ANSWERS

Answer 1

Question	Answer	Marks
8(a)	**1 mark** per bullet point to **max 4** • Reads/writes data to/from RAM • … e.g. current data/instructions from a game so the CPU can access it • Allocates virtual memory • … when there is insufficient RAM to run a program/game • Allocates RAM to optimise performance • Paging • Segmentation	4

Question	Answer	Marks
8(b)	**1 mark** per bullet point to **max 2** • Software will have been built using a compiler // the software is pre-compiled • Software is an executable file // the game is already in machine code // the game is already set-up to run on the console • Source code is not provided so does not need compiling/interpreting	2
8(c)	**1 mark** for each similarity, **max 2** • Both devices regulate network traffic between two networks // connect two networks • Both receive packets from a network and both forward packets onto a network **1 mark** for a difference • A Router connects two networks using the same protocol, a Gateway can connect two networks using different protocols	3

Answer 2

Question	Answer	Marks
9(a)	**1 mark** for each correctly identified utility program	3

Description	Utility program
Reorganises files on a disk to improve efficiency	**Defragmentation software**
Scans a hard disk to identify bad sectors	**Disk contents analysis / repair software**
Prepares a hard disk for first use	**Disk formatter**

Answer 3

Question	Answer	Marks
2(c)(i)	**1 mark** per bullet point to **max 3** • Disk contents analysis checks for errors/problems with the disk • Disk repair attempts to fix the errors • The disk formatter prepares the disk for (initial) use (again).	3

Question	Answer	Marks
2(c)(ii)	**1 mark** per bullet point to **max 3** • Installation of device driver software • Managing interrupts / signals from the device • Sending control signals to the device • Control of buffers • Management of queues	3

Answer 4

Question	Answer	Marks
4(a)(i)	**1 mark** per reason to **max 3** • DDL file is only loaded into memory when required • ... so the executable file for the game is smaller • Changes/improvements in the DLL file are independent of the main program • ... the game program will not need to be recompiled • ... the game program will get the benefit of the updates automatically • The same DDL file can be used in several game programs (at the same time) • (DLL) routines are pre-written saving the developers time • (DLL) routines are pre-tested so should be reliable • Developers can take advantage of other programmers' expertise	3
4(a)(ii)	**1 mark** per reason to **max 2** • Game will not work if DDL is corrupted • An external change to the DDL could stop the game working or change the way it works • The DDL file must be present at run-time otherwise there is an error	2

Question	Answer	Marks
4(b)	**1 mark** for interpreter **1 mark** per bullet point for justification to **max 2** • The interpreter translates and executes line by line • The interpreter stops translation when an error is found • ...the error can be corrected, and the program continues running from where it stopped • There is no need for an executable file (at this stage)	3

Answer 5

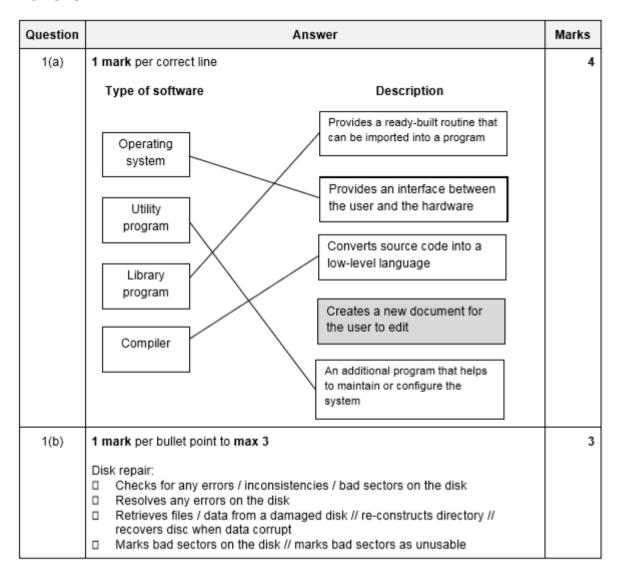| Question	Answer	Marks
3(a)(i)	**1 mark** for each advantage, **1 mark** for a valid expansion to **max 2 · 2** ☐ Code is already tested ☐ …so it is more robust/likely to work ☐ Saves programming time ☐ …code does not have to be written/re-written from scratch ☐ The programmer can use e.g. mathematical functions ☐ …that s/he may not know how to code ☐ If there is an improvement in the library routine ☐ …the program updates automatically	4
3(a)(ii)	**1 mark** per bullet point to **max 2** ☐ A collection of self-contained (shared library) programs ☐ …that are already compiled ☐ Linked to the main program during execution ☐ Library program code is separate from the .EXE file ☐ Library file only loaded into memory when required at run time ☐ A DLL file can be made available to several applications (at the same time) ☐ If DLL routine is updated the program that uses it will run the update	2
3(b)(i)	**1 mark** per bullet point to **max 1** ☐ Errors can be corrected as they occur ☐ Can run a partially complete program when developing ☐ The effect of any change made to the code can be seen immediately	1

