

Question 1.

Markup languages are computer languages used to define and annotate text elements in electronic documents.(Reference 1).

Computers can then interpret the markup languages to manipulate and be instructed on how the document should be displayed electronically - such as in web page format.. Examples of markup languages include HTML (Hypertext Markup Language) and, XML (Extensible Markup Language) (Reference 2)

A trait of markup languages is that they are easily readable, as opposed to having more complex syntax found in programming languages. (Reference 1)

A key feature of markup languages is the use of tags. Tags are used throughout a document to instruct the computer on the formatting of text elements within a document. (Reference 1). For example, in HTML, a page tag such as <body> is used to define the body section of a page. (Reference 3).

Question 2.

Packets

Packet switching is a procedure for splitting and sending data. Packets are small data segments created when a computer file is broken up. These packets are then transferred across a network before being reassembled back into a single file at the destination (Reference 4).

Originally, computers utilised a 'circuit switching' method. The downside of this is that a connection can only be made with one other computer at a time, requiring one single dedicated link. It would also take a longer time for the computer to connect to somewhere else (Reference 5).

Instead, packet switching allowed computers to send messages, called packets, along the same set of wires instead of each individual computer getting one. (Reference 5). Packet switching allowed for data to be reliably sent securely and use bandwidth efficiently, proving to be a better method of connection than the prior 'circuit switching' method (Reference 4).

IP addresses (IPv4 and IPv6)

An IP address is a unique address which identifies a device on a network (Source 6).

IP stands for Internet Protocol, which is a set of guidelines on how data is sent across a network. (Reference 7).

IPv4 is the fourth version of the IP and is currently the most widespread version today. (Reference 8). IPv6, version 6, was intended to be the next generation to replace the current widely used IPv4. (Reference 9)

IP addresses have allowed for a unique IP number to be assigned to devices connected to the internet, thus allowing for devices to accurately find the location of others and communicate effectively (Reference 4).

Routers and routing

A router is a network hardware device that connects and allows for communication between two or more networks. (Reference 7).

The method of selecting a path in any network is called routing. (Reference 10).

The use of routing has allowed for the minimisation of network failure, congestion and crashes by managing data traffic. Essentially, routing makes efficient network communication, thus being an important part in the development of the Internet (Reference 10).

Domains and DNS

A domain or domain name, is an easily remembered name that is associated with an IP address. Domain names are more easier for people to memorise and remember, as opposed to IP addresses that contain long numbers. (Reference 11).

DNS stands for Domain Name System, where domain names are converted by it to computer-readable IP addresses so that they can be accessed by web browsers (Reference 10).

The technology of DNS has allowed for users to type an easily-remembered domain anime into a web browser and then the DNS will then translate that request to reach the correct IP address (Reference 10).

Question 3.

TCP

TCP stands for Transmission Control Protocol. Devices are able to communicate across a network by utilising the TCP as a communication standard (Reference 12).

Key characteristics of TCP include (Reference 13 & 14):

1. Connection-oriented

- Devices are required to first establish a connection before they can send data

2. Stream Data Transfer

- Data is split into packets in order to be sent across a network

3. Reliability

- TCP tracks data to ensure that it is reliably delivered and received.

4. Flow control

- The TCP connection is managed to ensure that data flows efficiently and smoothly

5. Multiplexing

- Allowing connected device to have multiple connections open without conflict

6. Full duplex

- Provides for data transfer in both directions. Once connection is established, both devices can send and receive data.

Before TCP, users formatted their packets differently. This meant that even if different networks could be connected, there would arise issues due to different formatting of data transfer. TCP/IP (Transmission Control Protocol/Internet Protocol) was established as a guideline for how packets were formatted, thus setting a standard to follow for users and computers (Reference 5). This led to connections between networks to

become much easier and smoother, and these numbers of different networks connecting to each other led to the formation of the Internet (Reference 5).

