

- C programming word problems (50)

1→ WAP calculating the total cost of items in a shopping cart. if the total cost exceeds \$100, the user gets a 10% discount. calculate final amount to be paid.

Sol→

Step 1 START

Step 2 Input a positive integer into a variable named total\_items

Step 3 total\_cost = 0

Step 4 item\_cost = 0  
Step 5 Repeat steps (a)-(b) while total\_items > 0

(a) Input a positive number into a variable named item\_cost

(b) total\_cost += item\_cost

END WHILE

Step 6 if total\_cost > 100 THEN

total\_cost = total\_cost -  $\left( \frac{10 * \text{total\_cost}}{100} \right)$   
END IF

Step 7 DISPLAY total\_cost

Step 8 END

2→ A teacher needs to calculate the average score of 5 students in a class. WAP that takes 5 scores as input and outputs the average score.

Sol→

Step 1 START

Step 2 AVERAGE = 0

Step 3 student\_score = 0

Step 4 students = 5

Step 5 Repeat steps (a)-(b) while students > 0

(a) Input a non negative integer into student\_score.

(b) AVERAGE = AVERAGE + student\_score

END WHILE

Step 6 AVERAGE = AVERAGE / students ~~\_\_\_\_\_~~

Step 7 DISPLAY AVERAGE

Step 8 END

③ WAP that asks user to input a temperature in Celsius and converts it into Fahrenheit using the formula :  $F = (9/5)*C + 32$ .

Sol→

Step 1 START

Step 2 Fahrenheit = 0

Step 3 Celsius = 0

Step 4 Input a number <sup>(temperature in °C)</sup> into the variable Celsius.

Step 5 Fahrenheit =  $\left( \frac{9}{5} * \text{Celsius} + 32 \right)$

Step 6 Display Fahrenheit

Step 7 END

④ A company wants to calculate the annual salary of the employee given their monthly salary. If the employee has been with the company for more than 5 years, they get a bonus of 10% on their annual salary. WAP to calculate final salary.

Sol→

Step 1 START

Step 2 Monthly-salary = 0

Step 3 experience = 0

Step 4 BONUS = 10

Step 5 Input <sup>a</sup> the salary of employee (positive number) into the variable monthly-salary

Step 6 Input the experience (<sup>non negative</sup> positive number) into variable experience.

Step 7 Annual-salary =  $12 * \text{monthly-salary}$

Step 8 IF experience > 5 THEN

Annual-salary = Annual-salary +  $\left( \frac{\text{Bonus} * \text{Annual-salary}}{100} \right)$

END IF

Step 9 DISPLAY Annual-salary

Step 10 END

⑤ WAP which takes user's age as input and print out whether they are eligible to vote (age 18 or older) or not.

Sol→

Step1 START

Step2 Input age of user (positive integer) into variable Age.

Step3 IF Age  $\geq 18$  THEN

DISPLAY "Eligible to vote"

ELSE

DISPLAY "not Eligible to vote"

END IF

Step4 END

⑥ WAP that calculates the <sup>(simple interest)</sup> SI on a savings account - take principal amount, interest rate, no. of years as input.

Sol→

Step1 START

Step2 Input principal amount (positive Number) into variable p

Step3 Input interest rate (positive number) into variable r

Step4 Input no. of years (positive numbers) into variable t

$$Step5. \quad SI = (p * r * t) / 100$$

Step6 DISPLAY SI

Step7 END

⑦ A shopkeeper offers a discount of 5% on purchases above \$200. WAP to calculate the final amount the customer has to pay after discount.

Sol→

Step1 START

Step2 discount = 5

Step3 Input amount to be paid without discount (positive number) into variable amount

Step 4 IF amount > 200 THEN

$$\text{amount} = \text{amount} - [\text{amount} * (\text{discount}/100)]$$

Step 5 END IF

Step 6 DISPLAY amount

Step 7 END

⑧ Write a C program to convert given number of days into years, months and days.

Sols

Step 1 START

Step 2 Input no. of days (non-negative) into variable days

Step 3 YEARS = days / 365

Step 4 days = days % 365

Step 5 MONTHS = days / 30

Step 6 days = days % 30

Step 7 DISPLAY YEARS

Step 8 DISPLAY ~~MONTHS~~ MONTHS

Step 9 DISPLAY days

Step 10 END

⑨ A bakery sells loaves of bread for \$3 each. Write a program to calculate total cost of purchasing a given no. of loaves.

