***ASSIGNMENT NO.1***

***NAME: WALEED AHMED***

***ROLL NO: 25K-2517***

***FIELD: BSDS (DATA SCIENCE)***

***SECTION: 1A***

***{PART A}***

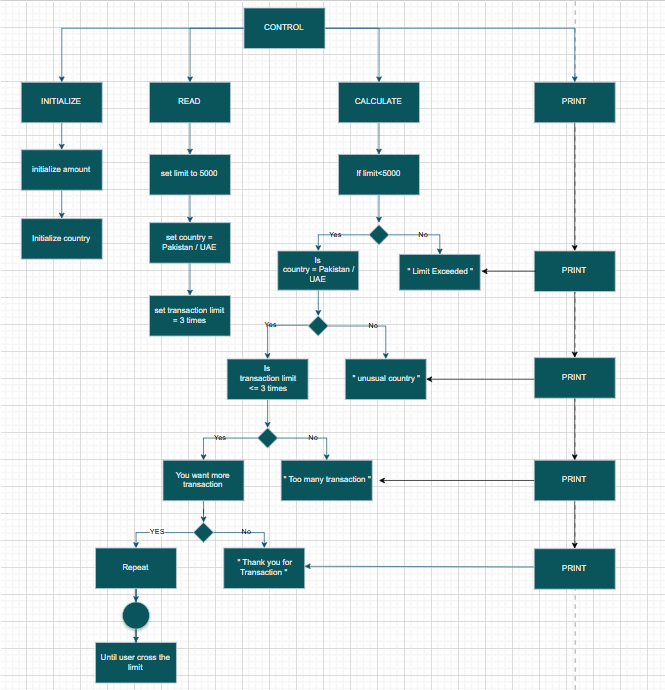
***PAC [Question no.1]***

|  |  |
| --- | --- |
| **GIVEN DATA** | **REQUIRED RESULT** |
| * Daily spending limit = 5000 * Total spent = 0 * Transaction amount * Transaction country * Transaction time (hour) | * Identify if a transaction should be: * Flagged (Limit Exceeded) * Marked Suspicious (Unusual Country) * Marked Fraud (Too many transactions in same hour) |
| **PROCESSING REQUIRED** | **SOLUTION ALTERNATIVE** |
| Add each transaction amount to total spent • Check if total > daily limit • Check if country is unusual • Count transactions per hour • If >3 in one hour → fraud | * Use IF conditions to evaluate rules: * IF total spent > 5000 → Flag * IF country not in usual list =Suspicious * IF transaction count in hour > 3 = Fraud Detected |

***IPO***

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **PROCESSING** | **MODULE** | **OUTPUT** |
| * Read Country | * Check if   country =Pakistan or UAE | |  | | --- | | * Country Validation |  |  | | --- | |  | | * If country != Pakistan or UAE, Display “unusual country” |
| * Read transaction amount | * Check if transaction amount >limit | |  | | --- | | * Amount Validation |  |  | | --- | |  | | * If amount>limit   Display “Amount Exceeded” |
| * Read (y/n) for more transaction | * Track number of transactions in hour   (transaction in hours>3) | |  | | --- | | * Transaction Monitoring |  |  | | --- | |  | | * If transaction in hour >3   Display “too many transaction occurs” |
| * Ask for more transaction | |  | | --- | | * User Interaction |  |  | | --- | |  | | If user press ‘n’ Display “ Thank you” |

***IC***

******

***Algorithm***

Step 1: START

Step 2: Set a fixed daily limit.

step 3: initialize total spent =0

Step 4: For each transaction add amount to total spent.

Step 5: If customer spends more than daily limit, it is considered unusual.

Step 6: display “Daily spending limit exceeded”

Step 7: If a transition occurs is foreign country, which is not usual, the system should mark it as suspicious.

Step 8: Display “Unusual Country”

Step 9: If many transitions occur in very short period, system considers it as fraud.

Step 10: If more than 3 transactions occur in same hour .