HTTP and HTTPS

HTTP stands for Hyper Text Transfer Protocol. HTTPS is simply the more secure version of HTTP. (Reference 7)

HTTP is used to load web pages using hyperlinks (Reference 7). It is the foundation of data exchange on the World Wide Web (Reference 16).

In 1989, it was first proposed by Tim Berners-Lee the idea for hyperlinks to connect documents together. HTTP was then developed as a protocol for computers to communicate HTML documents across the internet (Reference 4). HTTP was integral in the formation of the World Wide Web, with the other 3 building blocks being HTML, a format to represent hyper text documents, a client, a web browser to display and edit these documents, and a server that gave access to the document (Reference 16).

A HTTP request is how clients (such as browsers) communicate and load websites. The typical HTTP request contains (Reference 7):

1. HTTP version type
2. A URL
3. An HTTP method
4. HTTP request headers
5. Optional HTTP body

Communications between clients and servers are completed by HTTP requests and HTTP responses (Reference 15):

1. A HTTP request is first sent by a client (such as a browser) to the web
2. The request is then received by a web server
3. The server runs an application to process the HTTP request
4. The server then outputs a HTTP response to the browser
5. The HTTP response is received by the client

Web browsers (requests, rendering and developer tools)

A web browser is a software application which people utilise to access the Internet (Reference 17).

Key features usually found in web browsers include the browser window itself (also known as the user interface), the address and navigation bar as well as being able to view features such as user history, settings and access various tools (Reference 17).

There are additional features including bookmarks bar that save pages as well as add-ons or extensions to supplement user browsing (Reference 17). Modern web browsers also come with developer tools, which are tools that do a range of things such as showing which assets the page has requested and how long the load time was (Reference 16).

In 1990, Tim Berners-Lee created software that could present HTML documents in an easily readable format. This is the origin of the web browser that we know today (Reference 7). Information is transferred and communicated using HTTP (Hypertext Transfer Protocol), where the data is written in HTML. Web browsers then read the HTML code in order to display the information accordingly (Reference 18). Web browsers will

send a request to the server to fetch a HTML document, and when the server outputs a response back to the web browser the web browser will render the information to be displayed accordingly (Reference 19).

Question 4.

Three data structures used in Python include lists, tuples and dictionaries.

List

A list is an ordered collection of items. Each item in the list has an order that uniquely identifies it. Items in a list are put in square brackets and are separated by commas (Reference 20). An example of a list can be seen below:

```
List = [Item 1, Item 2, Item 3]
```

Lists are mutable (can be changed) and allow duplicate values. Lists are used when you want to store multiple items in a single variable. You would also want to use lists in cases where items are to have a defined order (Reference 15).

Tuple

Tuples are also a data structure in Python that is used to store multiple items in a single variable (Reference 15).

Whilst items in a tuple are also ordered, and allow for duplicate values, tuples are immutable - meaning that they cannot be changed. You cannot change or edit any items in a tuple. (Reference 15).

A tuple can be identified using round brackets. See example below:

```
Tuple = ("Item 1", "Item 2", "Item 3")
```

A use case for a tuple would be where you would want to store information in a variable that you know will never be changed. For example, coordinates can be represented using tuples.

Dictionary

Dictionaries are a data structure that is used to store data values in key:value pairs. (Reference 15).

From Python version 3.7, dictionaries are ordered which means the items in this data structure have a defined order and can be referred to by using an index. (Reference 15).

Dictionaries are also mutable (changeable) and do not allow duplicate items. (Reference 15).

A key feature of dictionaries is the use of curly brackets as well as pairs of keys and values. See in example below:

```
dict = {  
    "Name": "Tony",
```

```
"Gender": "Male",  
"Ethnicity": "Vietnamese"  
}
```

A use case for a dictionary would be where you would want to store data values with defined keys.

Question 5.

An interpreter is a computer program that executes instructions written in a particular programming language (Reference 21).

