Question 1

# ALGORITHM

1.⁠ ⁠START

2.⁠ ⁠RECEIVE ORDER

3.⁠ ⁠ASK FOR ADD ONS

4.⁠ ⁠IF (add-on)

THEN WRITE DOWN AND INCREASE BILL ELSE

CONFIRM THE ORDER

5.⁠ ⁠RECEIVE PAYMENT

6.⁠ ⁠PASS THE ORDER TO THE CHEF

7. AFTER THE CHEF COOKS IT SERVE IT

8.⁠ ⁠END

# PSEUDOCODE

1.⁠ ⁠START

2.⁠ ⁠INPUT THE OREDER

3.⁠ ⁠INPUT THE BILL

4.⁠ ⁠IF (add-on)

PRINT = ORDER + ADD ON IN THE BILL

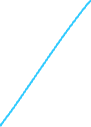
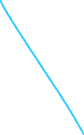
5.⁠ ⁠ELSE

PRINT ORDER IN THE BILL

6. END

# FLOW CHART

START



ORDER = BILL

BILL = ORDER + ADD ON

YES

NO

INPUT

ORDER

ADD ON

PROCESS

TAKE THE OREDER

OUTPUT

PRINT BILL



END



QUESTION 4

# ALGORITHM

1. START

2. INPUT NUMBER

3. READ THE INPUT

4. CHECK THE NUMBER

IF

NUMBER IS 1 THEN DISPLAY JANUARY

NUMER IS 2 THEN DISPLAY FEBRUARY

NUMBER IS 3DISPALY MARCH

THE NUMBER 4 DISPLAYED APRIL

THE NUMBER IS 5 THEN DISPLAY MAY

NUMBER 6 THEN DISPLAY JUNE

NUMBER IS 7 DISPLAY JULY

NUMBER IS 8 DISPLAY AUGUST

NUMBER 9 DISPLAY SEPTEMBER

NUMBER IS 10 DISPLAY OCTOBER

NUMBER IS 11 DISPLAY NOVEMBER

NUMBER IS 12 DISPLAY DECEMBER

ELSE

NUMBER IS NOT BEWTEEN 1-12 THEN PRINT ERROR 5. END

QUESTION 5

# PSEUDOCODE

1. START

2. INPUT NUMBER 1

3. INPUT NUMBER 2

4. CHECK THE OPERATION

IF

OPERATION IS + THEN ADD NUMBER 1 AND NUMBER 2

OPERATION IS - THEN SUBTRACT NUMBER 1 AND BUMBER 2

ELSE

DISPLAY ERROR

6. PRINT

7. END

QUESTION 7

1. START

2. INPUT NUMBER 1

3. INPUT NUMBER 2 \

4. READ THE OPERATION

IF

OPERATION IS + THEN ADD BOTH NUMBERS

OPERATION IS - THEN SUBTRACT BOTH NUMBERS

OPERATION IS \* THEN MULTIPLY BOTH NUMBERS

OPERATION IS / THEN DIVIDE BOTH NUMBERS

OPERATION IS % THEN TAKE THE MODULUS OF NUMBERS

ELSE

DISPLAY INVALID OPERATION

5. END

QUESTION 2

# ALGORITHM

1. START

2. RECEIVE INFORMATION ABOUT THE ACCOUNT

3. SEARCH THE ACCOUNT

4. IF

ACCOUNT IS ACTIVE PROCEED

ELSE

REJECT

5. CHECK FOR ACCOUNT VALIDITY

6. CHECK FOR DEPOSITORY CONDITIONS

IF

THE AMOUNT IS GREATER THAN 0 PROCEED

ELSE

REJECT

6. ENTER THE DEPOSITION AMOUNT

7. COLLECT THE DEPOSITION AMOUNT

8. PRINT DEPOSITED

9. GIVE THE DEPOSITION SLIP TO CUSTOMER

10. END

# PSEUDOCODE

1. START

2. INPUT ACCOUNT NUMER

3. INPUT DEPOSITION ACCOUNT

4. IF

ACCOUNT IS ACTIVE AND THE AMOUNT IS GREATER THAN 0 THEN COLLECT THE

DEPOSITORY AMOUNT

5. PRINT DEPOSITED

6. END

QUESTION 3

# ALGORITHM

1. START

2. INPUT NUMBER 1

3. INPUT NUMBER 2

4. INPUT NUMBER 3

5. COMPARE NUMBER 1 AND NUMBER 2

6. COMPARE NUMBER 3 WITH THE ANWSR OF STEP 5

7. IF

NUMBER 3 IS THE GREATEST PRINT NUMBER 3

ELSE

PRINT THE ANSWER OF STEP 5

8. END

QUESTION 2

# ALGORITHM

IS A RECIPE OF A PROGRAM OR A GUIDE THAT THOROUGHLY EXPLAINS A

PROGRAM.WHAT TO DO AND IN WHAT ORDER.

IT HAS NO SYNTAX.

WHEREAS A

# PSEUDOCODE

IS A WAY TO WRITE DOWN THE ALGORITHM IN AN EASY AND UNDERSTANDABLE

LANGUAGE. A LITTLE SIMILAR TO A CODE BUT NOT A CODE EXACTLY. IT ALSO HAS NO

SYNTAX AND IS ALSO NOT CONSIDERED A LANGUAGE.

QUESTION 9

GITIGNORE BASICALLY TELLS THE GITHUB REPOSITORY TO EXCLUDE

OR IGNORE CERTAIN FILES FOR VARIOUS REASONS, LIKE SECURITY, AS WE WANT THESE FILES TO REMAIN UNTRACKED