Step 11: Display “Too many transactions in very short time”

Step 12: END

***PSEUDOCODE***

START

SET daily Limit to 5000

SET total Spent to 0

SET transaction Count In Hour to 0

SET current Hour to -1

FOR each transaction DO

ADD transaction amounts to total Spent

IF total Spent > daily Limit THEN

PRINT "Amount should be flagged"

ENDIF

IF transaction occurs in unusual foreign country, THEN

PRINT "Unusual country"

ENDIF

IF transaction hour == current Hour THEN

Transaction Count In Hour = transaction Count In Hour + 1

ELSE

Current Hour = transaction hour

Transaction Count In Hour = 1

ENDIF

IF transaction Count In Hour > 3 THEN

PRINT "Too many transactions"

ENDIF

END FOR

END

***FLOWCHART***

A

A

C

YES

YES

Is

Transaction occurs in foreign unusual country

NO

Print "Unusual Country"

Print "Daily Limit Exceeded”

Is

Total spent >Daily limit

Daily limit to 5000,

total spent = 0

START

A

Print "Fraud"

YES

A

END

Print "No Fraud”

NO

C

NO

Is

Transaction occurs in hour > 3

***C CODE***

#include <stdio.h>

int main() {

    float limit = 5000;

    float transaction;

    char country[15];

    char answer;

    int transactionLimit = 1;

    printf("Enter country:\n");

    scanf("%s", country);

    if ((country[0] == 'P'|| country[0] =='p') && country[1] == 'a' && country[2] == 'k' &&

        country[3] == 'i' && country[4] == 's' && country[5] == 't' &&

        country[6] == 'a' && country[7] == 'n' && country[8] == '\0') {

    }

    else if ((country[0] == 'U' || country[0] == 'u') && country[1] == 'a' && country[2] == 'e' && country[3] == '\0') {

    }

    else {

        printf("Unusual Country\n");

        return 0;

    }

    printf("Enter your transaction amount:\n");

    scanf("%f", &transaction);

    if (transaction > limit) {

        printf("Limit Exceeded!\n");

        return 0;

    }

    printf("Do you want more transactions? (y/n):\n");

    scanf(" %c", &answer);

    if (answer == 'y' || answer == 'Y') {

        printf("Enter your transaction amount:\n");

        scanf("%f", &transaction);

        if (transaction > limit) {

            printf("Limit Exceeded!\n");

            return 0;

        }

        printf("Do you want more transactions? (y/n):\n");

        scanf(" %c", &answer);

        if (answer == 'y' || answer == 'Y') {

            printf("Enter your transaction amount:\n");

            scanf("%f", &transaction);

            if (transaction > limit) {

                printf("Limit Exceeded!\n");

                return 0;

            }

            printf("Do you want more transactions? (y/n):\n");

            scanf(" %c", &answer);

            if (answer == 'y' || answer == 'Y') {

                printf("Too many transactions in the same hour\n");

                return 0;

            } else {

                printf("Thank you\n");

                return 0;

            }

        } else {

            printf("Thank you\n");

            return 0;

        }

    } else {

        printf("Thank you\n");

        return 0;

    }

    return 0;

