**GLA UNIVERSITY**

**MATHURA**

****

ASSIGNMENT - 1

C-PROGRAMMING

|  |  |  |
| --- | --- | --- |
| SUBMITTED BY – DEEPAK NISHAD  B.TECH 1st (CSE)  SEC- AL  ROLL NO. - 20 |  | SUBMITTED TO –  MISS GURPEET KAUR MA’AM  DATE – 18/09/2023 |

Q1. WRITE A C PROGRAM FOR CALCULATING THE PRICE OF A PRODUCT AFTER ADDING THE SALES TAX TO ITS ORIGINAL PRICE. WHERE RATE OF TAX AND PRICE IS INPUTTED BY USER.

**ANS 1.** #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

FLOAT ORIGINALPRICE, TAXRATE, TOTALPRICE;

PRINTF("ENTER THE ORIGINAL PRICE OF THE PRODUCT - RS.");

SCANF("%F", &ORIGINALPRICE);

PRINTF("ENTER THE SALES TAX RATE - ");

SCANF("%F", &TAXRATE);

TOTALPRICE = ORIGINALPRICE + (ORIGINALPRICE \* (TAXRATE / 100));

PRINTF("THE TOTAL PRICE OF THE PRODUCT AFTER TAX IS – RS. %.2F\N", TOTALPRICE);

RETURN 0;

}

Q2. WRITE A C PROGRAM TO CALCULATE THE WEEKLY WAGES OF AN EMPLOYEE. THE PAY DEPENDS ON WAGES PER HOUR AND NUMBER OF HOURS WORKED. MOREOVER, IF THE EMPLOYEE HAS WORKED FOR MORE THAN 30 HOURS, THEN HE OR SHE GETS TWICE THE WAGES PER HOUR, FOR EVERY EXTRA HOUR THAT HE OR SHE HAS WORKED.

**ANS 2**. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

FLOAT HOURLYWORK, WEEKLYWORK;

INT HOURSWORKED;

PRINTF("ENTER THE HOURLY WORK - RS.");

SCANF("%F", &HOURLYWORK);

PRINTF("ENTER THE NUMBER OF HOURS WORKED IN A WEEK: ");

SCANF("%D", &HOURSWORKED);

IF (HOURSWORKED <= 30)

{

WEEKLYWORK = HOURLYWORK \* HOURSWORKED;

} ELSE {

INT REGULARHOURS = 30;

INT EXTRAHOURS = HOURSWORKED - 30;

WEEKLYWORK = (HOURLYWORK \* REGULARHOURS) + (2 \* HOURLYWORK \* EXTRAHOURS);

}

PRINTF("WEEKLY WORK: $%.2F\N", WEEKLYWORK);

RETURN 0;

}

Q.3 MR. X GOES TO MARKET FOR BUYING SOME FRUITS AND VEGETABLES. HE IS HAVING A CURRENCY OF RS 500 WITH HIM FOR MARKETING. FROM A SHOP, HE PURCHASES 2.0 KG APPLE PRICED RS. 50.0 PER KG, 1.5 KG MANGO PRICED RS.35.0 PER KG, 2.5 KG POTATO PRICED RS.10.0 PER KG, AND 1.0 KG TOMATO PRICED RS.15 PER KG. HE GIVES THE CURRENCY OF RS. 500 TO THE SHOPKEEPER. FIND OUT THE AMOUNT SHOPKEEPER WILL RETURN TO X BY WRITING A C PROGRAM.

