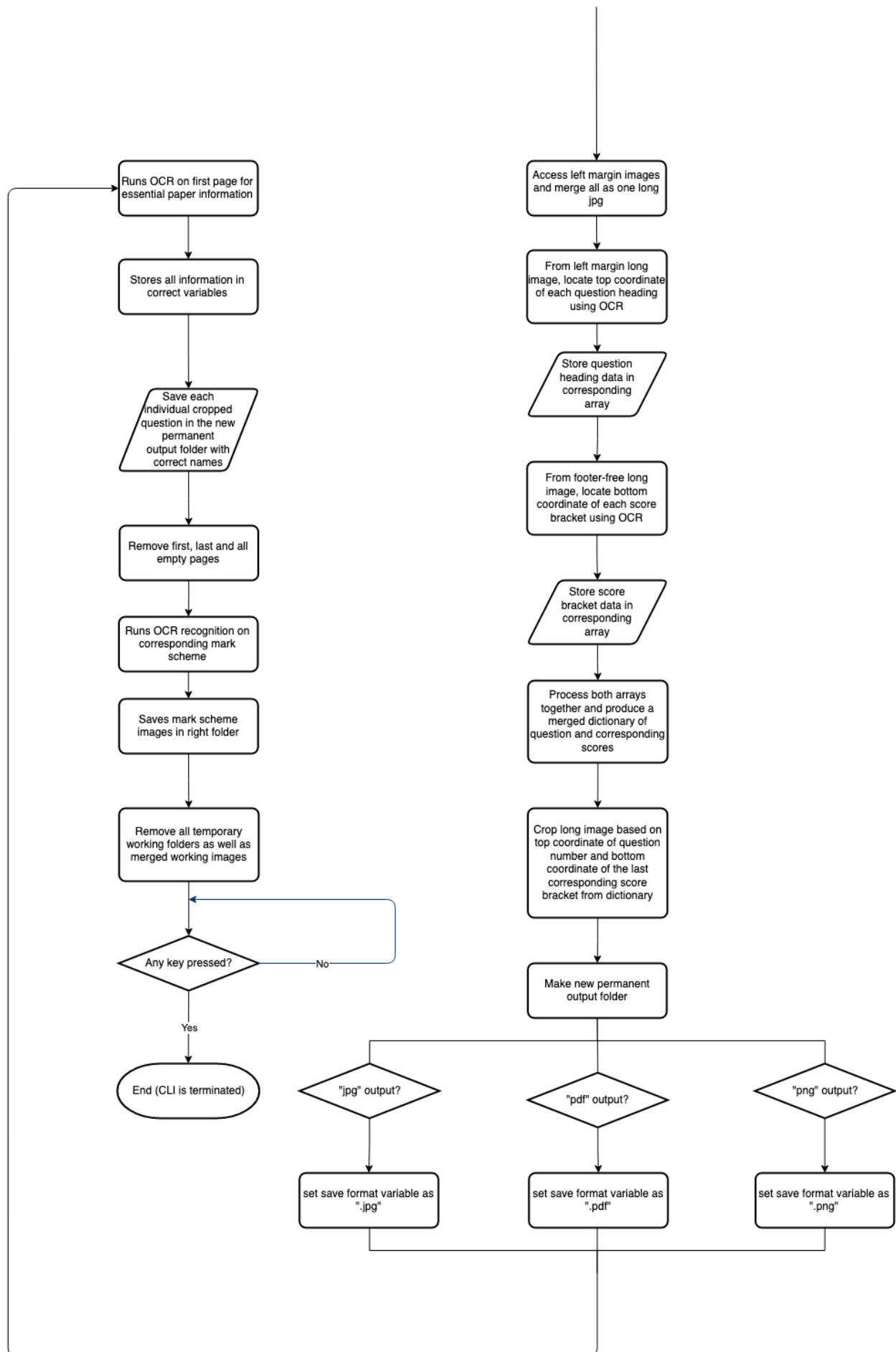
© UCLES 2017

Page 2 of 8

Colour Code	Section Description
Red outline	This table header is repeated and consistent for all tables in the mark scheme, making it useful when cropping individual tables out.
Red fill	First question mark scheme, contains 4 sub-questions
Orange fill	Second question mark scheme
Yellow fill	Third question mark scheme

iii/ Program Overview






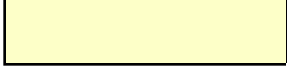








iv/ Program Files

trial_py	Main project folder
— Results	Final permanent output folder
— Q1.jpg	Individual question output images
— Q2.jpg	
— Q3.jpg	
— Q4.jpg	
— Q5.jpg	
— ...	
— main.py	Main python program file
— cropped	Temporary working folder containing footer-free images
— cropped1.jpg	Individual cropped images of each separate page of the original pdf
— cropped2.jpg	
— cropped3.jpg	
— cropped4.jpg	
— cropped5.jpg	
— ...	
— left	Temporary working folder containing left margin images
— left_tab1.jpg	Individual left margins of each page of the original pdf
— left_tab2.jpg	
— left_tab3.jpg	
— left_tab4.jpg	
— left_tab5.jpg	
— ...	
— left_long.jpg	Merged long image of all left margins of each page of the original pdf
— long.jpg	Merged long image of all footer-free pages of the original pdf
— og	Temporary working folder containing original separate pages of the pdf
— out0.jpg	

— out1.jpg	Individual original images of each separate page of the pdf
— out2.jpg	
— out3.jpg	
— out4.jpg	
— out5.jpg	
— ...	
— mark scheme	Folder containing the images of the corresponding mark scheme
— ms0.jpg	Individual cropped images of each separate question of the mark scheme
— ms1.jpg	
— ms2.jpg	
— ms3.jpg	
— ms4.jpg	
— ms5.jpg	
— ...	
— test1.pdf	Python test file (06_0417_11_2020)
— test2.pdf	Python test file (06_0417_11_2019)

v/ Program Functions

Function Name	Colour Code	Parameters	Purpose
initiate_program		N/A	Initiates program by importing all necessary libraries and storing all user inputs
make_folder		path	Creates temporary and permanent folders within user specified path
convert_pdf		path, pages	Reads individual pages of pdf located in user path as individual images and saves in respective temporary folder
crop		img, area, path	Crops footer and left margin off the individual pages and saves output images in respective temporary folder
merge		images	Merges images together into one long image for easier processing and cropping
find_qs		long_img, custom_config	Uses OCR to find the coordinates of specified element using custom OCR configuration
merge_qs		question_result, score_result	Merges question coordinate array and score bracket coordinate array into one dictionary and sorts the scores under their corresponding question numbers
crop_final		qs_dict, long_img	Crops merged long image into individual questions using coordinates from the merged dictionary

			and saves individual questions as separate images based on user specified output format in the respective output folder
delete_folder		path	Deletes all unnecessary files from user specified path
mark_scheme		path, count	Produces separate images of each question from the corresponding mark scheme