Answer 6

Question	Answer	Marks
1(a)	**1 mark** for each correct entry	4

Management task	Description
Memory management	Handles the allocation of memory to processes // Ensures two programs do not attempt to use the same memory locations // Keeps track of allocated and free memory locations
Security management	**Provides user accounts and passwords**
Interrupt processing	**Handles the signals sent when the attention of the processor is required elsewhere**
Provision of a software platform	Provides an environment within which programs can be run

1(b)(i)	**1 mark** per bullet point to **max 2** for formatter, **max 2** for defragmenter	4
	hard disk formatter ☐ Makes existing data inaccessible ☐ Partitions the disk into logical drives ☐ Sets up the (specified) file system ☐ Prepares the disk for initial use ☐ May check for errors on the disk **hard disk defragmenter** ☐ Re-organises the disk contents ☐ Moves split files so they are contiguous ☐ Creates a larger area of (contiguous) free space	
1(b)(ii)	**1 mark** per bullet point For example: ☐ Backup software ☐ File compression ☐ Virus checker ☐ Disk contents analysis / repair	3

Answer 7

2(b)	**1 mark** per bullet point to **max 4** for each management task, **max 6** in total	6
	Process management: ☐ Manages the **scheduling** of processes ☐ … allows multi-tasking / multi-processing ☐ … ensures fair access ☐ … handles priorities ☐ Manages the resources the processes need ☐ Enables processes to share information ☐ Prevents interference between processes // resolution of conflicts Provision of a user interface: ☐ Allows a user to **communicate** with the hardware // vice-versa ☐ … by making navigation around the system easier ☐ Provides facility for user inputting data ☐ Provides facility for outputting to the user ☐ By example e.g. command line / GUI / menu-driven	

Question	Answer	Marks
2(c)	**1 mark** per bullet point to **max 3** for each utility program, **max 4** in total Virus checker: ☐ **Scans** files stored on a **computer system** for malicious code ☐ Scans files when they **enter the system** / memory stick inserted / download etc. ☐ Sets up a schedule for virus-checking ☐ Isolates / quarantines / deletes viruses ☐ Regularly updates the virus definitions	4

	Backup software:
	☐ Creates a copy of the contents of a disk / partition. Can be set up to automatically backup // schedules backups
	☐ Allows the user to decide what is backed up, e.g. all data // all files that have changed since the last backup
	☐ Allows the user to set up an off-site backup
	☐ May encrypt the backup files
	☐ Restores the data if necessary

2(d)	**1 mark** per bullet point to **max 2** ☐ The code is translated one line at a time ☐ ... and executed immediately ☐ The interpreter stops as soon as it finds an error	2

Answer 8

Question	Answer	Marks
1(a)	**1 mark** per correct line Type of software — Description Operating system Utility program Library program Compiler Provides a ready-built routine that can be imported into a program Provides an interface between the user and the hardware Converts source code into a low-level language Creates a new document for the user to edit An additional program that helps to maintain or configure the system	4
1(b)	**1 mark** per bullet point to **max 3** Disk repair: ☐ Checks for any errors / inconsistencies / bad sectors on the disk ☐ Resolves any errors on the disk ☐ Retrieves files / data from a damaged disk // re-constructs directory // recovers disc when data corrupt ☐ Marks bad sectors on the disk // marks bad sectors as unusable	3

