1 Eve enjoys playing board games. Her favourite board game is called “Pot Luck”. This has a numbered grid of 10 squares by 10 squares. Each square has a number between 1 and 100.

Players place their game counters on square 1. A 30-minute timer is set which counts downwards. Each player rolls two 6-sided dice and then moves their game counter that number of squares. Some squares tell the player to pick up a card. These have instructions on, such as 'Move forward 10 spaces'. If the player lands on one of these squares they move according to the instruction on the card. The first player to land on square 100, is announced as the winner. If no winner is announced before the timer runs out, then it is a draw.

Eve would like to break the problem down into smaller sub problems so that each sub problem will complete one specific task.

Identify **three** sub problems that Eve can use in her game.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

2 A program is being designed that will allow a user to log into an account on a website using a username and password.

Identify **two** possible sub-procedures that could be used in this program.

1 .....

.....

2 .....

.....

[2]

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

3 Mabel is a software engineer. She is writing a computer game for a client. In the game the main character has to avoid their enemies. This becomes more difficult as the levels of the game increase.

The computer game allows a user to select a character (e.g. name, gender). They can then choose a level for the game (easy, normal, challenging). The user controls their character by moving it left or right. The character can jump using space bar as an input. If the character touches one of the enemies then it loses a life. The character has to make it to the end of the level without losing all their lives.

The game is designed in a modular way.

i. One sub-procedure will handle the user input.

Describe **three** other sub-procedures Mabel could create for the given game description.

1 ---

---

2 ---

---

3 ---

---

[6]

ii. Describe the decision that the program will need to make within the user input subprocedure and the result of this decision.

[2]

iii. Define pipelining and give an example of how it could be applied in the program.

[2]

4 Janet is designing a piece of software for a furniture company.

The software will allow a user to plan the position of furniture in a room. Users will be able to set the size and shape of a room, and then choose furniture from a library of furniture items. These pieces of furniture will have set sizes and designs and the user will be able to view the room in 3D to see how it looks from a variety of angles.

Janet is going to decompose the problem to produce a set of subprograms.

Explain the benefits of using subprograms to produce this software.

[4]

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

5 A group of A-level students are working together to program a computer game.

In the game, the player controls a character who moves through a virtual world. The game starts with a load-up screen. The player can select which area to move to on an on-screen map, and then they control the movements of their character using a keyboard to solve puzzles on the screen.

The game is to be created using sub-procedures. The following table identifies and describes one sub-procedure the students could use.

Complete the table below, identifying **three** additional sub-procedures that the students could create from the description at the start of question **2**.

Describe the purpose of each sub-procedure you have identified.

	Sub-procedure	Purpose
e.g.	`characterMovement`	*Takes the key the player pressed and moves the character in that direction*
1		
2		
3		

[6]

END OF QUESTION PAPER

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
1			1 mark per bullet to max 3 e.g. • Initialising a new board with 100 squares • Placing players on square 1 at the start • Setting a new 30-minute timer • Rolling two dice and displaying the total • Moving the counters the number of places on the dice • Choosing an instruction card and displaying the message • Checking if a player has landed on square 100 • Checking if the timer is 0	3	Allow other suitable examples that are relevant to the scenario. <u>Examiner's Comments</u> Some candidates struggled to apply decomposition to identify parts of the scenario that could be broken down into sub problems. Examples needed to relate to the gameplay for the given scenario and needed to be qualified in some way. For example, 'Dice' on its own was insufficient, whereas 'Generating dice score' was sufficient.
			Total	3	
2			1 mark each to max 2 e.g. • Connect to database • Access usernames in file/database • Check username against file/database • Hash password • Access password/hash in file/database • Check password entered/hashed vs stored • Output result	2	Allow other suitable subprocedures that link to the scenario. <u>Examiner's Comments</u> This question was also generally well answered, but answers had to be given in the context of the scenario. Thinking procedurally is specification point 2.1.3 which requires suitable procedures for a given context to be identified. The context was explicitly for logging in to a system, not for creating user accounts or setting up/validating password construction. The most common responses were checking if the username existed and checking whether the username/password combination was correct.
			Total	2	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Mark Scheme

Question			Answer/Indicative content	Marks	Guidance
3		i	1 mark per bullet, max 2 per sub-procedure e.g. • Select character (name, gender) • Gives the user options for choosing a character • Choose level • Give the user the choice of level (easy, normal, challenging) and take the user input • Touch enemy • Called to determine if the character touches an enemy • Lose life • Remove a life, if <0 then game over • End level • Move onto next level One mark for identifying sensible subroutine, 1 mark for description	6 AO2.1 (2) AO2.2 (2) AO3.2 (2)	Do not award any user *input* related procedures e.g. Left/Right input (but character movement *output* on screen left/right would be valid) Allow other reasonable responses from the scenario e.g. generate enemy() <u>Examiner's Comments</u> Most candidates presented reasonable procedures within the context of the question. Relatively few made the mistake of reiterating procedures to handle character movement that was specified as an example in the question.
		ii	1 mark per bullet • Decision based on what the user has input • E.g. If they click left move the character left // if they click right move the character right // if they click space bar make the character jump	2 AO2.1 (1) AO2.2 (2)	<u>Examiner's Comments</u> A number of answers lacked the clarity of a clear decision and subsequent action that would be performed. A number of answers given did not relate to the user input sub-procedure as specified.
		iii	1 mark per bullet • The result from one process / procedure feeds into the next • E.g. the result of detecting a character touching an enemy feeds into reducing the number of lives	2 AO1.2 (1) AO2.2 (1)	Note: 1 Mark Max for a generic description of pipelining <u>Examiner's Comments</u> Many candidates could define the concept of pipelining. Fewer were able to apply it within the context of the given scenario.
			Total	10	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder

Question			Answer/Indicative content	Marks	Guidance
4			1 mark per bullet to max 4 • Efficiency • Small sub-programs are easier to read / understand / modify • Write once and call repeated times • Avoids repeated code • Subroutines can be tested individually then added to the main program. • Can reuse in other programs • Can give procedures to different programmers to build	4 AO1.1 (1) AO1.2 (1) AO2.1 (1) AO2.2 (1)	Examiner's Comments Some candidates spent time defining decomposition which was not required. Many candidates scored some credit for their response, but there was often repetition in many responses. Fewer candidates could identify three or more distinct points.
			Total	4	
5			1 mark for sub-procedure names (to max 3) 1 for description (to max 3) e.g. • `startGame` [1] • loads the game, displaying the load-up screen[1] • `displayMap` [1] • outputs the map onto the screen[1] • `selectArea` [1] • the user clicks on the area of the map and this is displayed on screen[1] • `loadPuzzle` [1] • loads the puzzle for the chosen area and displays it on screen [1]	6 AO2.2 (6)	Allow any reasonable sub-procedure and description for the context Do not award marks for character movement. Examiner's Comment: Most candidates scored well, but some repeated items from the stem of the question and gave answers related to character movement which were not creditworthy.
			Total	6	

© OCR 2026. You may photocopy this page.

Created in ExamBuilder