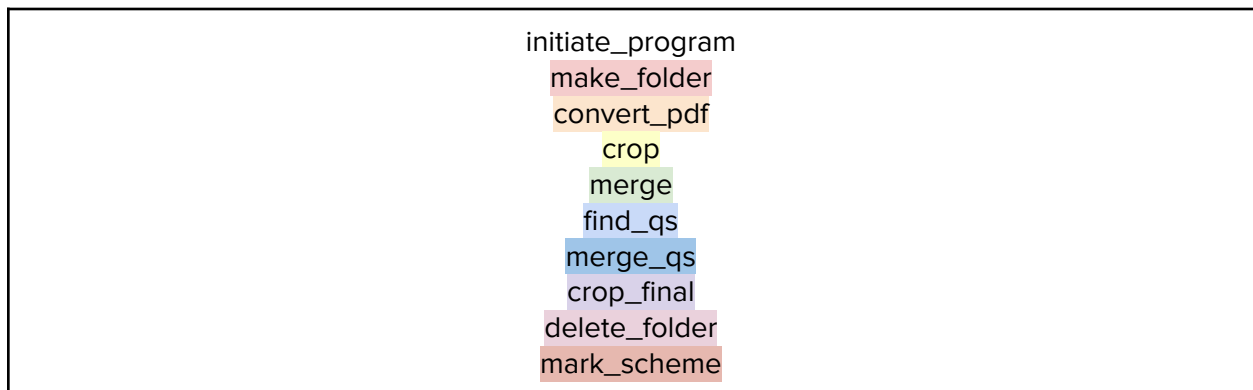
vi/ Program Variables

Variable Name	Variable Type	Purpose
path	Raw string	Holds user input folder path
pages	List (2D)	Holds individual PIL images of each page of the original pdf
img	PIL jpg file	Holds the PIL jpg file of the image to be manipulated
area	Tuple	Holds the fixed area cropped area of the footer/left margin
images	List (2D)	Holds names of each image file for counting purposes
left	List (2D)	Holds names of each left margin file for counting purposes
long_img	PIL image	Holds the PIL image of the long image to be manipulated
custom_config	Raw string	Holds the custom OCR configuration
question_result	List (2D)	Holds the coordinates and data for each individual question number read by OCR
score_result	List (2D)	Holds the coordinates and data for each individual score bracket read by OCR

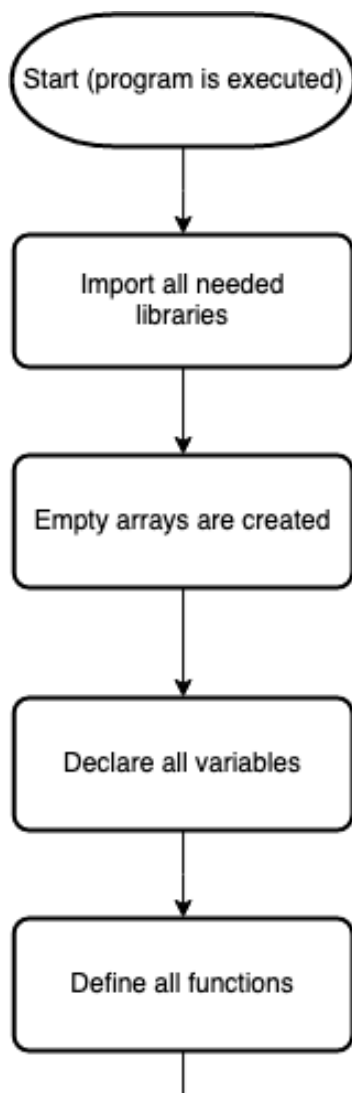
qs_dict	Dictionary	Holds merged and sorted individual questions with corresponding score brackets with question as the key and score brackets as the elements
out_format	String	Holds user specified output format
count	Integer	Holds any temporary counts
total_height	Integer	Holds the total height of all images merged together
min_img_width	Integer	Holds the minimum image width
data	Dictionary	Holds raw data from OCR
pattern	Regex pattern	Holds regex pattern for OCR data manipulation

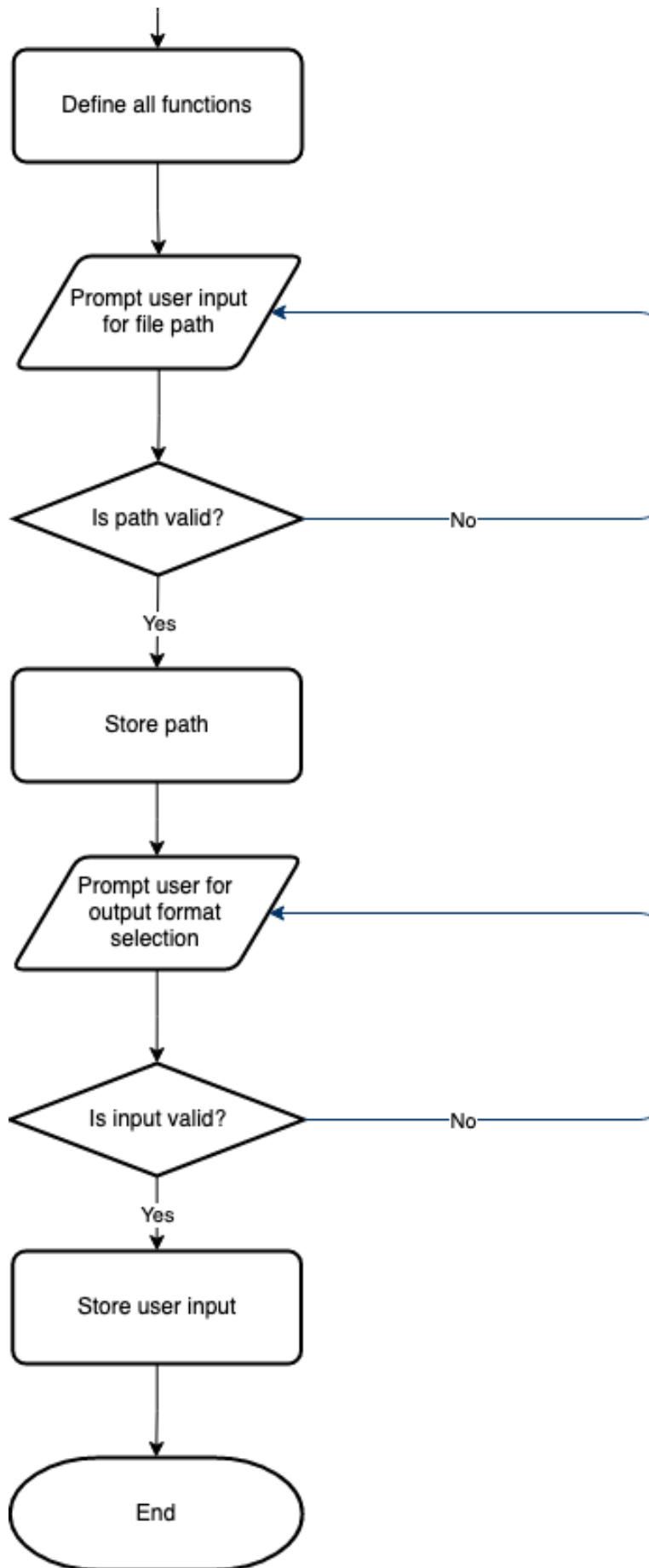
vii/ Detailed Flowchart of each part of program