}

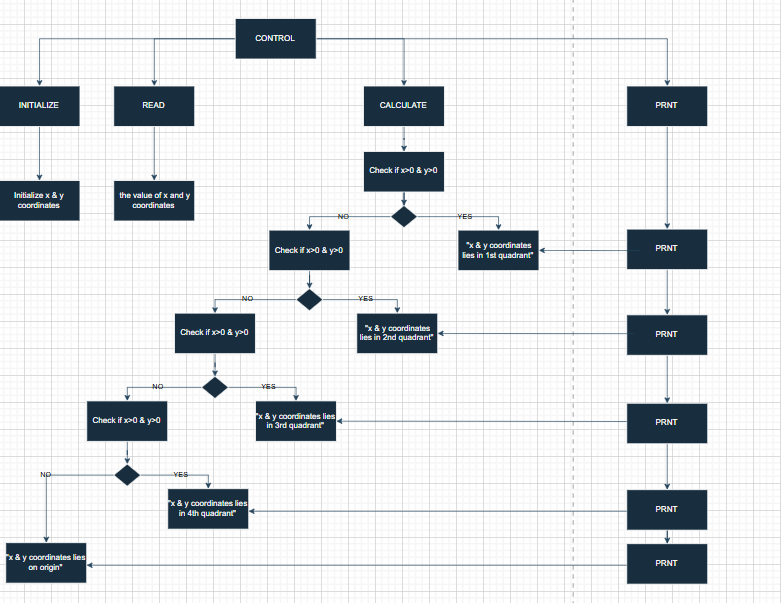
***PAC [Question no.2]***

|  |  |
| --- | --- |
| **GIVEN DATA** | **REQUIRED RESULT** |
| * x coordinate. * y coordinate. * Condition of quadrants | * Which quadrant does the point lie in (I, II, III, IV) |
| **PROCESSING REQUIRED** | **SOLUTION ALTERNATIVE** |
| * Take input values of ‘x’ and ‘y’. | * Check the point by plotting graph. |
| * Compare sign of ‘x’ and ‘y’ coordinates. | * Check the quadrant by measuring angles. |
| * Use conditional statements to find correct quadrant. | * Check the quadrants by using switch statement. |

***IPO***

|  |  |  |  |
| --- | --- | --- | --- |
| ***INPUT*** | ***PROCEESING*** | ***MODULE*** | ***OUTPUT*** |
| * x coordinate | * read value of x and y | * Input Collection | * Display: “The point (x, y) lies in quadrant I or II or III or IV” |
| * y coordinate | * check sign of coordinates | |  | | --- | | * Sign Evaluation |  |  | | --- | |  | |
| * Use conditional statement to determine quadrant | |  | | --- | | * Quadrant Decision |  |  | | --- | |  | |

***IC***

******

***Algorithm***

**Step** 1: Start  
**Step** 2: enter the values for x and y  
**Step** 3: if both x and y are positive, then the point is in Quadrant 1  
**Step** 4: if y is positive and x is negative, then the point is in Quadrant 2  
**Step** 5: if both x and y are negative, then the point is in Quadrant 3  
**Step** 6: if x is positive and y is negative, then the point is in Quadrant 4  
**Step** 7: End

***PSEUDOCODE***

* START
* READ x, y
* IF x > 0 AND y > 0 THEN

PRINT "The point is in Quadrant 1"

* ELSE IF y > 0 AND x < 0 THEN

PRINT "The point is in Quadrant 2"

* ELSE IF x < 0 AND y < 0 THEN

PRINT "The point is in Quadrant 3"

* ELSE x > 0 AND y < 0 THEN

PRINT "The point is in Quadrant 4"

* END

***FLOWCHART***

START

**READ**

**x, y**

A

**PRINT (x, y)**

**“Is in Quadrant 1”**

**IS x>0, y>0**

A

**PRINT (x, y)**

**“Is in Quadrant 4”**

NO

END

A

**PRINT (x, y)**

**“Is in Quadrant 3”**

YES

NO

**IS x<0, y<0**

A

**PRINT (x, y)**

**“Is in Quadrant 2”**

YES

**IS x<0, y>0**

NO

YES

***CODE***

#include<stdio.h>

int main () {

    int x , y;

    printf("Enter value of x ");

    scanf("%d",&x);

    printf("Enter value of y ");

    scanf("%d",&y);

    if (x>0 && y>0)

    {

        printf("The point (%d, %d) lies in quadrant I", x, y);

    } else if (x<0 && y>0)

    {

                printf("The point (%d, %d) lies in quadrant II", x, y);

    } else if (x<0 && y<0)

    {

                printf("The point (%d, %d) lies in quadrant III", x, y);

    } else if (x>0 && y<0)

    {

                printf("The point (%d, %d) lies in quadrant IV", x, y);

    }else{

printf("The point (%d, %d) lies on axis", x, y);

}

    return 0;