Answer 9

Question	Answer	Marks
2(a)	**1 mark** per bullet point to **max 4** for each management task, **max 6** in total **Process Management** ☐ Manages the **scheduling** of processes ☐ ... allows multi-tasking / multi-processing ☐ ... ensures fair access ☐ ... handles priorities ☐ Manages which resources the processes require ☐ Enables processes to share information ☐ Prevents interference between processes // resolution of conflicts **Memory Management** ☐ Allocates memory to processes ☐ Ensures fair usage of memory ☐ Organises memory / by example ☐ Makes use of virtual memory ☐ Keep processes separate ☐ To release memory when a process stops	6

Answer 10

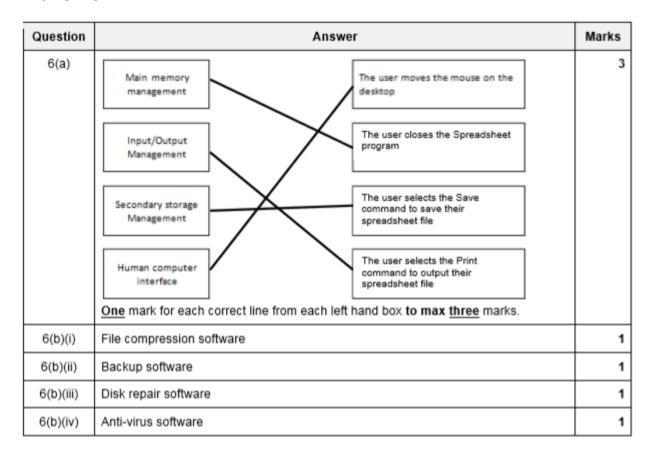| Question	Answer	Marks
3(a)(i)	**1 mark** per bullet to **max 3** ☐ Allocates / deallocates RAM to programs/tasks/processes ☐ Keeps track of allocated and free memory locations ☐ Swaps data to and from the hard drive ☐ Handles virtual memory ☐ Paging // segmentation ☐ Memory protection, preventing a process accessing memory not allocated to it	3
3(a)(ii)	**1 mark** per bullet to **max 3** ☐ Sets up user accounts ☐ Checks usernames, passwords // Authentication ☐ Implements access rights ☐ Automatic backup ☐ System restore / roll back (to previous stable state)	3
3(a)(iii)	**1 mark** per bullet to **max 2** ☐ Device / peripheral management ☐ File management ☐ Process management ☐ Input / output management ☐ Error detection / recovery ☐ Provides a user interface ☐ Facilitates communication between hardware and software / hardware devices	2

3(b)	**1 mark** for each correct box ticked			4

Programs	True	False
Disk Defragmenter	✓	
Word Processor		✓
Library program		✓
Compression Software	✓	

Answer 11

Question	Answer	Marks
1(a)(i)	**1 mark** per bullet to **max 3** ☐ Storage space divided into file allocation units ☐ Space allocated to particular files ☐ Maintains/creates directory structures ☐ Specifies the logical method of file storage (e.g. FAT or NTFS) ☐ Provides file naming conventions ☐ Controls access // implements access rights // implements password protection // Makes file sharing possible ☐ Specifies tasks that can be performed on a file (e.g. open, close, delete, copy, create, move etc.)	3
1(a)(ii)	**1 mark** per bullet to **max 3** ☐ Installs printer driver ☐ Sends data to the printer / buffer to print // sends documents to the print queue ☐ Sends commands to printer ☐ Receives and handles (error) messages/signals/interrupts from the printer	3
1(b)(i)	**1 mark** for each correct box ticked.	4

Program	True	False
Database		✓
Virus checker	✓	
Web browser		✓
Backup software	✓	

1(b)(ii)	**1 mark** for each valid utility program to **max 2** e.g. ☐ System clean up ☐ Automatic update ☐ Disk contents analysis / Disk checking / Disk repair ☐ File compression ☐ Disk formatter ☐ Firewall ☐ Disk Defragmenter	2