Each separate flowchart illustrates a different function of the program, indicated using the colour codes as delegated in section “Program Functions”:

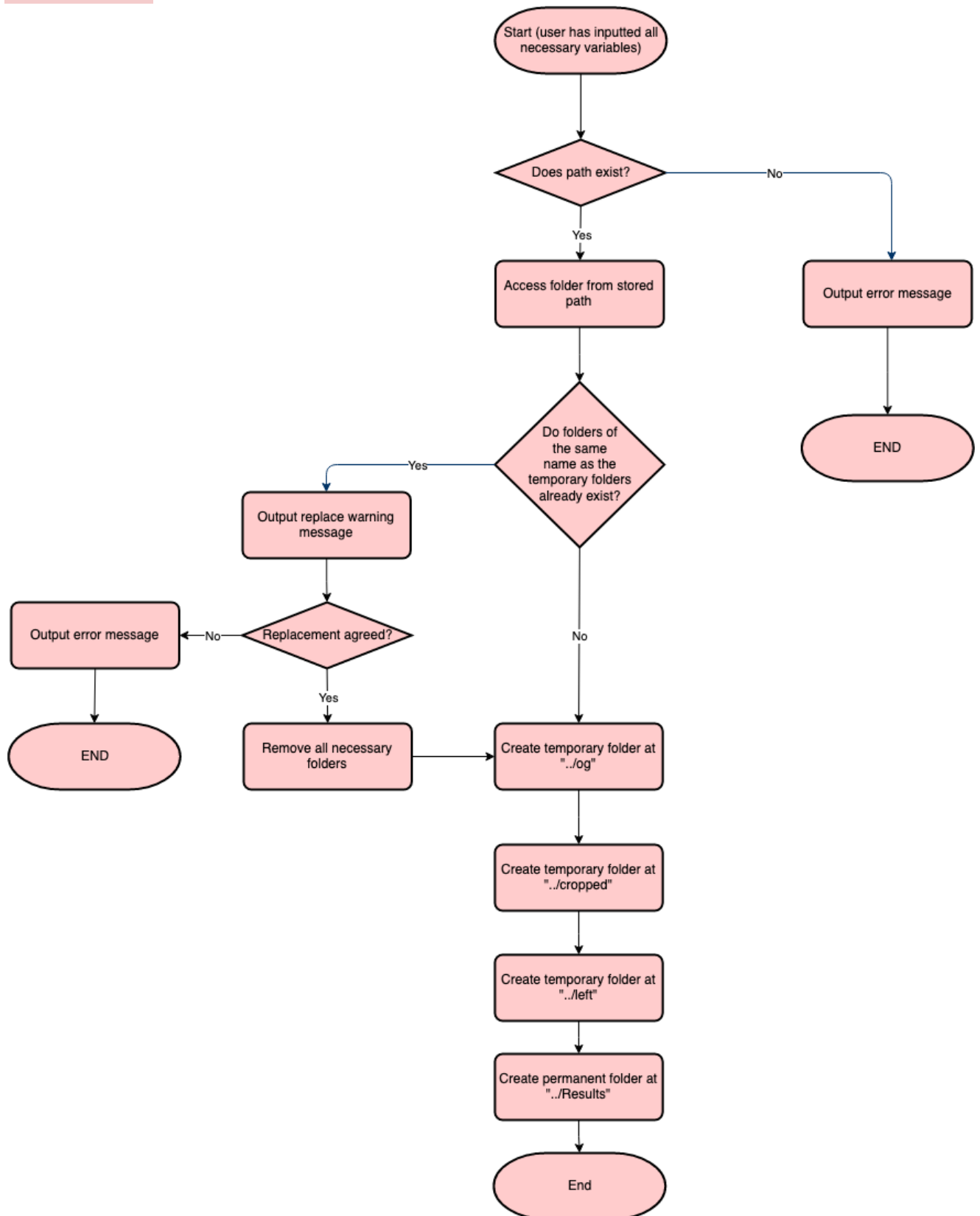


a/ initiate_program

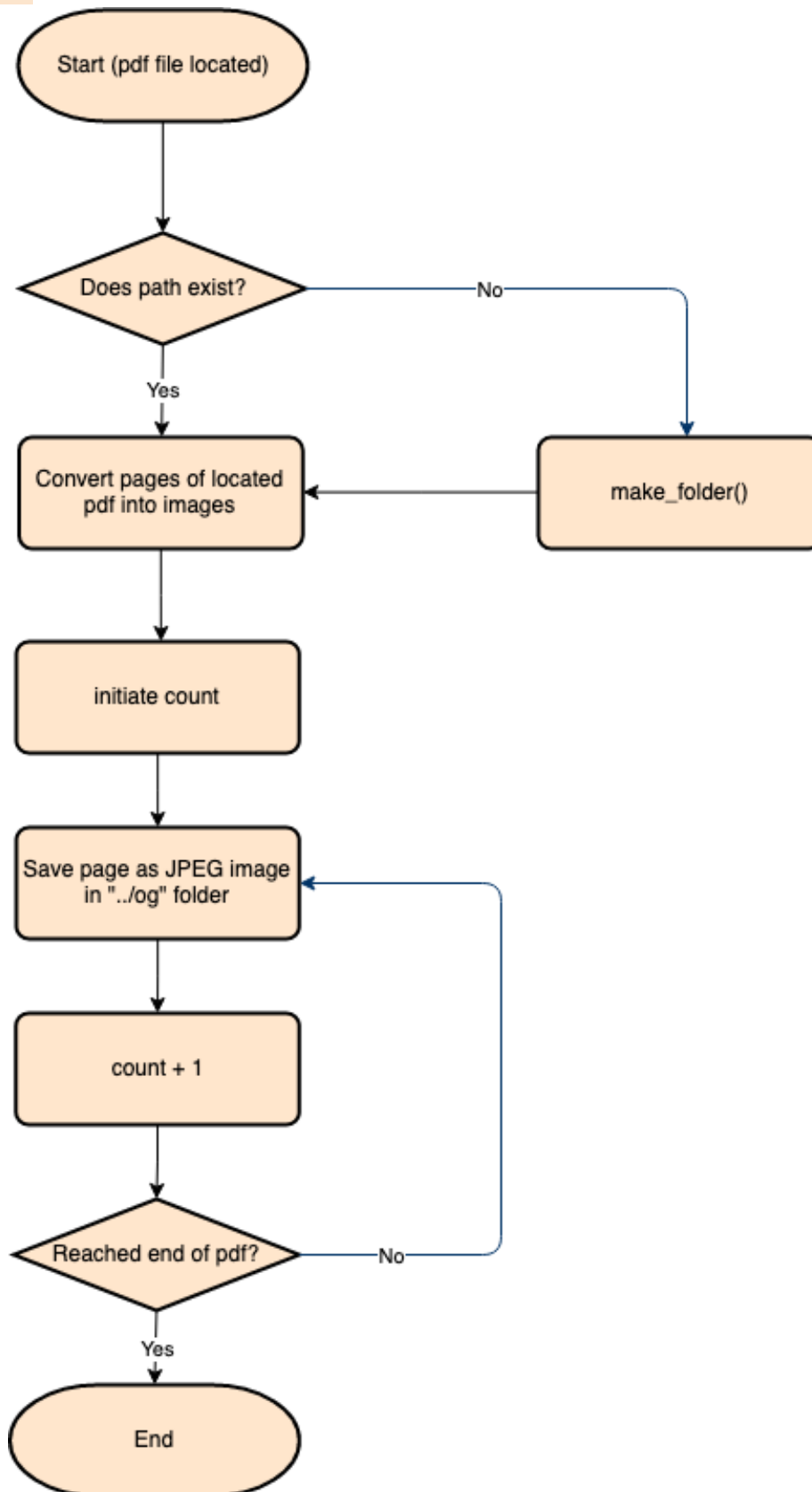




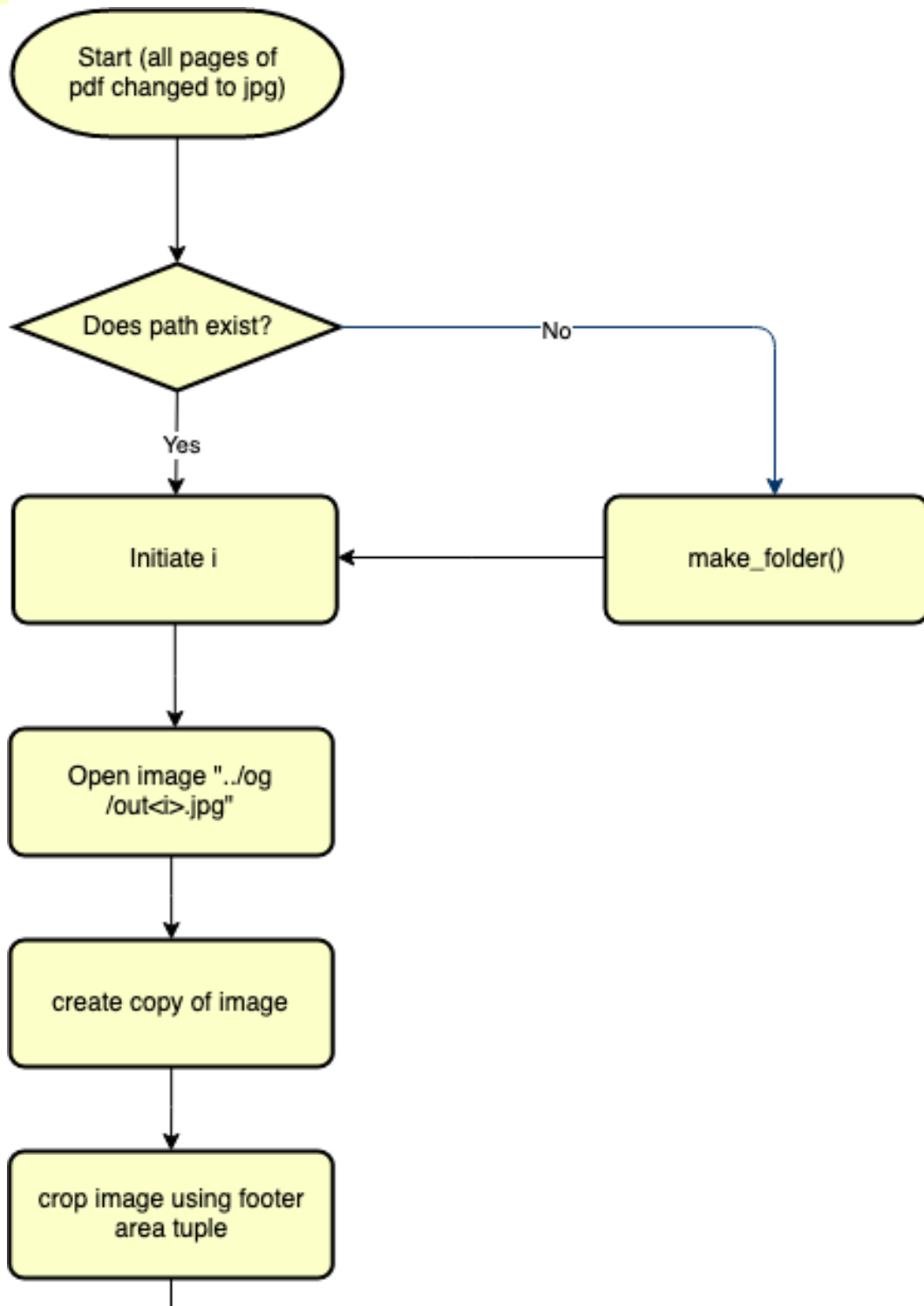
b/ make_folder

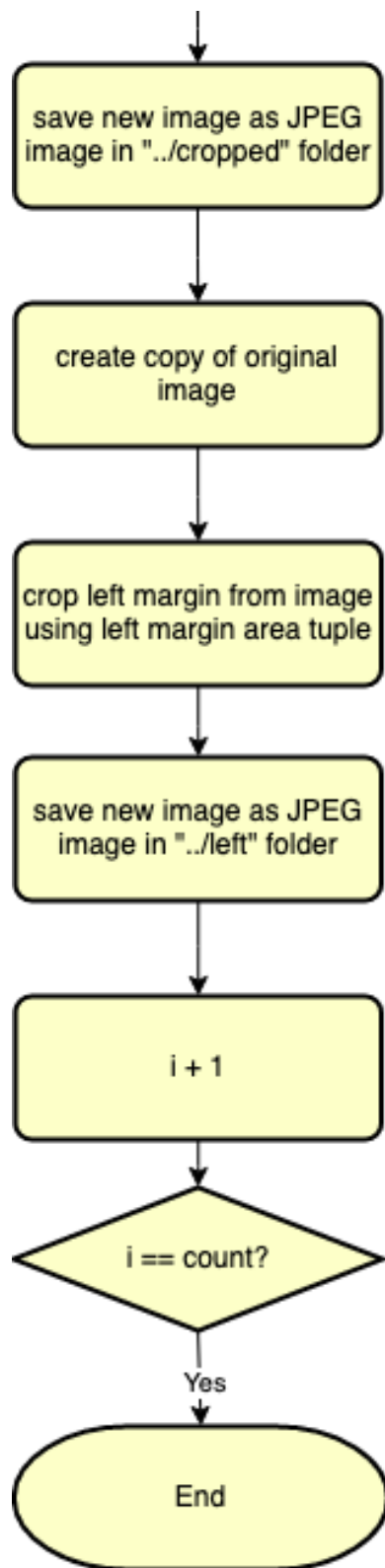


c/ convert_pdf

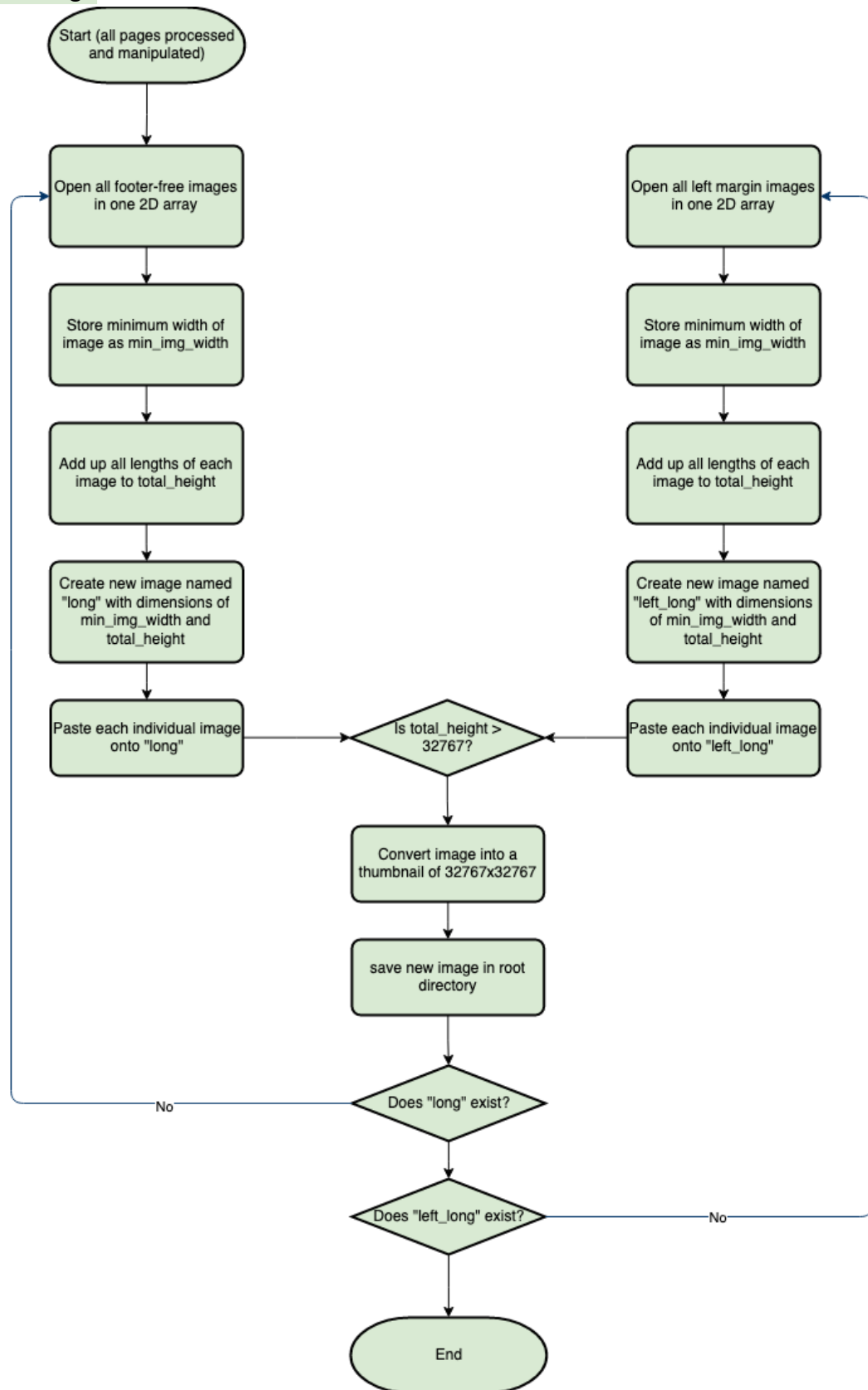


d/ crop

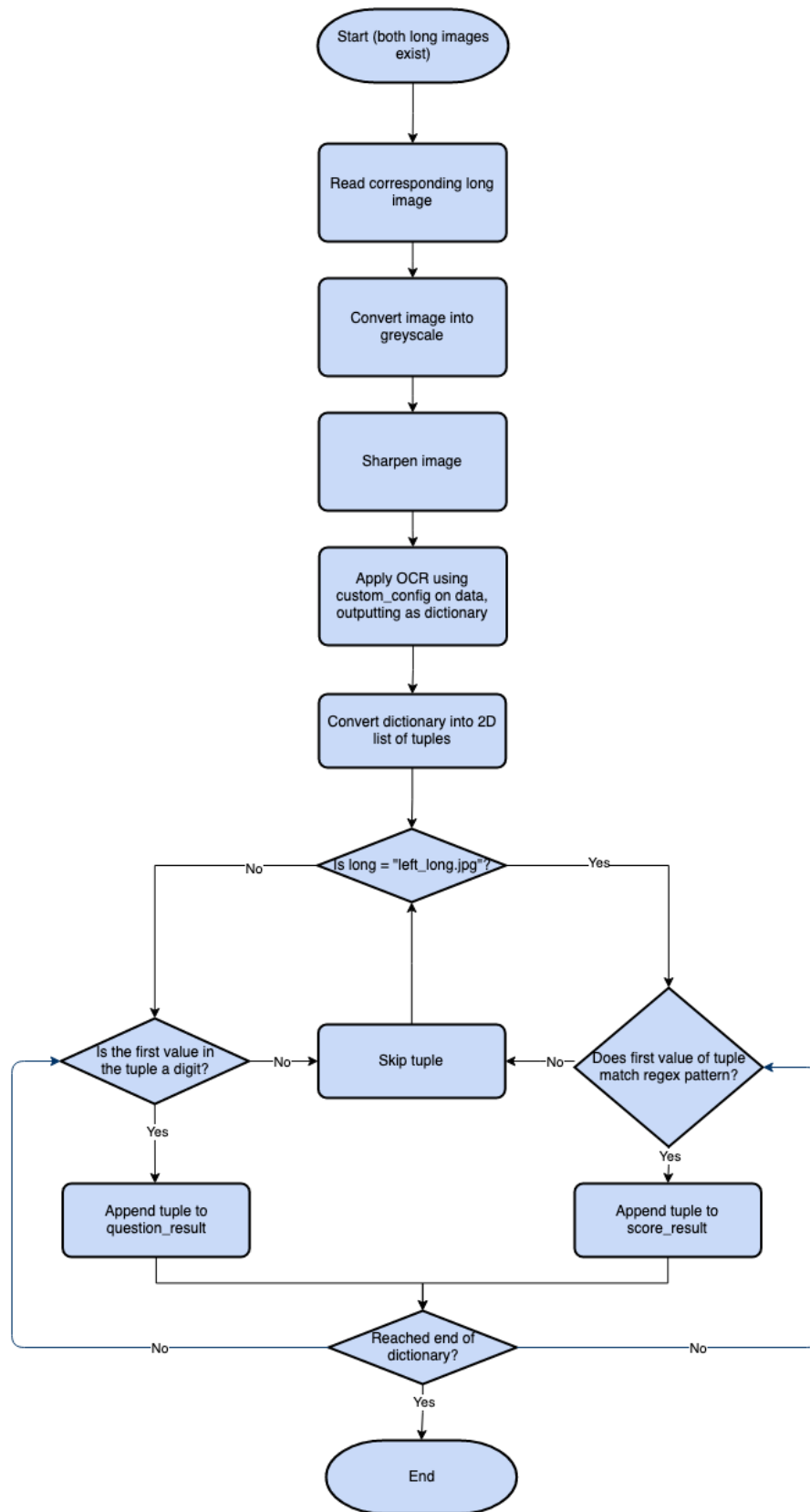


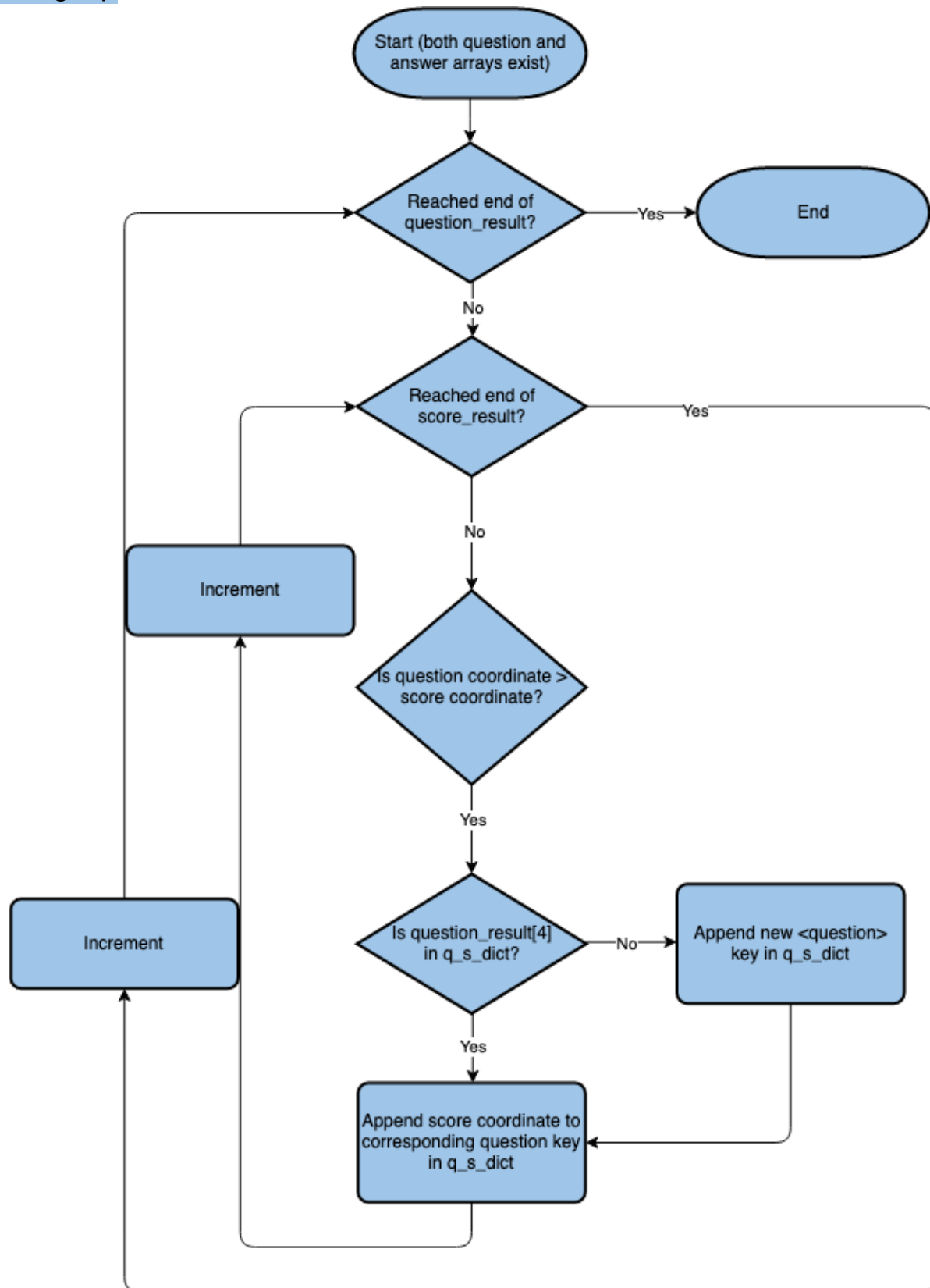


e/ merge

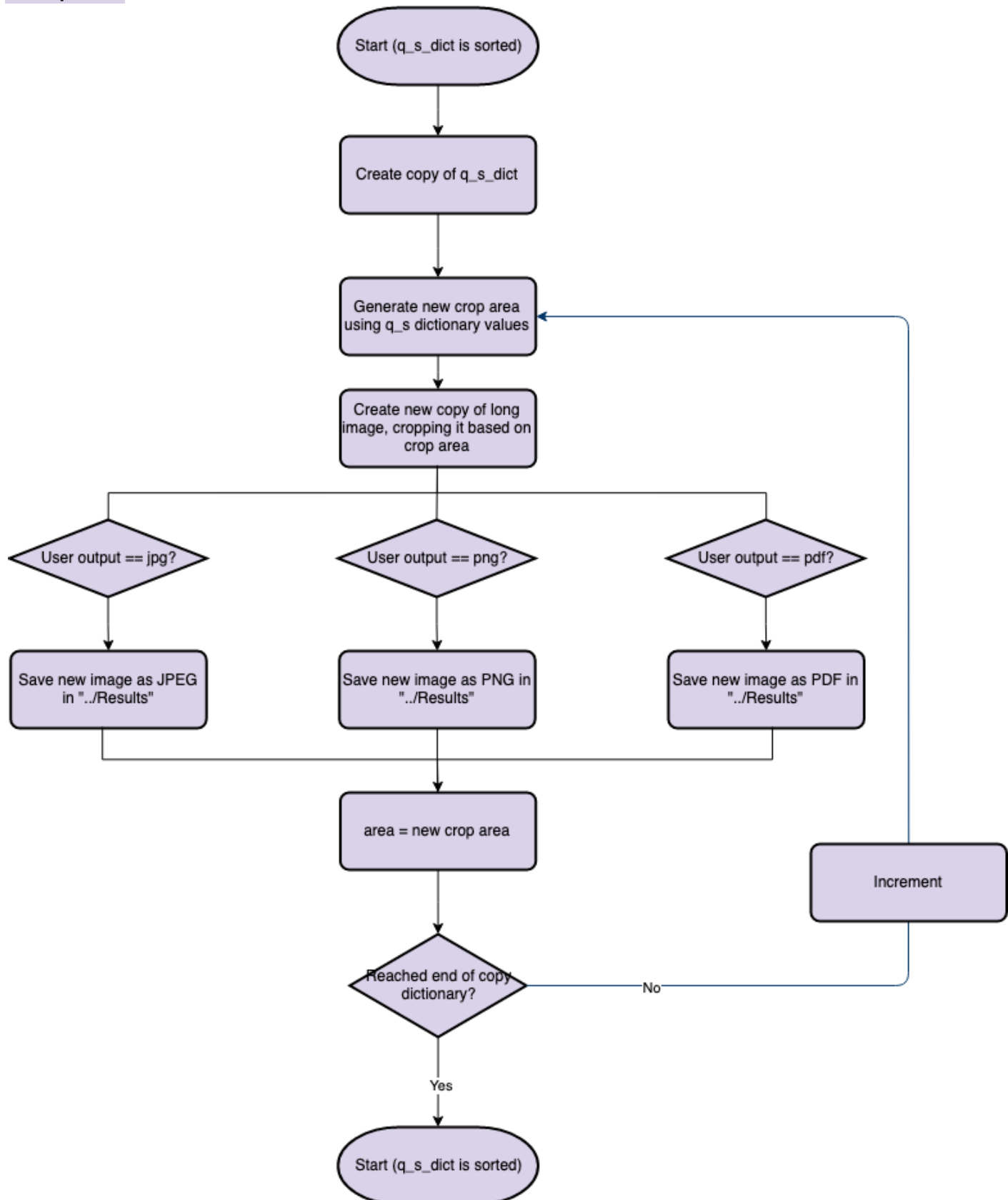


f/ find_qs

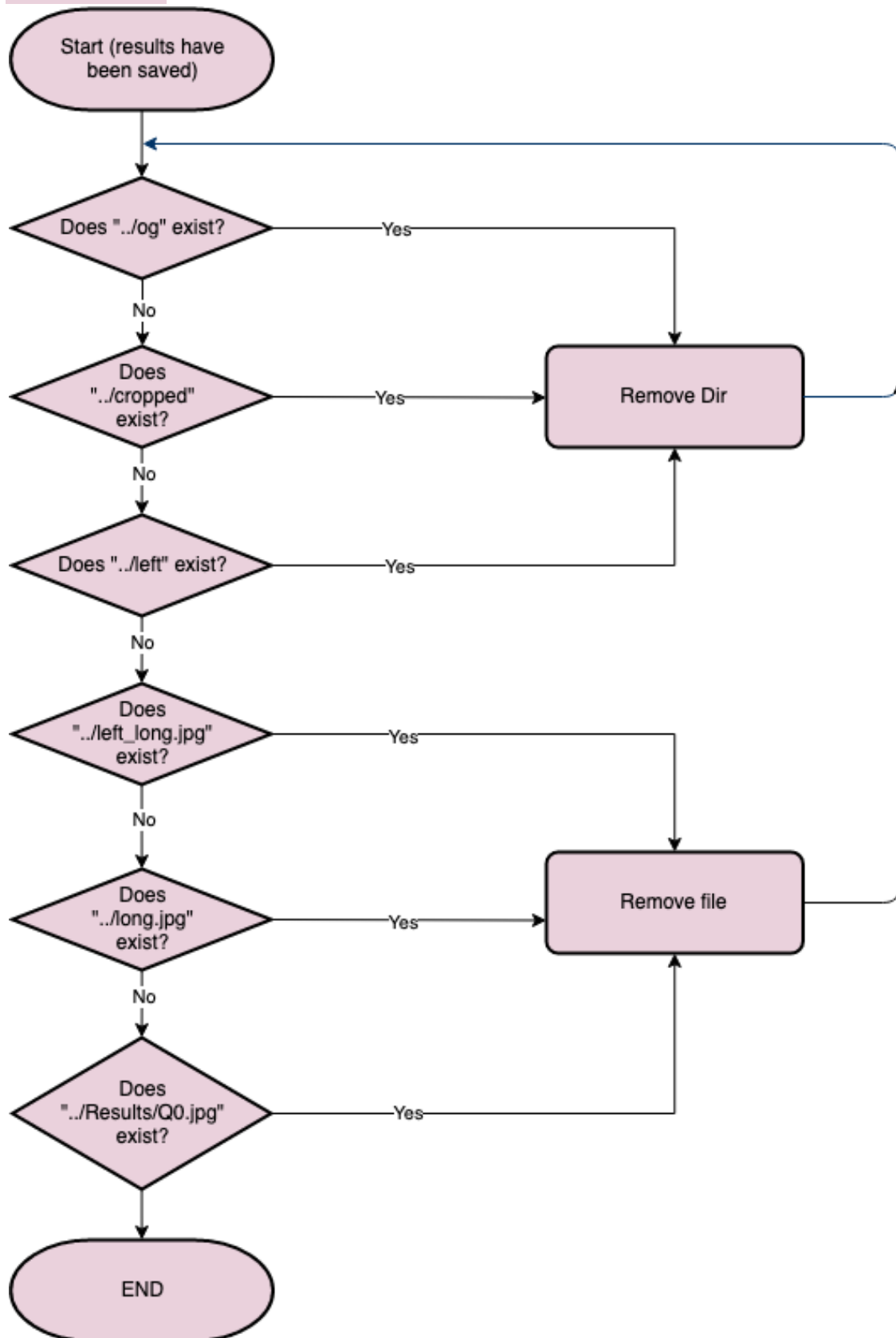




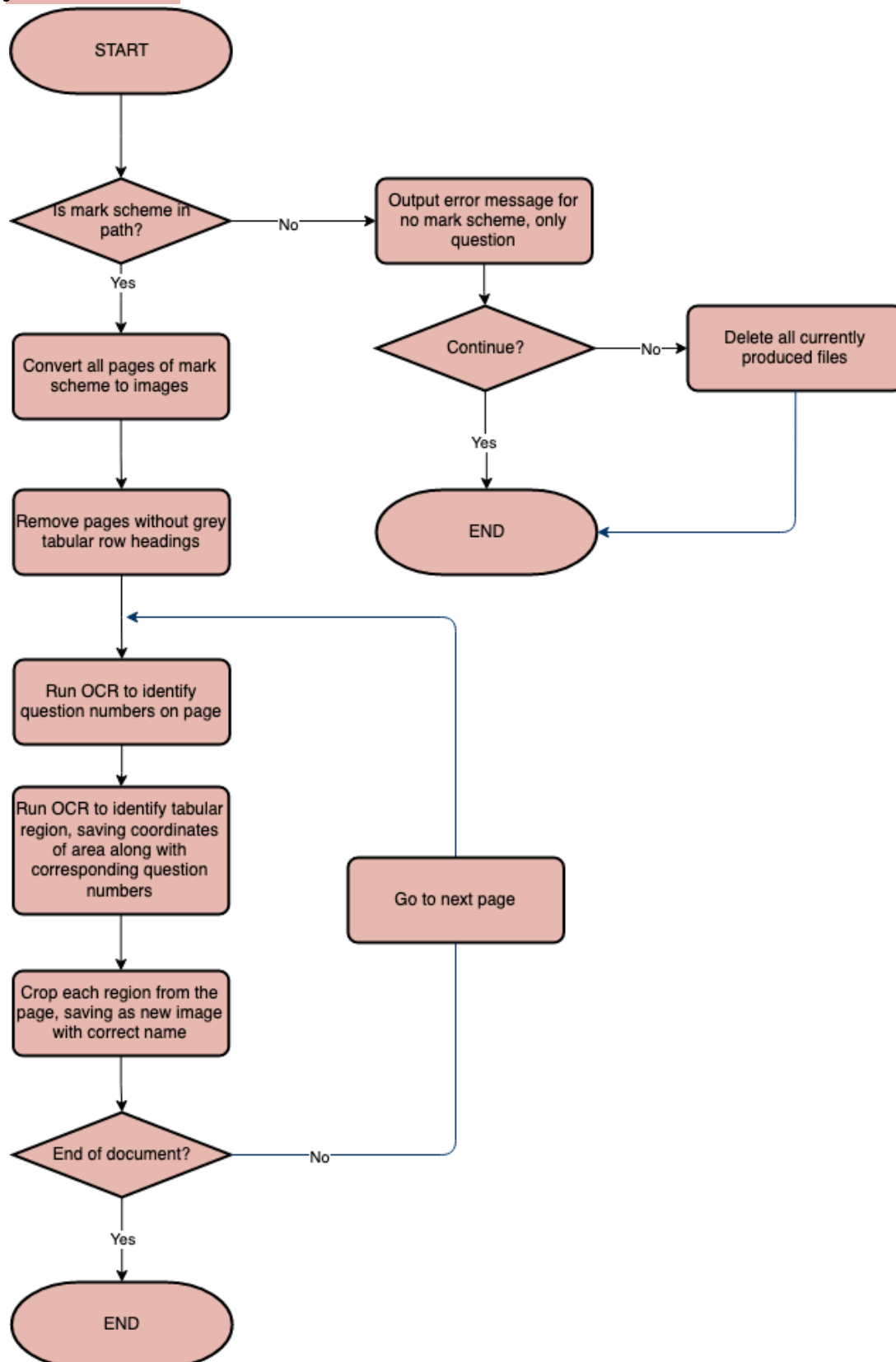
h/ crop_final



i/ delete_folder



j/ mark_scheme



viii/ External Libraries Needed

Library Name	Version	Function
cv2	4.5.1.48	A library called “OpenCV on wheels”. Mainly supports computer vision, but will be used to manipulate image colour thresholds and filtering in order to support PyTesseract’s OCR function.