}

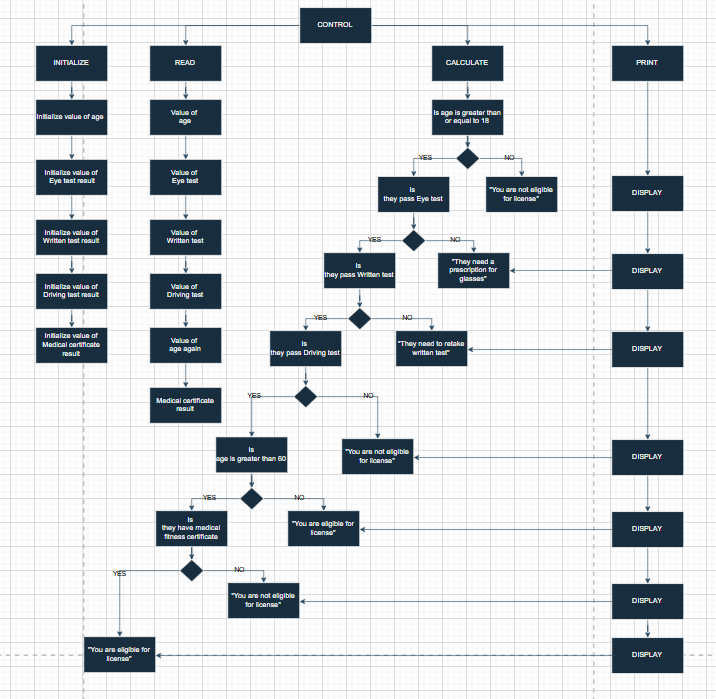
***PAC [Question no.3]***

|  |  |
| --- | --- |
| **GIVEN DATA** | **REQUIRED RESULT** |
| * Age * Eye test result (‘P’ or ’F’) * Written test result (‘P’ or ’F’) * Driving test (‘P’ or ’F’) * Medical fitness certificate | * Eligible for license * Not eligible for license * Need a prescription of glasses * Need to retake written test |
| **PROCESSING REQUIRED** | **ALTERNATIVE SOLUTION** |
| * Check applicant’s age >=18. * Check if they passed or failed eye test * Check if they passed or failed written test. * Check if they passed or failed driving test. * if applicant’s age >=60 ,Check the medical certificate. | * Using by decision table * Using by conditional statement * Using C code * Using a Flow chart * Using an Algorithm |

***IPO***

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **PROCESSING** | **MODULE** | **OUTPUT** |
|  | * If age>=18, continue | |  | | --- | | * Age Verification |  |  | | --- | |  | | * Eligible for license |
| * Eye test result | * If eye test= ‘P’ continue, else   Recommended glasses. | |  | | --- | | * Vision Test |  |  | | --- | |  | | * Need prescription for glasses. |
| * Written test result | * If written test= ‘P'   Continue, else  Recommended  Retaking the test | * Written Test | * Need to retake the test. |
| * Driving test | * If driving test= ‘p’   Check age>=60 | |  | | --- | | * Driving Skill Test |  |  | | --- | |  | | * Not eligible for license. |
| * Medical certificate | * If age >60, check medical certificate if ‘Y’, eligible, Else not eligible. | |  | | --- | | * Medical Evaluation |  |  | | --- | |  | |

**IC**

******

***ALGORITHM***

Step 1: Start

Step 2: Ask the applicant for their age.

Step 3: If the age is 18 or above, ask the next question.

Step4: Else display “the applicant is not eligible”.

Step 5: If applicant passed the eye test, continue with next question.

Step 6: Else display “they need a prescription for glasses”.

Step 7: If the applicant passed written test, then continue with next question.

Step 8: Else display “they need to retake the written test”.

Step 9: If the applicant passes the driving test, then continue with next question.

Step 10: else display you are not eligible.

Step 11: if applicant’s age is below 60.

Step 12: Display “they are eligible for the license”.

Step 13: else ask they have medical fitness certificate or not

Step 14: if they have, inform they are eligible for license

Step 15: else they are not eligible for license

Step 16: end

***PSEUDOCODE***

* START
* READ age

IF age >= 18 THEN

* READ eye Test Result

IF eye Test Result == 'P' THEN

* READ written Test Result

IF written Test Result == 'P' THEN

* READ driving Test Result

IF driving Test Result == 'P' THEN

IF age < 60 THEN

PRINT "You are eligible for the license."