Sol →

Step 1 start

Step 2 Cost-bread = 3

Step 3 Input total no. of loaves (non-negative) into variable total\_number

Step 4 total\_cost = Cost-bread \* number

Step 5 DISPLAY total\_cost

Step 6 END

(10) WAP to find max of 3 numbers entered by user.

Sol→

STEP1 START

STEP2 Input 3 numbers into variables A, B and C

STEP3 IF  $A \geq B$  and  $A \geq C$  THEN

DISPLAY A

ELSE IF  $B \geq A$  and  $B \geq C$  THEN

DISPLAY B

ELSE

ENDIF DISPLAY C

STEP4 END

(11) WAP to determine if a year is leap or not.

Sol→

STEP1 START

STEP2 Input the year <sup>from user</sup> (~~year~~ must be positive integer) into var year

STEP3 IF  $year \% 4 == 0$  and  $(year \% 100 != 0 \text{ || } year \% 400 == 0)$   
DISPLAY "leap"

ELSE

DISPLAY "not leap"

ENDIF

STEP4 END

(12) WAP that asks user to input a number & prints whether number is +ve, -ve or 0.

Sol→

STEP1 START

STEP2 Input a number in variable num

STEP3 IF num > 0 THEN

DISPLAY "POSITIVE"

ELSE IF num < 0 THEN

DISPLAY "negative"

ELSE DISPLAY "zero"

ENDIF

STEP4 END

(13) WAP to calculate the area of rectangle given its length and ~~width~~ breadth.

Sol-

STP

STEP1 START

STEP2 INPUT length and breadth of rectangle in variables L and B respectively.

STEP3  $A = L * B$

STEP4 DISPLAY A

STEP5 END

(14) Write a program to convert Km to miles

Sol-

STEP1 START

STEP2 INPUT distance in Km (positive number) in a variable DK

STEP3  $DM = DK * 1.609344$

STEP4 DISPLAY DM

STEP5 END

(15) WAP to check if character entered by user is a vowel or a consonant.

Sol-

STEP1 START

STEP2 Input character from user (a-z or A-Z) into variable ch

STEP3 IF  $ch == 'a'$  OR  $ch == 'e'$  OR  $ch == 'i'$  OR  $ch == 'o'$  OR  
 $ch == 'u'$  OR  $ch == 'A'$  OR  $ch == 'E'$  OR  $ch == 'I'$  OR  $ch == 'O'$  OR  
 $ch == 'U'$  THEN

DISPLAY "vowel"

ELSE

DISPLAY "consonant"

ENDIF

STEP4 END

(16) WAP to calculate factorial of a no. using loop

Sol→

STEP1 START

STEP2 Input a number (<sup>positive</sup> number integer or 0) in variable num

STEP3 factorial = 1

STEP4 Repeat steps (a)-(b) while num > 0

(a) factorial = factorial \* num

(b) num = num - 1

~~END WHILE~~

STEP5 DISPLAY factorial

STEP6 END

(17) WAP to find sum of no. between 1 to 100

Sol→

STEP1 START

STEP2 ~~sum~~ SUM = 0

STEP3 COUNT = 1

STEP4 Repeat steps ~~(a)-(b)~~ (a)-(b) while COUNT <= 100

(a) SUM = SUM + COUNT

~~END WHILE~~ (b) COUNT = COUNT + 1

STEP5 DISPLAY SUM

STEP6 END

(18) WAP to print multiplication table for a no. entered by user

Sol→

STEP1 START

STEP2 Input a number from user in variable num

STEP3 MAX = 10

STEP4 Repeat step4 COUNT = 0

STEP5 Repeat step5 <sup>until</sup> COUNT <= MAX

(a) DISPLAY num \* COUNT

(b) COUNT = COUNT + 1

STEP6 END

(19) WAP to calculate total marks and the percentage of 5 subjects entered by the user and determines the grade based on percentage.