A compiler is a program that transforms a source code into a more basic machine language that is more easily readable by a computer (Reference 21).

An interpreter translates code one line at a time. Interpreters convert code into machine code whilst a program is running.

Compilers on the other hand will transform source code into machine code first, before a program is run (Reference 22).

Key differences of interpreters are that:

1. Reads then executes code in a single process
2. They do not save machine code into storage
3. Interpreted code runs slower
4. They do not generate an output program
5. The program execution is performed line by line
6. Displays errors of each line one by one which can be useful for immediately identifying errors
7. Programming languages such as Python and Ruby use interpreters

Key differences of compilers are that:

1. Compilers scan and converts code that is written in a programming language into machine code first before a program is executed
2. Generates intermediate machine code
3. Compiled code runs faster
4. Compilers display all errors after compilation as opposed to line by line
5. They generate an output program (in the form of an .exe). This output program can then be executed independently from the original program
6. The output program is unable to be changed without returning back to the source code
7. Languages such as C and C++ use compilers

Question 6.

Two commonly used programming languages are JavaScript and Python.

JAVASCRIPT

There are various advantages for the use of JavaScript. The first being the popularity of the language. According to the developer survey conducted on Stack Overflow, JavaScript is currently the most commonly

used programming language (Reference 23). Given that JavaScript is so widely used, it is a very useful language to learn. Here are other various advantages of JavaScript (Reference 24 & 15):

1. **Speed** - As JavaScript is executed on the client side, it tends to be very fast
2. **Simplicity** - JavaScript's syntax is easy to learn as opposed to languages with harder to grasp syntaxes such as C++.
3. **Versatility** - It is possible to develop an entire JavaScript app with the frontend and backend using JavaScript
4. **Server Load** - As JavaScript is client-side, it reduces the server load
5. **Regular Updates** - There are regular updates for JavaScript by ECMA
6. **Rich interface** - JavaScript can be used to make various engaging features which enhance the user interface and experience

A disadvantage of JavaScript is the issue of client-side security, as the JavaScript code is visible to everyone which can be possibly exploited for malicious purposes. Another disadvantage is that JavaScript only supports single inheritance and is unable to have multiple inheritance. Different browsers also interpret JavaScript differently which means programmers have to test the code on various platforms, which can be a disadvantage (Reference 24).

PYTHON

See below various advantages of the Python programming language (Reference 25 & 26):

1. **Simplicity** - Python is relatively simply to learn and implement, in comparison to other programming languages like C++ with harder to understand syntax
2. **Productivity** - Python's simplicity allows for developers to spend less time learning the language's syntax, write less code and instead focus more on the task at hand
3. **Libraries** - There are extensive libraries of python code which means that developers can utilize these libraries without having to write it out from scratch.
4. **Debugging** - Python facilitates debugging as the code is executed line by line, meaning that errors that occur are immediately shown and the program is terminated
5. **Open-Source and Free** - Python is free to use and distribute as it is under an open-source license recognised by the OSI.

Python's disadvantages include that it is slow to execute. This is because Python is an interpreted language, where Python code is executed line by line. Where speed is necessary, Python isn't best suited (Reference 25). Another disadvantage of Python is that it's not optimal for client-side development or in mobile applications due to Python using a large amount of memory and having a slow processing capability.

Question 7.

Here are the two ethical issues I have chosen to discuss:

1. **Access to a user's personal information (medical, family, financial, personal attributes such as sexuality, religion, or beliefs)**

Under the Privacy Act 1988, businesses are responsible for protecting customer personal information. Under the privacy act, businesses may be required to protect customer personal information from:

1. Theft

2. Misuse
3. Interference
4. Loss
5. Unauthorised access
6. Modification
7. Disclosure

When customer information is no longer required, it is necessary that their information is then eliminated or de-identified (Reference 30).

Australian privacy law also states that an organization can use or disclose personal information for the reason they collected it. However, information can't be used or disclosed for an outside reason unless there has been customer consent, or it is required to by law or a legal body (Reference 29).