Answer 12

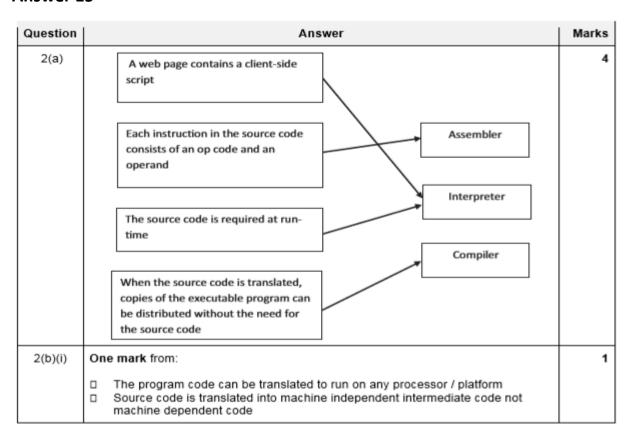| 6(b)(i)	**1 mark** per bullet to **max 2** • Pre-existing / pre-compiled / pre-written modules / code ... • ... can be linked into her program (without amendment) • To perform common / complex tasks	2

Question	Answer	Marks
6(b)(ii)	**1 mark** per bullet point. **Max 2** for <u>one benefit</u>, **max 2** for <u>one drawback</u> Benefit: • Less code needs to be written • … saves time / saves re-inventing the wheel • Pre-tested // Used by many people • … reduces time testing // can be fairly sure that the function will perform as it should • Can be written in a different programming language • … making use of special features of that language • Can be complex algorithms (e.g. mathematical/graphics functions) • … she does not need to work out how to write it // …that she may not know how to code • Simplifies the program • … since just the name of the function included in the source code Drawback: • Compatibility issues • … may not work with the other code/may require changing program for it to work • Not guaranteed thorough testing • … may be unknown or unexpected bugs / virus • Library routine may not meet exact needs • … may give unexpected results // … may need editing • If library routine is changed • … there may be unexpected results / errors	4

Question	Answer	Marks
6(c)	**1 mark** per bullet point. **Max 3** marks for interpreter, **max 3** marks for compiler Interpreter: • Used during development • Debugging is easier • … Because errors are reported as they are found // No need to wait until the end of the process for the error report • … Because errors can be corrected as they are found Compiler: • Compiler used when development complete // compiler used when program ready for distribution • Produces an executable file (.exe) • … After compilation the compiler does not need to be present for the program to run • … The program can be given to others without access to (source) code • … Final program does not need to be re-compiled each time it is run • Cross-compilation, the program can be compiled to run on different platforms	4

Answer 13

Question	Answer	Marks
6(a)	Main memory management — The user closes the Spreadsheet program Input/Output Management — The user selects the Print command to output their spreadsheet file Secondary storage Management — The user selects the Save command to save their spreadsheet file Human computer interface — The user moves the mouse on the desktop **One** mark for each correct line from each left hand box **to max three** marks.	3
6(b)(i)	File compression software	1
6(b)(ii)	Backup software	1
6(b)(iii)	Disk repair software	1
6(b)(iv)	Anti-virus software	1

Answer 14

4(a)(i)	**Two** from: ☐ The hardware is unusable without an OS // hides complexity of hardware from user 1 ☐ Acts as an interface/ controls communications between user and hardware / hardware and software // or by example 1 ☐ Provides software <u>platform / environment</u> on which other programs can be run 1	2
4(a)(ii)	**One** mark for the name and **one** mark for description. **Max two** management tasks. ☐ Provides the Human Computer Interface (HCI) 1 Controls communications between user and hardware// or by example 1 ☐ Main memory management 1 Memory protection to ensure that two programs do not try to use the same space // Use of virtual memory // Location of processes within the memory // By example 1 ☐ File / Secondary storage management 1 Maintains directory structures // Provides file naming conventions // Controls access 1	Max 4