ELSE

* READ has Medical Certificate

IF has Medical Certificate == 'Y' THEN

PRINT "You are eligible for the license."

ELSE

PRINT "You are not eligible for the license."

END IF

END IF

ELSE

PRINT "You are not eligible for the license."

END IF

ELSE

PRINT "You need to retake the written test."

END IF

ELSE

PRINT "You need a prescription for glasses."

END IF

ELSE

PRINT "The applicant is not eligible."

END IF

* END

***FLOWCHART***

START

READ AGE

Print “Applicant not eligible

NO

IS AGE>=18

B

B

B

**A**

NO

Print “they need to retake the written test”

IS Writtentest==”p”

YES

Print “they need a prescription for glasses”

NO

IS Eyetest==”p”

YES

B

STOP

YES

B

Print “they are not eligible for the license”

B

B

B

Print “You are not eligible”

Print “they are eligible for the license”

**A**

NO

YES

IS medical fitness certificate==”yes”

NO

Print “they are eligible for the license”

YES

IS AGE<=60

YES

NO

IS Drivingtest==”p”

YES

***CODE***

#include <stdio.h>

int main() {

    int age;

    char eyeTest, writtenTest, drivingTest, medicalCertificate;

    printf("Enter your age: ");

    scanf("%d", &age);

    if (age >= 18) {

        printf("Have you passed the eye test? (P/F): ");

        scanf(" %c", &eyeTest);

        if (eyeTest == 'P' || eyeTest == 'p') {

            printf("Have you passed the written test? (P/F): ");

            scanf(" %c", &writtenTest);

            if (writtenTest == 'P' || writtenTest == 'p') {

                printf("Have you passed the driving test? (P/F): ");

                scanf(" %c", &drivingTest);

                if (drivingTest == 'P' || drivingTest == 'p') {

                    if (age > 60) {

                        printf("Do you have a medical fitness certificate? (Y/N): ");

                        scanf(" %c", &medicalCertificate);

                        if (medicalCertificate == 'Y' || medicalCertificate == 'y') {

                            printf("You are eligible for the license.\n");

                        } else {

                            printf("You are not eligible for the license.\n");

                        }

                    } else {

                        printf("You are eligible for the license.\n");

                    }

                } else {

                    printf("You are not eligible for the license.\n");

                }

            } else {

                printf("You need to retake the written test.\n");

            }

        } else {

            printf("You need a prescription for glasses.\n");

        }

    } else {

        printf("The applicant is not eligible.\n");

    }

    return 0;

}

***PAC***

|  |  |
| --- | --- |
| **GIVEN DATA** | **REQUIRED DATA** |
| * 5 card numbers * Each card number is from 1 -13. * 1st represent Ace * 2-10 represents number * 11 represent jack * 12 represent queen * 13 represent king | * Determine “full house” if in 5 cards , any 3 cards are of same rank and 2 cards of other rank * Is it full house * Is it not a full house |
| **PRCESSING REQUIRED** | **SOLUTION ALTERNATIVE** |
| * Read 5 card numbers from the user. * Initialize an array to count occurrences of each card number. * For each card, increase the count of that card's number in the count array. * Check the count array to find:   If any card number appears exactly 3 times, and  If any other card number appears exactly 2 times. | * By using conditional statement * By using while loop |

***IPO***

|  |  |  |  |
| --- | --- | --- | --- |
| **INPUT** | **PROCESS** | **MODULE** | **OUTPUT** |
| * 5 card numbers | * Each card number is from 1 -13. | |  | | --- | | * Input Collection |  |  | | --- | |  | | * If it full fill condition print ”full house” |
| * Each card number is from 1 -13. | * Initialize an array, increase the count of that card's number in the count array | * Validation | * Display “not a full house” |
| \_\_\_\_\_\_ | * If any card appear =3   Or other appear=2 | |  | | --- | | * Frequency Count |  |  | | --- | |  | | * Display “full house” |
| -------- | * ---- | |  | | --- | | * Full House Check |  |  | | --- | |  | | ---- |