As an IT professional, you are responsible for ensuring that customer personal information is handled ethically and can refer to the law for assistance in acting in a legal manner. For example, developers should ensure that there is a process in place for destroying customer data once it is no longer needed so that there is no further access to the data. Another example would be ensuring the customer data is only used in accordance to what the customer agreed and consented to, as using customer data for other activities could prove to be illegal and unethical.

2. Aggressive sales and marketing practices designed to mislead and deceive consumers

According to the Australian Competition & Consumer Commission (ACCC), it is illegal for businesses to engage in activity that is likely to mislead or deceive consumers or other businesses. This law still applies in cases where there was no intention to mislead or deceive, including if no person or party suffered any loss or damage as a result of the misconduct (Reference 31).

This is another ethical issue where IT professionals are responsible in the undertaking of these business practices. Whilst upon initial thought it would seem that the sales and marketing teams in businesses are solely responsible for this issue, IT developers also have a part to play.

IT professionals can refer to the law in this instance to ensure that the software and programs that are created are ran and displayed ethically and legally, and do not mislead or deceive any customers. For example, marketing teams work closely with developers to make updates to websites for things such as campaigns. In this instance, developers are responsible in ensuring that items displayed on the business website and apps do not mislead or deceive customers as this would breach the law and result in business misconduct.

CASE STUDY ON ETHICAL ISSUE OF ACCESS TO CUSTOMER INFORMATION

Recently, the insurance company Medibank has encountered an issue regarding the breach of customer data. It was confirmed by Medibank that the personal information of up to 10 million Medibank customers was released online by hackers (Reference 32).

On December 1, the Office of the Australian Information Commissioner (OAIC) stated that an investigation has been undertaken to analyse whether Medibank had taken the necessary steps to protect customer personal information from any misuse, being breached, misconduct or unauthorised access. (Reference 32). The potential ramifications are that Medibank potentially faces a large fine if the investigation finds that Medibank had any negligence, unethical activity or breached privacy law whilst handling customer data.

This issue that has occurred with Medibank certainly affects IT professionals and is an opportunity for IT professionals to refer to in acting ethically in order to avoid such a catastrophe.

According to the OAIC, a breach of data happens when customer personal or sensitive information has been accessed or disclosed without any authorisation. Under the Privacy Act 1988, organisations must notify the affected individuals and any required legal bodies, when a data breach involving sensitive information is likely to cause serious harm (Reference 29).

Following these principles outlined by the OAIC, IT professionals can refer to the law to identify the occurrence and severity of any data breaches. It is the responsibility of IT professionals to analyse what data had been breached and how much of an impact the data breach has on any individuals involved. This is so that all affected parties are properly notified and informed if there has been an instance where their sensitive information has been compromised. Managers will also be able to make more informed decisions once they have received the full information on what occurred during the instance of a data breach.

These are the steps that IT professionals should follow in the instance of a data breach, however it is of course most important that any data breaches were avoided in the first place. From the Medibank case study, there are serious ramifications for employees, businesses and any responsible parties when an instance of a data breach has occurred.

IT professionals should certainly take all necessary precautionary steps to mitigate or prevent breaches of customer personal information.

One such step would be ensuring that the cyber-security protocols in place are stringent and constantly updated and monitored. This would ensure that IT professionals are following the latest and best practices for cyber-security in order to prevent any outside parties meaning to cause harm from accessing company and customer private information.

Another step would be ensuring that there are very tight restrictions on who can access certain types of data within an organization. When IT professionals take this into consideration during their planning and activities, they can mitigate the chances of sensitive data falling into the wrong hands. With stringent protocols in place regarding the access of customer personal information, there can be proper management and monitoring of the data and persons who have access to the data.

