National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Software Testing	Course Code:	CS4036
Degree Program:	Computer Science	Semester:	Fall 2022
Exam Duration:	180 Minutes	Total Marks:	90
Paper Date:	19th Dec, 2022	Weight	45
Section:	7A & 7B	Page(s):	14
Exam Type:	Final Exam		

Student : Name:_____________________________ Roll No.________________ Section:_______

Instruction/Notes:

Attempt all questions on the question paper. Answer sheets are not required.
Take Assumptions where required and note them down along with your answers.

Question #1:[5+10+10]

You have been hired as a consultant by Slack to automate their web API. One of the web API endpoints that you need to test is the following post request to archive a channel on slack.

POST* *https://slack.com/api/conversations.archive

The post request requires following header parameters:

- Content-type: application/json
- Authorization: Bearer xoxp-xxxxxxxxx-xxxx

Following are the body attributes that needs to be sent in JSON format:

- Channel: (the name of the channel to be achieved)

A sample post request:

POST /api/conversations.archive

Content-type: application/json

Authorization: Bearer xoxp-xxxxxxxxx-xxxx

{"channel":"channel-name"}

It has following possible responses:

- When successfully archived:
  - { "ok": true}
- When failed to archive because:
  - Channel does not exist: { "ok": false, "error": "channel_not_found"}
  - Trying to archive an already archived channel: { "ok": false, "error": "already_archived"}
  - Channel “General” cannot be archived { "ok": false, "error": "cant_archive_general"}
  - Token does not have permissions to archive: { "ok": false, "error": "access_denied"}
  - token inactivated: { "ok": false, "error": "account_inactive"}
  - No token provided: { "ok": false, "error": "not_authed"}
  - No channel name provided: { "ok": false, "error": "not_authed"}

a) Identify equivalence Classes and boundary values for following fields

   i) Content-Type
   ii) Authorization
   iii) Channel

b) List down 15 different test scenarios e.g.

   i) Verify that correct error message “channel_not_found” is returned when web api request contains a valid token but a random string in channel field

c) Write API test automation script with proper structure, post commands and assertions to automate 3 of the above web API test scenarios (Use your preferred language and respective libraries)

Question #2:[3 + 10 + 2 + 5]

You are asked to perform load and stress testing for the following two web pages of Daraz.pk

- Product listing Page
    - https://www.daraz.pk/baby-gear/
- Product Detail Page
    - https://www.daraz.pk/products/-i109258380-s1259290710.html

a) What are the different types of tools you will need to do performance testing

b) Write your load test script using your preferred tool (explain in steps if you are familiar with jmeter). Your load test should simulate following load:

i) Starting from 0 and reaching to maximum number of 4000 users within 10 minutes

ii) Maintaining the 4000 users load for 30 minutes

iii) 80% of the users should be navigating to Product Listing page where as 20% should navigate to product detail page

c) What changes would be required in above to perform the stress testing

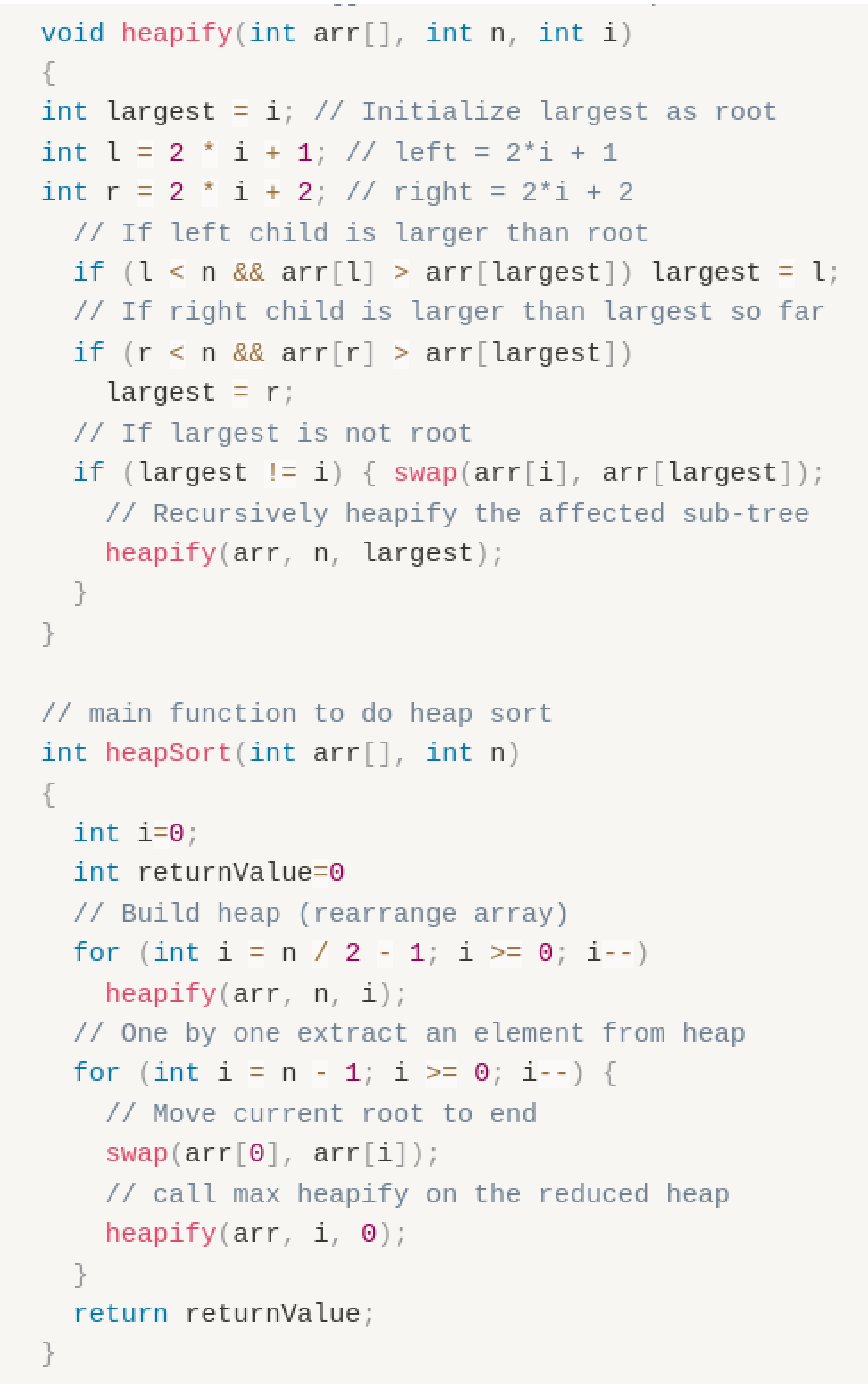
d) Load generation for performance testing should consider “A performance test load must represent many user inputs, not just one”. Is there any issue in our above test scenario that does not possibly comply with the condition? Explain why / why not?