Sol→

```

STEP1 START
STEP2 i = 0
STEP3 percentage → Total_Marks = 0
STEP4 subject = 5
STEP5 Temp_marks = 0
STEP6 REPEAT steps (a)-(c) while subjects > i
    (a) Input marks of a subject in variable Temp_marks
    (b) Total_Marks = Total_Marks + Temp_marks
    (c) i = i + 1
STEP7 Percentage = Total_Marks / subjects.
STEP8 IF percentage > 90 THEN
    DISPLAY 'A grade'
    ELSE IF percentage > 80 THEN
        DISPLAY 'B grade'
    ELSE IF percentage > 70 THEN
        DISPLAY 'C grade'
    ELSE IF percentage > 60 THEN
        DISPLAY 'D grade'
    ELSE
        DISPLAY 'F grade'
END IF
STEP9 DISPLAY percentage
STEP10 DISPLAY Grade At Total_Marks
STEP11 END

```

(20) WAP to reverse the digits entered by the user.→

Sol→

```

STEP1 START
STEP2 Input a number entered by the user in variable n

```

STEP3 IF  $n < 0$  THEN

IS-NEG = 1

$n = n * (-1)$

ELSE

IS-NEG = 0

STEP4 NEW-NUM = 0

STEP5 Repeat steps (a)-(c) while  $n > 0$

(a) NEW-NUM = NEW-NUM \* 10

(b) NEW-NUM = NEW-NUM +  $n \% 10$

(c)  $n = n / 10$

END while

STEP6 IF IS-NEG == 1 THEN

DISPLAY ( $-$  NEW-NUM)

ELSE

DISPLAY NEW-NUM

ENDIF

END

Q1 WAP to find GCD using Euclid's Algorithms.

Sol→

STEP1 START

(positive integers)

STEP2 Input two numbers in variables A and B

STEP3 temp = 0

STEP4 Repeat steps (a)-(c) while  $B \neq 0$

(a) temp = B

(b)  $B = A \% B$

(c)  $A = \text{temp}$

STEP5 DISPLAY A

STEP6 END

Q2 WAP to check if given number is a prime no.

Sol-

STEP1 START  
 STEP2 INPUT a number (positive) in variable  $n$   
 STEP3 IF  $n \leq 1$  THEN  
     DISPLAY "can't be prime"  
     (GOTO Step 7)  
     END IF  
 STEP4  $i = 2$   
 STEP5 Repeat steps (a)-(b) while  $i \leq n$   
     (a) IF ( $n \% i == 0$ ) THEN  
         DISPLAY "NOT PRIME"  
         (GOTO Step 7)  
     (b)  $i = i + 1$   
     END WHILE  
 STEP6 DISPLAY "PRIME"  
 STEP7 END

(23) WAP to print FIBONACCI SERIES upto a given no.

Sol-

STEP1 START  
 STEP2 Input a non negative number into variable  $n$   
 STEP3  $i = 0$  and temp\_var = 0  
 STEP4 temp1 = 0  
 STEP5 temp2 = 1  
 STEP6 If  $n \geq 0$  Then  
     DISPLAY temp1  
     ENDIF  
 STEP7 Repeat steps (a)- (d) while  $n > i$   
     (a) temp2 = temp2 + temp1  
     (b) DISPLAY temp2  
     (c) temp2 = temp2 + temp1  
     (d)  $i = i + 1$   
     END while

STEP 8 END

(24) WAP to calculate sum of digits of a number.

Sol-

STEP 1 START

STEP 2 INPUT a positive integer from user into variable  $N$

STEP 3  $SUM = 0$

STEP 4 Repeat steps while  $N > 0$

(a)  $SUM = SUM + (N \% 10)$

(b)  $N = N / 10$

END While

STEP 5 DISPLAY  $N$

STEP 6 END

(25) WAP to enter a string and check if it's a palindrome.

Sol→

STEP 1 START

STEP 2 Input the string in a variable  $S$

STEP 3 Initialize two pointer variables, one at start & one at end of  $S$

STEP 4 Repeat steps while start-pointer < end-pointer

(a) ~~for loop~~ IF value at start-pointer = value at end-pointer  
goto next step (i.e b)

ELSE

DISPLAY "not palindrome"

goto STEP 6

(b) increment and decrement start and end pointers respectively

END While

STEP 5 DISPLAY "is a palindrome"

STEP 6 END

(26)

WAP to sort a list of numbers entered by user.