Overall, IT professionals are responsible for ensuring they follow the most up to date legal and ethical practices in the handling of customer personal information. Any mishandling or breaches of data can have serious ramifications, as shown in the Medibank case study. Whilst having this in mind, IT professionals can ensure they have systems and protocols in place to mitigate or prevent ethical breaches of customers' sensitive information.

Question 8.

Control flow is the order in which computers execute or evaluate individual statements, instructions or function calls. In Python, the control flow is managed by loops, conditional statements and function calls (Reference 27 and 16).

Sequential, selection and repetition are the three types of control structures used in Python (Reference 28).

The execution of a sequential statement occurs in a sequence. See example below:


```
a = 5
b = 10
c = 5 + 10

print(f"Sum of numbers is equal to {c}")
```

The flow of the program above goes from top to bottom, in sequential order.

A selection statement is where the program will test any conditions and will execute the instructions depending on which condition is met.

See example using a Python if-else statement below:

```
n = 8
if n % 2 == 0:
    print ("n is even")
else:
    print ("n is odd")
```

The output of this program depends on if a condition is satisfied in the if-else statement.

Loops are used in repetition statements in Python. An example of this is a while loop where the statements are executed as long as a condition is true.

This can be displayed in the example below where the variable a is printed as long as it is less than 6:

```
a = 5
While a < 6:
    Print (a)
    a += 1
```

You can also see in the above examples the use of comparison operators. These comparison operators decide the relationship between the operands and thus control the flow of the program. Here are some examples of comparison operators:

\geq *This means greater or equal to*

\neq *This means not equals to*

$==$ *This means equals to*

Depending on how each of the comparison operators are satisfied, will the flow of the program be run.

Question 9.

Type coercion is the conversion of a value from data type to another. Type conversion is similar to this however type conversion can either be explicit or implicit whereas type coercion is always implicit (Reference 16).

There isn't type coercion in Python however there is type conversion in Python.

An example of type conversion in python is the use of the `str()` function to convert an integer to a string.

See example below where the integer 8 is converted to a string data type :

```
a = 8
b = str(a)

print(b)
```

Question 10.

Variables can store values in different data types. These data types define data in different types of categories.

Here are some examples of various data types used in Python:

1. The string data type where text is stored to a variable

```
a = "Hello"
```

2. In integer data type where a number is stored in a variable

```
a = 10
```

3. A float data type where a number with a decimal point is stored in a variable

```
a = 1.4
```

4. A list data type where a number of data points are stored in an ordered format.

```
a = ["Chicken", "Beef", "Eggs"]
```

5. Dictionary data type is where data values are stored as key:value pairs:

```
dict = {
    "Name": "Jeffrey",
    "Gender": "Male",
```

```
"Ethnicity": "Sudanese"  
}
```

5. A Boolean is where data is stored as a true or false value:

```
a = True
```

Question 11.

To solve this restaurant issue, there should be three classes created for food, customers and staff.

You want to define a class for customers so that you set assigned values for customers. Values to assign customers would be things such as name, age, gender and food preference. As each customer will have differing properties i.e. their food order, it will be good to set a class for customers in order to create objects representing customers to help solve the restaurant problem. An example of a customer class is shown below:

```
class Customer:  
    def __init__(self, name, age, gender, foodorder)
```

Another class to define would be for food to be served by the restaurant. The reason you would create a class for the food is because the restaurant serves a variety of food, meaning each food would be different to another. For example, values to assign to the food class would be things such as cost and food type (such as savoury & sweet or dessert & main course or just different types of foods such as burgers, fries and hotdogs). In solving the problem, the restaurant would need to consider the food that they're serving to customers on the menu, as well as things such as the cost of making the food. An example of a food class is shown below:

```
class Food:  
    def __init__(self, cost, type)
```

The last class to help solve this issue would be for staff. The reason is because different types of staff are required in order to upkeep the restaurant. An example would be chefs, waiters and managers - where their occupations and roles are different to one another. It would be good to define a class for staff and assign variables such as name, age, occupation and salary. This way, the objects created in application representing the staff, will have different properties that the restaurant will need to consider. An example of the staff class is shown below:

```
class Staff:  
    def __init__(self, name, age, occupation, salary)
```

The classes for Staff, Food and Customers will be able to be used to define the blueprint variables of the objects created (which represent different types of staff, food and customers) that can help the restaurant in

solving their problem.