***IC***

***A diagram of a work flow

AI-generated content may be incorrect.***

***ALGORITHM***

**Step 1:** Start  
**Step 2:** Create an array to store 5 card numbers  
**Step 3:** Ask the user to enter 5 card numbers (between 1 to 13)  
**Step 4:** Create another array to count how many times each number appears  
**Step 5:**  For each card in the array,  
   Increase the count of that card in the counting array by 1  
**Step 6:** Check if any number appears exactly 3 times, and another number appears exactly 2 times.  
**Step 7:** If yes, display "Full House"  
**Step 8:** Otherwise, display "Not a Full House"  
**Step 9:** End

***PSEUDOCODE***

START

DECLARE cards[5]

DECLARE num[14] ← 0

DECLARE i, three ← 0, two ← 0

DISPLAY "Enter 5 card numbers (1 to 13):"

FOR i ← 0 TO 4 DO

READ cards[i]

num[cards[i]] ← num[cards[i]] + 1

END FOR

FOR i ← 1 TO 13 DO

IF num[i] = 3 THEN

three ← 1

ELSE IF num[i] = 2 THEN

two ← 1

END IF

END FOR

IF three = 1 AND two = 1 THEN

DISPLAY "Full House"

ELSE

DISPLAY "Not a Full House"

END IF

END

Input Card

START

***FLOWCHART***

Read: 5 card numbers into card

A

No

Display “card doesn’t exsist”

A

Display “Not a Full House”

No

**A**

Any 3 cards equal?

(C = C = C)

yes

IS card number is in (1-13)

A

Display “Full House”

YES

A

END

Display “Not a Full House”

NO

Is other cards equal to each other but not to previous?

C=C

**A**

yes

***­­­CODE***

#include <stdio.h>

int main() {

    int C1,C2,C3,C4,C5;

    printf("Enter value of card : \n");

    scanf("%d %d %d %d %d",&C1,&C2,&C3,&C4,&C5);

    if ((C1 < 1 || C1 > 13) || (C2 < 1 || C2 > 13) || (C3 < 1 || C3 > 13) ||

        (C4 < 1 || C4 > 13) || (C5 < 1 || C5 > 13))

    {

        printf(" card number doesnot exsist!");

    }else{

        if ((C1 == C2 && C2 == C3 && C4 == C5) ||

            (C1 == C2 && C2 == C4 && C3 == C5) ||

            (C1 == C2 && C2 == C5 && C3 == C4) ||

            (C1 == C3 && C3 == C4 && C2 == C5) ||

            (C1 == C3 && C3 == C5 && C2 == C4) ||

            (C1 == C4 && C4 == C5 && C2 == C3) ||

            (C2 == C3 && C3 == C4 && C1 == C5) ||

            (C2 == C3 && C3 == C5 && C1 == C4) ||

            (C2 == C4 && C4 == C5 && C1 == C3) ||

            (C3 == C4 && C4 == C5 && C1 == C2)) {

                 printf("house full");

            }

        else{

            printf("House Not Full");

        }

    }

    return 0;

}

***{PART B}***

***CODE(Q5)***

#include <stdio.h>

int main() {

    int counts[10] = {0};

    int number;

    while (1) {

        printf("Enter a one digit number (0-9): \n");

        int result = scanf("%d", &number);

        if (result != 1 || number < 0 || number > 9) {

            printf("program is stopped due to invalid input.\n");

            break;

        }

        counts[number]++;

    }

    printf("\nNumber :  Number of Occurrences\n");

    for (int i = 0; i < 10; i++) {

        if (counts[i] > 0) {

            printf("%-9d %d\n", i, counts[i]);

        }

    }

    return 0;

}

***CODE(Q6)***

#include<stdio.h>

int main(){

    int i,j,k;

    printf("Enter value of i,j & k \n");

    scanf("%d %d %d",&i,&j,&k);

    if (i < j) {

    if (j < k)

        i = j;

}

else {

    j = k;

    if (j > k)

        j = i;

    else

        i = k;

}

printf("%d %d %d\n", i, j, k);

    return 0;