Sol →

STEP 1. START

STEP 2. Input an array of numbers into variables arr of length n

STEP 3. i = 0 and j = 0

STEP 4. Repeat steps (a) & (b) while  $i < n - 1$ (a) Repeat steps (a) & (b) while  $j < n - i - 1$ (a.) IF  $arr[j] > arr[j + 1]$  THEN

temp = arr[j]

arr[j] = arr[j + 1],

(b)  $j = j + 1$ 

ENDIF

(b<sub>1</sub>)  $j = j - 1$ 

END While

(b)  $i = i + 1$ 

END While

STEP 5. Display arr

STEP 6. END

(27)

WAP to find max & min numbers enter in a list  
entered by user.

Sol →

STEP 1. START

STEP 2. Input an array of integers from user into variable list of n

lengths

STEP 3. MAX = list[0]

STEP 4. MIN = list[0]

STEP 5. i = 0

STEP 6. Repeat steps (a) & (b) while  $i < n$ 

(a) IF MAX &lt; list[i] THEN

MAX = list[i]

ENDIF

(b) IF MIN &gt; list[i] THEN

MIN = list[i]

ENDIF

(c)  $i = i + 1$ 

STEP 7. END WHILE

DISPLAY MAX and MIN

STEP 8. END

(28) WAP to calculate power of a num using loop.

Sol -

STEP1 START

STEP2 INPUT a number and another number into variables b and e  
(base) (exponent)

STEP3 IF  $e < 0$

STEP3 i = 0 and power = 1

STEP4 Repeat steps 3 while  $i \leq e$

(a) power = power \* base

(b)  $i = i + 1$

END while

STEP5 DISPLAY power

STEP6 END

(29) WAP that reads a sentence and counts the no. of words in it.

Sol -

STEP1 START

STEP2 Input a sentence into a string S of length n characters

STEP3 i = SPACE = 0

STEP4 Words = 0 and i = 0 and len = 0

STEP5 Repeat steps (a) while  $i < n$  and  $S[i] \neq '\backslash 0'$

STEP6 (a) IF  $S[i] \neq '\backslash 0'$  THEN

END WHILE  
 $i = 0$

STEP7 Repeat steps while  $i < len$

IF  $S[i] == '$  THEN

IF Space == 0 THEN

Space = 1

Words = Words + 1

ELSE IF (Space == 1)

Space = 1

END IF

ELSE

SPACE = 0

END IF

STEP 8 DISPLAY words  
 STEP 9 END

(30) WAP to check if a no. is Armstrong no.

Ans

STEP1 START  
 STEP2 Input a number <sup>(+ve)</sup> into variable num  
 STEP3 temp = num  
 STEP4 temp1 = num  
 STEP5 digits = 0  
 STEP6 nd = 1 and i=0 and j=0  
 STEP7 Repeat steps (a)-(b) while temp > 0  
     (a) temp = temp / 10  
     (b) digits = digits + 1  
 END while  
 STEP8 temp = 0  
 STEP9 Repeat steps (a)-(e) while i < digits  
     (a) Repeat steps (a),(b) while j < digits  
         nd = nd \* temp1 % 10  
         j = j + 1  
     End while  
     (b) temp = temp + nd  
     (c) temp1 = temp1 / 10  
     (d) nd = 1  
     (e) i = i + 1  
 End while  
 STEP10 DISPLAY IF temp == num THEN  
     DISPLAY "Armstrong number"  
   ELSE  
     DISPLAY "NOT Armstrong number"  
 STEP11 END

(31) WAP to merge 2 user lists of numbers entered by user.

SOL -

~~QUESTION~~

STEP1 START

STEP2 Input 2 arrays of numbers ~~and~~ into A1 and A2 with n and m lengths respectively.

STEP3 Initialize an array A3 with length m+n.