Question 12.

The error is due to unsupported operand type(s) for /: 'str' and 'int'

It occurs we are multiplying the celsius with a number, which results in an error as a number has datatype of int but the variable celcius is in a string data type.

We need to convert the celsius variable to an integer by passing it's input into the int function.

Solution:

Line 1 of the program needs to be changed to the below code instead in order to work:

```
celsius = int(input())
```

Question 13.

```
arr = [5,22,29,39,19,51,78,96,84]

i = 0

while (i<len(arr)-1) and (arr[i] < arr[i+1]):

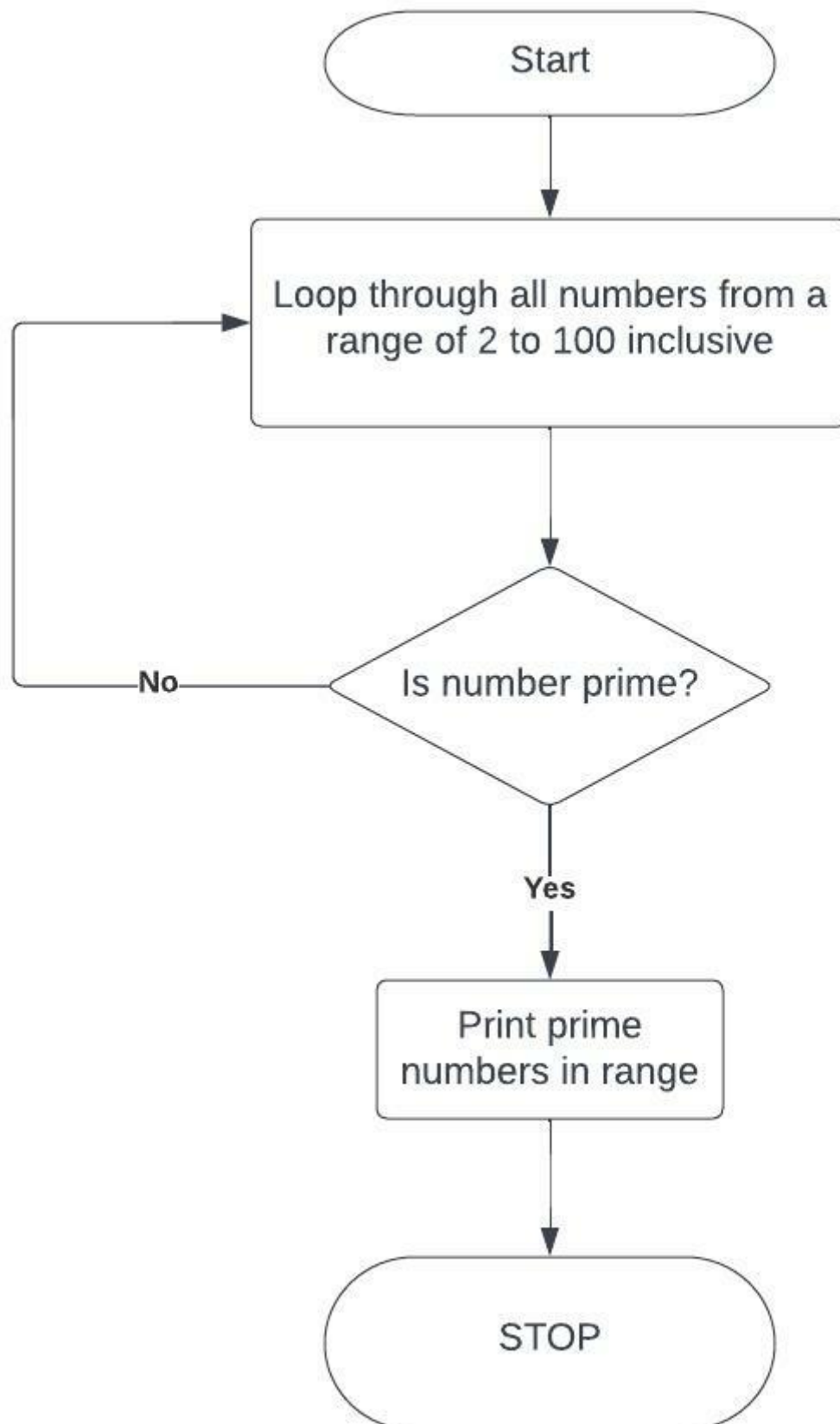
    print(i)

    arr[i] = arr[i+1]
    arr[i+1] = arr[i]

    i += 1
```

