QUESTION 1.

11 A game program is written which can be either interpreted or compiled. The table below shows five statements about the use of interpreters and compilers.

Tick (✓) to show whether the statement refers to an interpreter or to a compiler.

Statement	Interpreter	Compiler
This translator creates an executable file		
When this translator encounters a syntax error, game execution halts		
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		

[5]

(b) **(i)** A black and white image is 512 pixels by 256 pixels.

Calculate the file size of this image in kilobytes (KB) (1 KB = 1024 bytes).
Show your working.

.....

.....

.....

.....

.....[2]

(ii) Give a reason why it is important to estimate the file size of an image.

.....

.....

.....[1]

9 **(a)** Give a brief description of each of the following terms:

Validation .....

.....

.....

Verification .....

.....

.....[2]

(b) Data are to be transferred between two devices. Parity checks are carried out on the data.

Explain what is meant by a parity check. Give an example to illustrate your answer.

.....

.....

.....

.....

.....

.....

.....

.....[4]

QUESTION 2.

2 **(a)** The diagram shows three items of software that translate program code.

Draw **one** line from each context to the correct item of translation software.

Context	Item of translation soft
A web page contains a client-side script.	Assembler
Each instruction in the source code consists of an op code and an operand.	Interpreter
The source code is required at run-time.	Compiler
When the source code is translated, copies of the executable program can be distributed without the need for the source code.	

[4]

(b) The Java programming language is said to be machine or platform independent.

(i) Describe what is meant by **machine independent**.

.....

.....[1]

(ii) Describe how a Java source code program is translated.

.....

.....

.....

.....[2]

QUESTION 3.

2 **(a)** The diagram shows three items of software that translate program code.

Draw **one** line from each context to the correct item of translation software.

Context	Item of translation soft
The source code is written in a high-level language. An executable file is produced.	Assembler
The source code uses instructions from the processor's instruction set.	Interpreter
The source code and translation software must both be in main memory at execution time.	Compiler
A web page contains some JavaScript code.	

[4]

(b) A programmer is developing software and has both a compiler and interpreter for the high-level language used.

Describe **two** benefits of using each form of translation software.

(i) Benefits of a compiler

1 .....

.....

2 .....

.....

[2]

(ii) Benefits of an interpreter

1 .....

.....

2 .....

.....

[2]

QUESTION 4.

4 The table shows assembly language instructions for a processor which has one register, the Accumulator (ACC) and an index register (IX).

Instruction		Explanation
Op code	**Operand**	
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the Index Register. Copy the contents of this calculated address to ACC.
LDR	#n	Immediate addressing. Load the number n to IX.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
INC	<register>	Add 1 to the contents of the register (ACC or IX).
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX).
CMP	<address>	Compare contents of ACC with contents of <address>.
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to the screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

(a) (i) State what is meant by **direct addressing** and **indirect addressing**.

Direct addressing .....

.....

Indirect addressing .....

.....

[2]

(ii) Explain how the instruction ADD 20 can be interpreted as either direct or indirect addressing.

Direct addressing .....

.....

Indirect addressing .....

.....

[2]

(b) The assembly language instructions in the following table use either symbo
absolute addressing.

Tick (✓) **one** box in each row to indicate whether the instruction uses symbolic or
addressing.

Instruction	Symbolic	Absolute
`ADD 90`		
`CMP found`		
`STO 20`		

[2]

(c) The current contents of a general purpose register (X) are:

X	1	0	1	1	1	0	1	0

(i) The contents of X represent an unsigned binary integer.

Convert the value in X into denary.

.....[1]

(ii) The contents of X represent an unsigned binary integer.

Convert the value in X into hexadecimal.

.....[1]

(iii) The contents of X represent a two's complement binary integer.

Convert the value in X into denary.

.....[1]

(d) The current contents of the main memory, Index Register (IX) and selected ASCII character set are provided with a copy of the instruction set.

Address	Instruction
70	LDX 200
71	OUT
72	STO 203
73	LDD 204
74	INC ACC
75	STO 204
76	INC IX
77	LDX 200
78	CMP 203
79	JPN 81
80	OUT
81	LDD 204
82	CMP 205
83	JPN 74
84	END
…	
200	130
201	133
202	130
203	0
204	0
205	2

IX	0

ASCII code table (selected codes only)

ASCII code	Character
127	?
128	!
129	“
130	*
131	$
132	&
133	%
134	/

Instruction set

Instruction		Explanation
Op code	Operand	
LDD	<address>	Direct addressing. Load the contents of the location at the given address to ACC.
LDX	<address>	Indexed addressing. Form the address from <address> + the contents of the Index Register. Copy the contents of this calculated address to ACC.
LDR	#n	Immediate addressing. Load the number n to IX.
STO	<address>	Store contents of ACC at the given address.
ADD	<address>	Add the contents of the given address to ACC.
INC	<register>	Add 1 to the contents of the register (ACC or IX).
DEC	<register>	Subtract 1 from the contents of the register (ACC or IX).
CMP	<address>	Compare contents of ACC with contents of <address>.
JPE	<address>	Following a compare instruction, jump to <address> if the compare was True.
JPN	<address>	Following a compare instruction, jump to <address> if the compare was False.
JMP	<address>	Jump to the given address.
OUT		Output to the screen the character whose ASCII value is stored in ACC.
END		Return control to the operating system.

9

Complete the trace table for the given assembly language program.

Instruction address	ACC	Memory address						IX	C
		200	201	202	203	204	205		
70	130	130	133	130	0	0	2	0	

[8]