}

***Dry run***

**For (a)**

**Step 1:** Check value of i = 3, j = 5, and k = 7  
**Step 2:** The condition i < j, 3 < 5 is true, so we enter the first if block.  
**Step 3:** Inside if block, we check j < k, 5 < 7, which is also true.  
**Step 4:** So, we execute i = j, i = 5  
**Step 5:** Now we skip the else block because the first condition was true.  
**Step 6: Final values:**

* i = 5, j = 5, k = 7
* Output

5 5 7

**For (b)**

Step 1: Check value of i = 3, j = 7, and k = 5  
Step 2: The condition i < j, 3 < 7 is true, so we enter the first if block.  
Step 3: Inside if block, we check j < k, 7 < 5, which is false.  
Step 4: So, we skip i = j  
Step 5: Now we skip the else block because the first condition was true.  
Step 6: Final values:

* i = 3, j = 7, k = 5
* OUTPUT:

3 7 5

**For (c)**

Step 1: Check value of i = 5, j = 3, and k = 7  
Step 2: The condition i < j, 5 < 3 is false, so we enter the else block.  
Step 3: Inside else block, we assign j = k, so j = 7  
Step 4: Now we check j > k, 7 > 7, which is false  
Step 5: So, we execute i = k, i = 7  
Step 6: Final values:

* i = 7, j = 7, k = 7

Output:

7 7 7

**For (d)**

Step 1: Check value of i = 5, j = 7, and k = 3  
Step 2: The condition i < j, 5 < 7 is true, so we enter the first if block.  
Step 3: Inside if block, we check j < k, 7 < 3, which is false.  
Step 4: So, we skip i = j  
Step 5: Now we skip the else block because the first condition was true.  
Step 6: Final values:

* i = 5, j = 7, k = 3

Output:

5 7 3

**For (e)**

Step 1: Check value of i = 7, j = 3, and k = 5

Step 2: The condition i < j, 7 < 3 is false, so we enter the else block

Step 3: Inside else block, we assign j = k, so j = 5

Step 4: Now we check j > k, 5 > 5, which is false

Step 5: So, we execute i = k, which means i = 5

Step 6: Final values:

* i = 5, j = 5, k = 5

Output:

5 5 5

**For (f)**

Step 1: Check value of i = 7, j = 5, and k = 3  
Step 2: The condition i < j, 7 < 5 is false, so we enter the else block.  
Step 3: Inside else block, we assign j = k, so j = 3  
Step 4: Now we check j > k, 3 > 3, which is false  
Step 5: So, we execute i = k, i = 3  
Step 6: Final values:

* i = 3, j = 3, k = 3

Output:

3 3 3

***CODE(Q7)***

***{Switch Case}***

#include<stdio.h>

int main(){

    int day;

    printf("Enter a value in the range(1-7) to see the corresponding day name \n");

    scanf("%d",&day);

    switch (day)

    {

    case 1:

        printf("The day is Monday.");

        break;

    case 2:

        printf("The day is Tuesday.");

        break;

    case 3:

        printf("The day is Wednesday.");

        break;

    case 4:

        printf("The day is Thursday.");

        break;

    case 5:

        printf("The day is Friday.");

        break;

    case 6:

        printf("The day is Saturday.");

        break;

    case 7:

        printf("The day is Sunday.");

        break;

    default:

        printf("Not Found! Please enter in range (1-7).");

        break;

    }

    return 0;

}

***{If-else Case}***

#include <stdio.h>

int main() {

int day;

printf("Please enter a number between 1 and 7 to know the day of the week: ");

scanf("%d", &day);

if (day == 1) {

printf(" it's Monday.\n");

} else if (day == 2) {

printf(" it's Tuesday.\n");

} else if (day == 3) {

printf(" it's Wednesday.\n");

} else if (day == 4) {

printf(" it's Thursday.\n");

} else if (day == 5) {

printf(" it's Friday.\n");

} else if (day == 6) {

printf(" it's Saturday.\n");

} else if (day == 7) {

printf(" it's Sunday.\n");

} else {

printf(" %d is not a valid input. Please enter a number between 1 and 7.\n", day);

}

return 0;

}