**ANS 3.**

#INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

FLOAT PRICEMANGOPERKG = 35.0;

FLOAT PRICEAPPLEPERKG = 50.0;

FLOAT PRICEPOTATOPERKG = 10.0;

FLOAT PRICETOMATOPERKG = 15.0;

FLOAT BUYMANGOKG = 1.5;

FLOAT BUYAPPLEKG = 2.0;

FLOAT BUYPOTATOKG = 2.50;

FLOAT BUYTOMATOKG = 1.0;

FLOAT TOTALCOST = (PRICEAPPLEPERKG \* BUYAPPLEKG ) +

(PRICEMANGOPERKG \* BUYMANGOKG ) +

(PRICEPOTATOPERKG \* BUYPOTATOKG ) +

(PRICETOMATOPERKG \* BUYTOMATOKG );

FLOAT AMOUNTPAID = 500.0;

FLOAT CHANGE = AMOUNTPAID - TOTALCOST;

IF (CHANGE >= 0) {

PRINTF("MR. X'S CHANGE: RS. %.2F\N", CHANGE);

} ELSE {

PRINTF("MR. X HAS NOT PAID ENOUGH. HE STILL OWES RS. %.2F\N", -CHANGE);

}

RETURN 0;

}

Q4.WRITE A C PROGRAM TO PRINT YOUR NAME, DATE OF BIRTH AND MOBILE NUMBER IN 3 DIFFERENT LINES.

**ANS 4.** #INCLUDE <STDIO.H>

INT MAIN()

{

CHAR NAME[] = "DEEPAK";

CHAR DOB[] = "DATE OF BIRTH: ";

CHAR MOB[] = "MOBILE NUMBER: 000000\*\*\*\*";

PRINTF("%S\N",NAME);

PRINTF("%S\N",DOB);

PRINTF("%S\N",MOB);

RETURN 0;

}

Q5.WRITE A PROGRAM TO READ AN INTEGER, A CHARACTER AND A FLOAT VALUE FROM KEYBOARD AND DISPLAY THE SAME IN DIFFERENT LINES ON THE SCREEN.

ANS 5. **.** #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

INT INTEGER\_INPUT;

CHAR CHAR\_INPUT;

FLOAT FLOAT\_INPUT;

PRINTF("ENTER AN INTEGER: ");

SCANF("%D",&INTEGER\_INPUT);

PRINTF("ENTER A CHARACTER: ");

SCANF(" %C", &CHAR\_INPUT);

PRINTF("ENTER A FLOAT VALUE: ");

SCANF("%F", &FLOAT\_INPUT);

PRINTF("INTEGER: %D\N", INTEGER\_INPUT);

PRINTF("CHARACTER: %C\N", CHAR\_INPUT);

PRINTF("FLOAT: %F\N", FLOAT\_INPUT);

RETURN 0;

}

Q6.WRITE A PROGRAM TO PRINT THE FOLLOWING LINE ( ASSUME THE TOTAL VALUE IS CONTAINED IN A VARIABLE NAMED COST)

THE SALES TOTAL IS : $ 172.53

ANS 6 #INCLUDE <STDIO.H>

INT MAIN()

{

PRINTF("THE SALES TOTAL IS ASSUME THE TOTAL VALUE IS CONTAINED IN A VARIABLE NAMED COST\N");

PRINTF("THE SALES TOTAL IS : $ 172.53");

RETURN 0;

}

Q7.RAJU GOT 6 AND HALF APPLES FROM EACH OF RAGHU, SHEENU AND AKASH. HE WANTS TO KNOW HOW MANY APPLES HE HAS IN TOTAL WITHOUT ADDING THEM. WRITE A PROGRAM WHICH COULD HELP RAJU IN DOING THIS.

ANS 7. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

INT APPLES\_FROM\_RAGHU = 6;

FLOAT APPLES\_FROM\_SHEENU = 0.5;

FLOAT APPLES\_FROM\_AKASH = 0.5;

FLOAT TOTAL\_APPLES = APPLES\_FROM\_RAGHU + APPLES\_FROM\_SHEENU + APPLES\_FROM\_AKASH;

PRINTF("RAJU HAS A TOTAL OF %.1F APPLES.\N", TOTAL\_APPLES);

RETURN 0;

}

Q8.WRITE A PROGRAM THAT PRINTS THE FLOATING POINT VALUE IN EXPONENTIAL FORMAT CORRECT TO TWO DECIMAL PLACES.