Question #3:[2 + 3 + 5 + 3 + 2]

a) What are the different benefits of static analysis

b) Explain in steps on how you will add a linter to your project. (You can use your preferred linter in your preferred language). And how you can customise the rules configuration according to your needs.

c) Identify issues that static analysis tools can help us identify in the following code



```
void heapify(int arr[], int n, int i)
{
int largest = i; // Initialize largest as root
int l = 2 * i + 1; // left = 2*i + 1
int r = 2 * i + 2; // right = 2*i + 2
  // If left child is larger than root
  if (l < n && arr[l] > arr[largest]) largest = l;
  // If right child is larger than largest so far
  if (r < n && arr[r] > arr[largest])
    largest = r;
  // If largest is not root
  if (largest != i) { swap(arr[i], arr[largest]);
    // Recursively heapify the affected sub-tree
    heapify(arr, n, largest);
  }
}

// main function to do heap sort
int heapSort(int arr[], int n)
{
  int i=0;
  int returnValue=0
  // Build heap (rearrange array)
  for (int i = n / 2 - 1; i >= 0; i--)
    heapify(arr, n, i);
  // One by one extract an element from heap
  for (int i = n - 1; i >= 0; i--) {
    // Move current root to end
    swap(arr[0], arr[i]);
    // call max heapify on the reduced heap
    heapify(arr, i, 0);
  }
  return returnValue;
}
```

d) Create a control flow graph for ***heapSort*** function and compute cyclomatic complexity. (Consider ***heapify*** function call in the he ***heapSort*** function as single line statement)

e) Identify input value for ***heapSort*** function to achieve 100% branch coverage

Question #4 [5 + 10]:

My e-commerce platform needs to have some loyalty benefits for customers that utilise my platform regularly. They are awarded points on each purchase. The points are awarded in following way:

- If a customer has purchased total of worth Rs 10000 - Rs 50000 then the customer gets 1 point per Rs 1000
- If a customer has purchased total of worth Rs 50000 - Rs 100000 then the customer gets 5 points per Rs 1000
- If a customer has purchased total of worth Rs 100000 or more then the customer gets 10 points per Rs 1000
- No points are awarded to people who has less than Rs 10000 of total purchase
- If a customer makes a single purchase of more than Rs 5000 then they will get 10 extra points

a) Correctly identify the inputs and their possible values based on the above requirement: Hint (Inputs does not need to be only T/F. One input can have multiple values)

b) Create a reduced decision table to identify the optimised number of test cases

Question #5 [5 + 10]:

Whatsapp wants to add functionality to create a community in Whatsapp. Each community can have multiple group chats. People can be added to each group chat individually or the community. As the tester in the team, you have been asked to create a test plan for the new feature before the final release of the project.

a) Which of the test strategies will you apply? You need to choose the most appropriate strategies. Give valid reasons.

b) Assuming that Whatsapp is already following good software development and delivery practices (Agile + DevOps). And therefore they already have well developed testing practices in place (e.g. Test Case management, bug reporting process, test environments, test data management, etc). Considering that you need to a quick plan for following user story:

   i) As a whatsapp user, I should be able to create a new community by entering a valid and unique community name

   Considering the different parts of a test plan, Identify the test plan sections that will be critical to create a small test plan for this individual user story