pdf2image	1.14.0	Library solely used to convert pdf documents into PIL images. The “convert_from_path” function will be used.
pytesseract	0.3.7	Optical Character Recognition (OCR) library for python; a more heavily used library that can be used to read and pick up on text as well as its corresponding data from PIL images.
PIL	8.1.1	Python’s imaging library; the Image module is heavily used for its ability to process and “open” images.
numpy	1.20.1	Used once in tandem with cv2 to apply a sharpen filter.
re		Used once to filter data from tesseract.
os		Used to check the existence of paths and make new directories.
sys		Used to grab command line arguments.
shutil		Used to remove non-empty directories.

ix/ Developer Test Plan

#	S.C.	Test Action	Expected result
1	2	PDF specified has been opened	No error bounced back when trying to convert from path (ie. "PDFPageCountError" due to invalid PDF file path or an invalid (corrupted) PDF)
2	2	PDF specified has been read properly	New updated images list containing separate pages
3	7	Folders have been created in the correct path	Folders can be accessed from system file management
4	2	Images have been correctly cropped	No text overflow or cut-offs
5	3	Cropped images have been saved in the correct folder with the correct name	Corresponding folders contain all expected images and are in order
6	2	Left margins have been correctly cropped	No text overflow or cut-offs, question numbers should flow in order down page
7	2	Left margins have been saved in the correct folder with the correct name	Corresponding folders contain all expected images and are in order
8	2	Cropped images have been merged into one long image and saved with the correct name	One long image file should be newly saved in root folder
9	2	Left margins have been merged into one long image and saved with the correct name	One long image file should be newly saved in root folder
10	2	Question numbers have been read correctly and thoroughly	List variable (question_result) has all question numbers in ascending order
11	3d	Question number coordinates have been identified and stored correctly	Only coordinates have been appended to the question_result list variable