STEP4 i=0

STEP5 Repeat steps (a)-(b) while  $i < m+n$

(a) IF  $i < n$  THEN

$A3[i] = A1[i]$

ELSEIF  $i \geq n$  and  $i < m$  THEN

$A3[i] = A2[i]$

ENDIF

(b)  $a = a + 1$

END while

STEP6 DISPLAY A3

STEP7 END

(32) WAP to find second largest no. in a list entered by user

Step1 START

STEP2 Input the list in an array variable arr of length n

STEP3 i=0 and j=0

STEP4 Repeat steps (a)-(b) while  $i < n-1$

(a) Repeat steps (a<sub>1</sub>)-(b<sub>1</sub>) while  $j < n-i-1$

(a<sub>1</sub>) If  $arr[j] > arr[j+1]$  THEN

$temp = arr[j]$

$arr[j] = arr[j+1]$

$arr[j+1] = temp$

(b<sub>1</sub>)  $j = j + 1$

ENDIF

b<sub>1</sub>  $j = j + 1$

END while

(b)  $i = i + 1$

END while

STEP5 Display arr[n-1]

STEP6 END

(38) WAP to remove duplicate numbers from a list entered by the user.

Sol-

STEP1 Start

STEP2 Input an array into variable arr of length n

STEP3 i = 0 & j = 0

STEP4 ~~Declare~~ Declare an array result of size n, declare avar for

STEP5 Repeat steps (a)-(e) while i < n  
 result\_size  
 and set it 100

(a) current\_element = arr[i].

(b) is\_duplicate = 0

(c) Repeat steps(a) while  $j < \text{result\_size}$ .

(a.) if  $\text{current\_element} == \text{result}[j]$

is\_duplicate = 1;

(b.) ~~i=j+1~~  
~~break;~~

END IF

END while

(d) IF (not isDuplicate)

result [result\_size] = current\_element

result\_size = result\_size + 1

END IF

(e) i = i + 1

END WHILE

STEP6 DISPLAY result

STEP7 END

(34) WAP to find sum of all numbers in a matrix entered by user.

Ans -

STEP1 Start

STEP2 Input a 2D array (matrix) into variable arr of rows n and m columns

STEP3 i = 0 and j = 0

STEP4 SUM = 0

STEP5 Repeat steps while  $i \leq n$

(a) Repeat steps (a)-(b) while  $j \leq m$

(a)  $SUM = SUM + arr[i][j]$

(b1)  $j = j + 1$   
END while

(b)  $i = i + 1$

END while

STEP6 DISPLAY SUM

STEP7 END

(35) WAP to check if two strings are anagrams.

STEP1 START

STEP2 Input 2 strings in variables a and b of length m and n

STEP3 i = 0, & j = 0

STEP4 Repeat steps while IF  $m \neq n$  THEN  
DISPLAY 'NO'  
GOTO Step 7

ENDIF

STEP5 Repeat steps (a)-(d) while  $i \leq m$

(a) Repeat steps while  $j \leq m$

(a1) IF  $a[i] \neq a[j]$  THEN

is-duplicate = 0

ELSE  
is-duplicate = 1  
ENDIF

(a2)  $j = j + 1$

(b) IF is-duplicate = 1 THEN

~~DISPLAY 'NO'~~

GOTO Step 7

(c)  $i = 0$   
(d)  $i = i + 1$   
END while

STEP 6 DISPLAY 'YES'

STEP 7 END

(36) WAP to implement a basic calculator that can add subtract, multiply and divide two numbers entered by user.

Sol-

STEP 1 START

STEP 2 Input 2 numbers in variables x and y

STEP 3 DISPLAY X+Y

STEP 4 DISPLAY X-Y

STEP 5 DISPLAY X\*Y

STEP 6 IF  $y \neq 0$  THEN

DISPLAY X/Y

ELSE IF ELSE IF

DISPLAY "Division not possible"

END IF

STEP 7 END

(37) WAP to calculate total price of items bought in a store based on their quantities and prices entered by user.

Sol- STEP 1 - START

STEP 2 total-price = 0

STEP 3 quantity = 0

STEP 4 price = 0

STEP 5 Input total no. of items in variable ti

STEP 6 i = 0

STEP 7 Repeat steps (a)-(c) while  $i < ti$

(a) Input quantity and price of item in variable price & quantity

(b) total-price = quantity \* price

(c)  $i = i + 1$

END while

STEP8 DISPLAY total-price  
STEP9 END

(38) WAP to calculate avg temp over a week based on daily temps entered by user.