	☐ Peripheral / hardware / device / Input-Output management 1 Installation of appropriate driver software // Controls access to data being sent to/from hardware/peripherals // Controls access to hardware/peripherals // manages communication between devices. 1 ☐ Interrupt handling 1 Identifies priorities of interrupts // Saves data on power outage // Loads appropriate Interrupt Service Routine (ISR) // By example 1 ☐ Security management 1 Makes provision for recovery when data is lost // Provides usernames and passwords // Prevents unauthorised access // Ensures privacy of data 1		
4(b)(i)	File compression software		1
4(b)(ii)	Backup software		1
4(b)(iii)	Disk defragmenting software		1
4(b)(iv)	Anti-virus software		1

Answer 15

Question	Answer	Marks
2(a)		4
2(b)(i)	**One mark** from: ☐ The program code can be translated to run on any processor / platform ☐ Source code is translated into machine independent intermediate code not machine dependent code	1

Question	Answer	Marks
2(b)(ii)	**Two marks** from: ☐ Java uses a two-step translation process ☐ Java code is partially interpreted – partially compiled ☐ Code is translated first into intermediate code / "bytecode"... ☐ ...using the Java compiler ☐ The bytecode is finally interpreted by the Java Virtual Machine	Max 2

Answer 16

1 **One mark** for each box on the left. [3]

Term		Definition

Compiler

Assembler

Interpreter

The software reads the source code and reports all errors. The software produces an executable file.

The software reads each statement and checks it before running it. The software halts when it encounters a syntax error.

The software translates a high level language program into machine code for the processor to execute

The software translates low-level statements into machine code for the processor to execute.

Marks allocated as follows:

Answer 17

1 **Four** from: [4]

- Compiler creates an executable//an interpreter does not create an executable.
- The compiled program can be independently distributed.
- Compiler reports all errors at the end of compilation//an interpreter stops when it reaches an error.
- Interpreter executes each statement immediately after decoding/checking it//a compiler checks the whole program for errors.
- The interpreter software/source code must be present in main memory every time the program is executed//the compiled program does not require compiler/source code to be present.
- Cross-compilation is possible/compile on one hardware platform to run on another.

Answer 18

(b) (i) Any **two** from

- The hardware is unusable without an OS // hides complexity of hardware from user

- Acts as an interface / controls communications between user and hardware / hardware and software

- Provides software platform / environment on which other programs can be run [2]

(ii) Any **two** from:

- Process / task / resource management

- Main memory management

- Peripheral / hardware / device management

- File / secondary storage management

- Security management

- Provision of a software platform / environment on which other programs can be run – only if not given in part (b)(i)

- Interrupt handling

- Provision of a user interface run – only if not given in part (b)(i) [2]

(c) Any **two** from:

- A DLL file is a shared library file

- Code is saved separately from the main .EXE files

- Code is only loaded into main memory when required at run-time

- The DDL file can be made available to several applications (at the same time) [2]

Answer 19

7 (a) **One mark** for the name and **one mark** for the explanation for **three** utility programs

- Disk formatter
- Prepares a hard disk to allow data to be stored on it

- Virus checker
- Checks for viruses <u>and then</u> quarantines removes any virus found

- File compression
- Reduces file size by removing redundant details (lossy / lossless)

- Backup software
- Makes copy of files on <u>another medium</u> in case of corruption / loss of data

- Firewall
- Prevents unauthorised access to computer system from external sources [6]

Answer 20

7   ONE mark per bullet point, **MAX TWO** marks per task.

- Process / resource management
- Scheduling of processes / multi-tasking / multi-programming etc.
- Resolution of conflicts when two or more processes require the same resource

- Main memory management
- Memory protection to ensure that two programs do not try to use the same space
- Use of virtual memory
- Deciding which processes need to be in main memory at any one time
- Location of processes within the memory
- By example, e.g. when process terminates, memory is made available

- Peripheral / hardware / device management
- Installation of appropriate driver software
- Controls access to data being sent to / from hardware / peripherals
- Controls access to hardware / peripherals
- Manages communication between devices / hardware and software

- File / secondary storage management
- Maintains directory structures
- Provides file naming conventions
- Controls access

- Security management
- Makes provision for recovery when data is lost
- Provides usernames and passwords / encryption / user accounts
- Prevents unauthorised access
- Ensures privacy of data

- Provision of a software platform / environment
- On which other programs can be run

- Interrupt handling
- Identifies priorities of interrupts
- Save current memory / process values / saves data on power outage
- Loads appropriate Interrupt Service Routine (ISR)
- Any relevant example

Answer 21

8   (a)   ONE mark for each bullet point from **MAX TWO** groups.