12	2	Score brackets have been read correctly and thoroughly	List variable (score_result) has all question numbers in ascending order
13	2	Score bracket coordinates have	Only coordinates have been appended to

		been identified and stored correctly	the score_result list variable
14	2	Question numbers and scores have been sorted correctly and in a dictionary	q_s_dict variable should be fully updated and have question numbers as keys and scores as elements, including their coordinates
15	4	Long image has been cropped correctly to match all corresponding sub-questions to the correct main question	All sub-questions are included under the main question and there are no text overflows or cut-offs
16	2 + 3	Final output image has been saved in the correct folder with the correct formatted names	Corresponding results folder contains all expected images and are in order with correct names
17	7	All unnecessary files have been removed	Root folder should only contain permanent results folders
18	1	Path as a CLI argument has been validated and accepted	Invalid paths should bounce back error messages (ie. "Error: make sure your spelling is correct AND the file is in the same folder as this program")
19	1	No other CLI arguments are accepted	Additional arguments should bounce back error messages
20	3a	Course code has been stored correctly	Course code variable should be permanent
21	3b	Year and session codes have been read from the first page and stored correctly	Year variable should be updated to correct information, session code as well
22	3c	Paper code has been stored correctly	Paper code variable should be permanent
23	6	Post-2017 Papers can be used as input	Program should produce correct outputs for Post-2017 papers
24	5	Program identifies correct mark scheme based on path name	No error bounced back when trying to convert from path based on expected file name
25	5	Program reads each page of mark scheme correctly and stores as image	New updated mark_scheme images list containing separate pages

26	5	Program deletes unnecessary pages from mark scheme	Images list should be updated to remove all unnecessary pages
27	5	Program identifies each separate table based on grey heading row	Should store each separate instance coordinates in correct dictionary variable
28	5	Program reads first column of each separate table and stores it as question numbers	Should store each separate question number under the correct instance key from the aforementioned dictionary variable
29	5	Program identifies coordinates of each separate table	Each table is identified as separate and stored as such, no overflowing and double saving. Coordinates are updated to dictionary element values.
30	5	Crops based on coordinates on each page as separate images	All sub questions are included under the main question mark scheme table and there are no text overflows or cut-offs
31	5	Saves cropped images as individual mark scheme images with correct naming in the correct folder	Corresponding results folder contains all expected mark scheme images and are in order with correct names