Sol-

STEP1 START

STEP2 i = 0

STEP3 temp = 0

STEP4 total\_temp = 0

STEPS Repeat steps (a)-(c) while  $i < 7$

(a) Input temp ( $\frac{^{\circ}\text{C}}$  a day) in variable temp

(b)  $\text{total\_temp} = \text{total\_temp} + \text{temp}$

(c)  $i = i + 1$

END While

STEP6 DISPLAY total\_temp / 7

STEP7 END

(39) WAP to convert time in seconds to hour, minute & seconds.

Sol-

STEP1 START

STEP2 Input time in seconds to variable s

STEP3  $h = s \div 3600$

STEP4  $s = s \% 3600$

STEP5  $m = s \% 60$

STEP6  $s = s \% 60$

STEP7 Display h, m, s

STEP8 END

(40) WAP to create a simple banking system where users can deposit, withdraw & check balance.

STEP1      START  
 STEP2      balance = 0  
 STEP3      Input initial balance into variable balance  
 STEP4      Repeat (a)-(c) steps while True  
 (a) DISPLAY " Enter 1 for deposit , 2 for withdrawal, 3 for Check balance"  
 (b) Take input in variable *inp*  
 (c) IF *inp* == 1 THEN  
     ~~DISPLAY~~ " Input deposit amount in variable temp  
     balance = temp + balance  
 ELSE  
     IF *inp* == 2  
         (+) Input withdraw amount in temp  
         IF temp <= balance then  
             balance = temp  
             balance = balance - temp  
         ~~END~~  
     ELSE  
         DISPLAY " insufficient balance "  
     END IF  
 ELSE IF *inp* == 3  
     DISPLAY BALANCE  
 ELSE IF *inp* == 4  
     DISPLAY " THANKS FOR USING BANK "  
     break;  
 ELSE  
     DISPLAY " invalid input "  
 STEPS      END

(Q1) WAP to calculate BMI of a person based on their weight & height.

Sol-

STEP1 START

STEP2 Input weight (in Kg) and height (in m) in variables wt and ht respectively.

STEP3  $BMI = wt / (ht * ht)$

STEP4 DISPLAY BMI

STEPS END

(Q2) WAP to convert a decimal no. to its binary equivalent.

STEP1 START

STEP2 Input decimal number into variable n

STEP3 Binary-string ~~= ""~~ declare b-str variable (a string variable)

STEP4 i = 0

STEP5 Repeat steps (a)-(c) while  $n > 0$

(a) IF  $n \% 2 == 0$  Then

b-str[i] = '0'

ELSE IF  $n \% 2 == 1$  Then

b-str[i] = '1'

END IF

(b)  $n = n / 2$

(c)  $i \leftarrow i + 1$

End while

STEP6 b-str[0] = '10'

STEP7 Display b-str

STEP8 END

STEP9 b-str = '10'

STEP10 Reverse b-str and store it into bin

STEP11 Display bin

STEP12 END

(43) WAP to make a voting system where users can vote for a candidate and the program counts the votes.

Sol

STEP1 START

STEP2 Input no. of candidates  $n$  in a variable  $m$  and  $mc$  <sup>& no. of voters</sup>  
                  organically

STEP3 Initialize array of length  $m+1$ , all elements to 0, name it  $votes$

STEP4  $i=0$

STEP5 Repeat Step1<sup>(a)</sup> while  $i < m+1$

- (a) input vote (1, 2, ..., m) for candidate and  $mc[i]$  for no. of voters above & store it in temp variable
- (b)  $votes[temp-1] = votes[temp-1] + 1$
- (c)  $i = i + 1$

End while

STEP6 DISPLAY votes

STEP7 END

Q4) WAP to find difference in two dates entered by user

STEP1 START

STEP2 Input 2 dates in variables <sup>but</sup> ~~in~~  $d_1, m_1, \text{ and } y_1$  and ~~as~~  $d_2, m_2$  and  $y_2$  respectively.

STEP3  $d=0$  and  $m=0$  and  $y=0$

STEP4 ~~if  $d_1 > d_2$~~  subtract larger ~~date~~ variable from smaller and then store it in  $d, m, y$  respectively for  $d_2-d_1$  or  $d_1-d_2$ ,  $m_1-m_2$  or  $m_2-m_1$ ,  $y_1-y_2$  or  $y_2-y_1$

STEP5 display  $d, m, y$

STEP6 END

(25) WAP to determine the day of the week based on no. entered by user.

