MODULE O ROBUST PROGRAMS · O4

Authentication in programs: mark scheme

22 marks · OCR J277 specification 2.3.1 (defensive design: anticipating misuse, authentication to confirm the identity of a user, practical experience of designing simple authentication, e.g. username and password), with 2.1.2 (trace tables) and 2.2.3 (arrays, functions)

How to use. 1 mark per bullet up to the maximum shown. **/** separates alternative words or answers; **//** separates alternative statements for the same mark. Words in (brackets) are not needed for the mark. … at the end of one point and the start of the next means the second depends on the first. Code may be in OCR Exam Reference Language or any high-level language: mark the logic, and do not penalise minor syntax slips (e.g. a missing `then`, `endif`, colon or bracket). Credit any other valid answer that shows the same understanding.

QUESTION	ANSWER	MARKS	GUIDANCE
1(a) J277 2.3.1	Confirming / checking the identity of a user (before they are allowed access to the program / its data)	**1**	Allow: checking that a user is who they say they are // checking a user is allowed to access the system. An example alone (e.g. "entering a username and password") is NE. Allow BOD if the example is linked to the identity check (e.g. "checking the password matches the stored one to prove who the user is"). Do not accept: checking that the data entered is sensible / valid (that is validation).
1(b) J277 2.3.1	1 mark per correct row: Seats between 1 and 8: Input validation Password entered again before a booking is deleted: Authentication Username and password compared with the stored ones: Authentication	**3**	A row with both boxes ticked scores 0.
2(a) J277 2.3.1, 2.2.1	1 mark per correct row: `boxoffice2`, `Curtain#47`: Access granted `boxoffice2`, `Curtain#74`: Access denied `boxoffice3`, `Curtain#47`: Access denied	**3**	Ignore quotation marks and case of the output text. The message must be the program's exact output: do not accept "denied" / "wrong password" / "no access".

<table>
<tr><th>QUESTION</th><th>ANSWER</th><th>MARKS</th><th>GUIDANCE</th></tr>
<tr><td>2(b)
J277 2.3.1, 2.2.1</td><td>Max 2 from:
Access would be granted if only one of the username or the password is correct // both no longer need to be correct
Someone who knows / guesses (only) a valid username could log in without the password // someone could log in with the password but any / a wrong username
So people who are not authorised / not the staff member could get in // the program no longer confirms the user's identity</td><td>2</td><td>Allow the correct row of 2(a) as an example: the 2nd and 3rd inputs would now give Access granted (1 mark, as MP1 or MP2).
“Less secure” / “anyone could get in” on its own is NE.
Do not accept: “nobody could log in” / “both must still be correct”.</td></tr>
<tr><td>3(a)
J277 2.1.2, 2.3.1</td><td>1 mark per bullet:
<code>attempts</code>: 1, 2, 3 (after the given 0)
<code>loggedIn</code>: changes to <code>true</code> only once, on the 3rd attempt (the row where <code>pwd</code> is Curtain#47)
<code>pwd</code>: Curtain47, Curtain#74, Curtain#47 (in this order)
Output: Try again, Try again, Welcome (in this order, Welcome last)
<table>
<tr><th>ATTEMPTS</th><th>LOGGEDIN</th><th>PWD</th><th>OUTPUT</th></tr>
<tr><td>0</td><td>false</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Curtain47</td><td>Try again</td></tr>
<tr><td>1</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>Curtain#74</td><td>Try again</td></tr>
<tr><td>2</td><td></td><td></td><td></td></tr>
<tr><td></td><td>true</td><td>Curtain#47</td><td></td></tr>
<tr><td>3</td><td></td><td></td><td>Welcome</td></tr>
</table></td><td>4</td><td>Values may be repeated on later rows (e.g. <code>false</code> written on each row until it changes). Allow rows to be combined, e.g. <code>attempts</code> 1 on the same row as Curtain47 and Try again.
Allow True/False, T/F, 1/0 for <code>loggedIn</code>.
<code>attempts</code>: do not award if 4 appears.
Output: do not award if Locked appears, or if a third Try again appears. Ignore quotation marks.</td></tr>
<tr><td>3(b)
J277 2.3.1 (anticipating misuse, authentication), 1.4.1</td><td>Max 2 from:
To stop someone (repeatedly) guessing / trying many passwords // to prevent a brute-force attack ...
... after 3 wrong attempts the account / program is locked // the person only gets 3 tries ...
... so an unauthorised person is very unlikely to find the correct password // it would take far too long to try every possible password</td><td>2</td><td>“To make it more secure” / “to stop hackers” on its own is NE.
Allow “stops a program / bot trying every combination” for MP1.</td></tr>
</table>

QUESTION	ANSWER	MARKS	GUIDANCE

4(a)
J277 2.3.1, 2.2.3 (arrays, functions), 2.2.1
	1 mark per bullet to max 5:
	Loop / iteration that goes through the array(s)...
	... covering every item, index 0 to 19 / `users.length - 1` (and no further)
	Compares `users[i]` with the parameter `user`
	Compares `passwords[i]` with `pwd` using the **same** index as the matching username
	Returns `true` when both match
	Returns `false` only after every item has been checked with no match (e.g. after the loop)
	Example answer
	`function checkLogin(user, pwd)`
	`    for i = 0 to users.length - 1`
	`        if users[i] == user AND passwords[i] == pwd then`
	`            return true`
	`        endif`
	`    next i`
	`    return false`
	`endfunction`
	Alternative (flag)
	`function checkLogin(user, pwd)`
	`    found = false`
	`    i = 0`
	`    while i < 20 AND found == false`
	`        if users[i] == user then`
	`            if passwords[i] == pwd then`
	`                found = true`
	`            endif`
	`        endif`
	`        i = i + 1`
	`    endwhile`
	`    return found`
	`endfunction`
		5	Allow `for i = 0 to 19`, `for i in range(20)` / `range(len(users))`. Do not award MP2 for `0 to 20` or `1 to 20`.
			MP3 and MP4 may be in one condition joined with AND, or as nested IFs. Do not award MP4 if the password is searched for in a separate loop / at a different index (e.g. any password in the array).
			Do not award MP6 if `return false` is in an `else` inside the loop (the function would stop after checking only the first item).
			A flag variable returned at the end is acceptable for MP5 and MP6.
			Must use the parameters: if `user` / `pwd` are overwritten by `input`, max 4.
			Printing "true"/"false" instead of returning: do not award MP5 or MP6.
			Allow = or == for comparison.

QUESTION	ANSWER	MARKS	GUIDANCE
5 **J277 2.3.1 (anticipating misuse, authentication), 1.4.2**	1 mark for a way, 1 mark for how it makes logging in harder, e.g.: Make staff use strong passwords (e.g. a minimum length, a mix of letters, numbers and symbols) … so passwords are harder to guess / take longer to brute-force Show the same message whichever item is wrong / do not say whether it was the username or the password … so an attacker cannot find out which usernames are valid Add a second step / two-step (two-factor) authentication, e.g. a code sent to the staff member's phone … so a person who knows the password also needs the staff member's device Add a delay / wait time after each failed attempt … so guessing many passwords takes too long Hide the password as it is typed (e.g. show *) … so people nearby cannot read it Store the passwords encrypted / not as plain text in the code or file … so someone who sees the program / file cannot read the passwords	**2**	Mark the first way only; the 2nd mark must explain that way. Credit any other valid way with an explanation (e.g. biometrics; regular password changes). Do not accept: locking the account after failed attempts (that is limiting attempts); firewall / anti-malware; encryption not linked to the stored passwords (e.g. 'encrypt the data' alone); 'more secure' alone.

Total: 22 marks