Question 14.

Flowchart



Pseudocode

```
# Loop through numbers in range of 2 to 101  
# Check if number is not prime
```

```
# Else print number  
# Display result
```

Code to solve the question

```
for num in range(2,101):  
    for i in range(2,num):  
        if (num % i) == 0:  
            break  
    else:  
        print(num)
```

Question 15.

In the code below, the variables "raining" and "temperature" can be changed to achieve the desired outcome.

```
raining = True  
temperature = 14  
  
if raining and temperature < 15:  
    print("It's wet and cold")  
  
elif not raining and temperature < 15:  
    print("It's not raining but cold")  
  
elif not raining and temperature >= 15:  
    print("It's warm but not raining")  
  
else:  
    print("It's warm and raining")
```

Question 16.

```
skills_weights = {  
    "python":1,  
    "ruby":2,  
    "bash":4,  
    "git":8,  
    "html":16,  
    "tdd":32,  
    "css":64,  
    "javascript":124}  
  
# (a)  
user_skills = []  
  
while True:
```

```
skill = input("Please enter your coding skill: ").lower()

user_skills.append(skill)

upskill = input("Do you want to learn any more skills? enter y or n: ")

if upskill != "y":

    break

score = 0

for skill in user_skills:

    score = score + skills_weights[skill]

print(f"Your coding score is {score}.")

# (b)

learning_skills = []

while True:

    skill = input("Please enter the coding skill you want to learn: ").lower()

    learning_skills.append(skill)

    upskill = input("Do you want to add any more skills to learn? enter y or n: ")

    if upskill != "y":

        break

score = 0

for skill in learning_skills:

    score = score + skills_weights[skill]

print(f"You will get a coding score of {score} after learning these skills.")
```

REFERENCES

- (1) [Amelia](#)
- (2) [Britannica](#)

- (3) [TechTerms](#)
- (4) [Science and Museum](#)
- (5) [SciShow](#)
- (6) [Kaspersky](#)
- (7) [Cloud Flare](#)
- (8) [Blue Cat Networks](#)
- (9) [Thousand Eyes](#)
- (10) [aws Amazon](#)
- (11) [Google Support](#)
- (12) [Fortinet](#)
- (13) [OmniSecu](#)
- (14) [TC/IP Guide](#)
- (15) [W3schools](#)
- (16) [Developer Mozilla](#)
- (17) [Avast](#)
- (18) [Mozilla](#)
- (19) [Medium](#)
- (20) [Corporate Financial Institute](#)
- (21) [Techopedia](#)
- (22) [Guru 99](#)
- (23) [Stack Overflow](#)
- (24) [freeCodeCamp](#)
- (25) [Python Plain English](#)
- (26) [Data Flair](#)
- (27) [Python Cheat Sheet](#)
- (28) [Educative](#)
- (29) [OAIC](#)
- (30) [Business Gov](#)

(31) [ACCC](#)

(32) [Sydney Morning Herald](#)