The Robert Napier School

Programming: Translators and Debugging

A	Translators vocab
Assembly language	A simple low-level language where opcodes are replaced with mnemonics and the instruction set is small (maybe 9 instructions)
Compiler	A program which turns source code into object code and saves it as an executable file
Editor	A program which allows the user to write code
GUI builder	An IDE for developing a graphical user interface
High-level (language)	A language which is easy to read and requires translating before the computer understands it
Instruction set	The full list of commands available within a language
Integrated Development Environment (IDE)	Software for writing code, which will usually incorporate an editor, debugging tools, an interpreter and compiler
Interpreter	A program which translates source code as it is read, stopping if it reaches an error
Linker	A tool which can combine different compiled codes
Low-level (language)	A language which is close to the format read by the computer
Machine code / Object code	Code written in binary
One-to-many	A language where one written instruction corresponds to a number of actions by the processor
One-to-one	A language where one written instruction corresponds to one action by the processor
Pretty printing	A feature of an editor which makes code easier to read by colouring and indenting
Runtime environment	Everything you need to run a program
Translation	Conversion of high-level language to machine code
Translator	A program which converts high-level language or assembly language to machine code

B	Command breakdown
Opcode	The part of the instruction which tells the CPU what operation is to be done
Operand	The part of the instruction which is to be operated on

C	A single command at different levels	
	Opcode	**Operand**
Machine code	0000 0001	0010 1110
Hex	01	2E
Assembly	ADD	2E
Python	+	num
Effect	adds	the value at 0010 1110 (named num)

D	Debugging
Trace table	An offline method of tracking the values of variables through the running of a procedure
Overflow error	An error produced when a number becomes longer than the number of bits allocated to it. The extra bits are lost.
Logic error	An error with code where it compiles correctly but produces incorrect results
Syntax error	An error with the code where the computer can not recognise it as code
Runtime error	An error which occurs during operation of the program, not during compilation

Questions
1) What is an interpreter?
2) What is the purpose of a trace table?
3) What is machine code?
4) What is a translator?
5) Give one example of a language that is high level code?
6) What does a compiler do?

The Robert Napier School

Programming: Essential Programs 1

A — Count from 1 to 20

	Python	Pseudocode	Main Differences

Condition controlled loop	`x = 1`
	`while x < 21:`
	`  print(x)`
	`  x = x + 1`
		`x = 1`
		`while x < 21`
		`  print(x)`
		`  x = x + 1`
		`endwhile`
		
		`x = 1`
		`do`
		`  print(x)`
		`  x = x + 1`
		`until x == 21`
			• Pseudocode has ENDWHILE
			• Pseudocode can use DO UNTIL

Count controlled loop	`for i in range(1, 21):`
	`  print(i)`
		`for i=1 to 20`
		`  print(i)`
		`next i`
			• Pseudocode FOR loop looks like this.
			• Must have NEXT i

B — One Question Quiz

Python	Pseudocode	Main Differences

`ans = input("5 x 3?")`
`if ans == "15":`
`  print("Yes")`
`elif ans == "16":`
`  print("Close")`
`else:`
`  print("No")`
	`ans = input("5 x 3?")`
	`if ans == "15" then`
	`  print("Yes")`
	`elseif ans == "16" then`
	`  print("Close")`
	`else`
	`  print("No")`
	`endif`
	
	`ans = input("5 x 3?")`
	`switch ans:`
	`  case "15":`
	`  print("Yes")`
	`  case "16":`
	`  print("Close")`
	`  default:`
	`  print("No")`
	`endswitch`
		THEN instead of colon
		ELSEIF instead of elif
		ENDIF at the end
		Indentation not necessary
		SWITCH CASE is not in Python

C — Output all the members of an array which are multiples of 3.

```
a = [2,3,5,8,13,21,34,55]
for x in a:
  if x % 3 == 0:
    print(x)
```

Makes use of modulo division – **x % 3** means **x MOD 3** which means **the remainder when x is divided by 3**. If the remainder is 0, there is no remainder. Which means that x is an exact multiple of 3. This program will output 3 and 21

Questions

1) What are the main difference between Python and Pseudocode?

2) What are the key difference between Python and Pseudocode when using a for loop?

3) What will the following code do? Unsure, run this in Python and see.

```
x = 1
while x < 21:
  print(x)
  x = x + 1
```