- The code is already written
- (So the programmer is not starting over again) which saves time

- The code will have been used by many people
- So it should be already thoroughly tested // relatively error-free

- The programmer can use, e.g. mathematical / graphics functions, etc. (may not know how to code)
- Can be sure that the function will perform as it should // simplifies the program.

- The code should conform to industry standards
- And therefore contribute towards a more robust program

[4]

  (b) (i)   ONE mark for each benefit, and ONE mark for a further expansion.

- The executable file is smaller / the executable does not contain all the library routines ...
- ... DLL files are only loaded into memory when required.

- Changes / improvements / error correction to the DLL file code are done independently of the main program...
- ... So there is no need to recompile the main program
- ... All programs using it will benefit

- A single DLL file can be made available to several application programs...
- ... Saving space in memory / easing the pressure on memory

[4]

    (ii)   ONE mark for each bullet point from **MAX ONE** group.

- The executable code is not self-contained ...
- ... the DLL file(s) needed to be included at run time.

- Appropriate (linking) software must be available at run-time ...
- ... to link / include / import the DLL files.

- The DLL file must be present ...
- ... otherwise (unable to find X.dll) errors

- Unexpected changes to the DLL file / corrupted DLL file ...
- ... could mean the program stops working as expected

- Malicious changes to the DLL file ...
- ... could install a virus on the user's computer / related files could be corrupted

[2]

Answer 22

Activity	First pass or second pass
any symbolic address is replaced by an absolute address	2
any directives are acted upon	1
any symbolic address is added to the symbolic address table	1
data items are converted into their binary equivalent	1
forward references are resolved	2

[5]

Answer 23

11

Statement	Interpreter	Compiler
This translator creates an executable file		✓
When this translator encounters a syntax error, game execution will halt	✓	
The translator analyses and checks each line just before executing it	✓	
This translator will produce faster execution of the game program		✓
Use of this translator makes it more difficult for the user to modify the code of the game supplied to the user		✓

1 mark for each correct row   [5]

Answer 24

(a) 1 mark per point

CLI

- user types in instructions to open/launch an application
- usually a number of instructions need to be typed in
- user is in direct communication with the computer system
- user has to type in the commands each time they want to open/launch the application

GUI

- user interacts with the system by using icons
- user doesn't need to know where application resides in the computer
- the application is opened/launched by clicking on an icon using a mouse (e.g.)
- windows is an example of GUI interface   [2]

(b) **CLI**

- programmer/technician } need to access system and communicate
- systems engineer } at system level

GUI

- end user } does not need computer knowledge
  } just click on the icon to launch the application   [2]

Answer 25

(a) (i) Any **two** points from:

- – obsolescence/out of date
- – specific examples e.g. floppy disk, mag tape etc.
- – not compatible with new equipment
- – key components no longer manufactured/spares are hard to find
- – software support no longer in existence/problems with maintenance   [2]

(ii) Any **two** points from:

- – uprating/updating of system (using parts which are outside normal specified range)
- – buying enough spare parts to meet system's forecasted lifetime requirements
- – part substitution (different parts with similar fit are used where possible)
- – redesign system to allow introduction of new components
- – emulation (parts with identical function and fit are made from new technologies)
- – aftermarket sources (third parties continue to make "obsolescent" parts)
- – training in-house programmers/maintenance personnel   [2]

Answer 26

(d) Any **two** points from:

- – OS will allow one user **at a time** to use the computer
- – each approved user is identified by a user id/password
- – allows multi-tasking
- – provides security for user files/profiles   [2]