Sol-

STEP1 START  
STEP2 Input ~~day~~<sup>number</sup> in a variable num.  
STEP3 if num  $\leq 1$  or num  $\geq 7$  Then  
    Display "Invalid input"  
END if  
if num == 1 THEN  
    DISPLAY "Monday"  
Else if num == 2 THEN  
    DISPLAY "Tuesday"

```

ELSE if num == 3 THEN
    DISPLAY "WEDNESDAY"
ELSE if num == 4 THEN
    DISPLAY "THURSDAY"
ELSE if num == 5 THEN
    DISPLAY "FRIDAY"
ELSE if num == 6 THEN
    DISPLAY "SATURDAY"
ELSE
    DISPLAY "SUNDAY"
END IF

```

STEPS

END

Q6) WAP to calculate final grade of a student based on their marks in multiple subjects.

Sol

STEP1

START

STEP2

Do Avg-marks = 0

STEP3

Marks=0

STEP4

Subjects = 5

STEP5

i=0

STEP6

Repeat steps while i &lt; subjects

(a) Input marks of subject in variable marks

(b) Avg-marks = Avg-marks + marks

(c) i = i + 1

STEP7

END while

Avg-marks = Avg-marks / subjects.

STEP8

IF Avg-marks &gt; 90 THEN

DISPLAY "A"

ELSE IF Avg-marks &gt; 80 THEN

DISPLAY "B"

ELSE IF Avg-marks &gt; 70 THEN

DISPLAY "C"

ELSE

DISPLAY "F"

END IF

STEP 9 END

Q7 WAP to calculate distance between points  $(x_1, y_1)$  and  $(x_2, y_2)$

Sol→

STEP 1 START

STEP 2 Input  $(x_1, y_1)$  and  $(x_2, y_2)$  in variables  $x_1, y_1, x_2$  and  $y_2$  respectively

STEP 3  $DISTANCE = \sqrt{(x_1 - x_2)^2}$

STEP 3  $D = \sqrt{(x_2 - x_1) * (x_2 - x_1) + (y_2 - y_1) * (y_2 - y_1)}$

STEP 4 DISPLAY D

STEP 5 END

Q8 WAP to calculate monthly installment of a loan based on principal, interest rate and no. of months.

Sol→

STEP 1 START

STEP 2 Input interest rate, principal, months in variables r, p and m respectively.

STEP 3  $y = m / 12$  (float division)

STEP 4  $SI = (p * r * y) / 100$

STEP 5 DISPLAY  ~~$p + SI$~~   $(p + SI) / m$  : princ of monthly installment

STEP 6 END

Q9 WAP to determine if a year is leap or not.

Sol→

STEP 1 START

STEP 2 Input year in variable y

STEP 3 IF  $(year \% 4 == 0)$  or and  $(year \% 100 != 0)$  or  $(year \% 400 == 0)$   
THEN

DISPLAY "leap year"

ELSE

DISPLAY "not leap year"

STEP 4 END

50 → WAP to stimulate a basic ATM where users can check their balance, deposit and withdraw money

Sols  
STEP 1 - START

STEP 2 - Input initial balance <sup>from</sup> from bank database in var bal

STEP 3 - Repeat steps while True <sup>& bal >= 0</sup>

(a) Input user's account number & compare it from bank database.

(b) IF Details not correct then

DISPLAY "Invalid account"  
GOTO step 3

END IF

(c) Input user's choice 1 for checking balance, 2 for deposit  
3 for withdraw money in variable inp  
and 4 to quit

(d) IF inp == 1 Then

Display balance

ELSE IF inp == 2 Then

Input withdraw deposit money amount in variable tb.  
balance = balance + tb

ELSE IF inp == 3 Then

Input money to be withdrawn in variable tb

IF tb <= balance and tb <= ~~bal~~ withdraw\_limit Then  
balance = balance - tb

ELSE

Display "Amount too large"

ENDIF

ELSE if inp == 4 Then

Display "Thanks for using"  
break

ELSE

Display "invalid choice"

END IF

STEP 4 → END