ANS 8. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

DOUBLE VALUE = 123456.789;

PRINTF("VALUE IN EXPONENTIAL FORMAT: %.2E\N", VALUE);

RETURN 0;

}

Q9.WRITE A PROGRAM TO INPUT AND PRINT YOUR MOBILE NUMBER (I.E. OF 10 DIGITS).

ANS 9. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

LONG LONG INT MOBILENUMBER;

PRINTF("ENTER YOUR 10-DIGIT MOBILE NUMBER: ");

SCANF("%LLD", &MOBILENUMBER);

IF (MOBILENUMBER >= 1000000000LL && MOBILENUMBER <= 9999999999LL) {

PRINTF("YOUR MOBILE NUMBER IS: %LLD\N", MOBILENUMBER);

} ELSE {

PRINTF("INVALID INPUT. PLEASE ENTER A 10-DIGIT MOBILE NUMBER.\N");

}

RETURN 0;

}

Q10.THE POPULATION OF A CITY IS 30000. IT INCREASES BY 20 % DURING FIRST YEAR AND 30% DURING THE SECOND YEAR. WRITE A PROGRAM TO FIND THE POPULATION AFTER TWO YEARS? (ANS: 46800)

ANS 10. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

INT INITIAL\_POPULATION = 30000;

// INCREASE IN THE FIRST YEAR (20%)

DOUBLE INCREASE\_IN\_FIRST\_YEAR = INITIAL\_POPULATION \* 0.20;

// POPULATION AFTER THE FIRST YEAR

DOUBLE POPULATION\_AFTER\_FIRST\_YEAR = INITIAL\_POPULATION + INCREASE\_IN\_FIRST\_YEAR;

// INCREASE IN THE SECOND YEAR (30%)

DOUBLE INCREASE\_IN\_SECOND\_YEAR = POPULATION\_AFTER\_FIRST\_YEAR \* 0.30;

// POPULATION AFTER THE SECOND YEAR

DOUBLE POPULATION\_AFTER\_TWO\_YEARS = POPULATION\_AFTER\_FIRST\_YEAR + INCREASE\_IN\_SECOND\_YEAR;

// PRINT THE POPULATION AFTER TWO YEARS

PRINTF("POPULATION AFTER TWO YEARS: %.2LF\N", POPULATION\_AFTER\_TWO\_YEARS);

RETURN 0;

}

Q11. WRITE A PROGRAM TO FIND THE ASCII VALUE OF A CHARACTER.

ANS. 11 #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

CHAR CHARACTER;

// INPUT A CHARACTER FROM THE USER

PRINTF("ENTER A CHARACTER: ");

SCANF("%C", &CHARACTER);

// FIND AND PRINT THE ASCII VALUE

INT ASCII\_VALUE = (INT)CHARACTER;

PRINTF("ASCII VALUE OF %C IS %D\N", CHARACTER, ASCII\_VALUE);

RETURN 0;

}

Q12. WRITE A PROGRAM TO CALCULATE SALARY OF AN EMPLOYEE, GIVEN HIS BASIC PAY (ENTERED BY USER), HRA=15% OF THE BASIC PAY AND TA=20% OF THE BASIC PAY.

ANS 12 #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

DOUBLE BASIC\_PAY, HRA, TA, SALARY;

// INPUT BASIC PAY FROM THE USER

PRINTF("ENTER THE BASIC PAY: ");

SCANF("%LF", &BASIC\_PAY);

// CALCULATE HRA AND TA

HRA = 0.15 \* BASIC\_PAY; // HRA IS 15% OF BASIC PAY

TA = 0.20 \* BASIC\_PAY; // TA IS 20% OF BASIC PAY

// CALCULATE THE TOTAL SALARY

SALARY = BASIC\_PAY + HRA + TA;

// DISPLAY THE SALARY BREAKDOWN

PRINTF("BASIC PAY: %.2LF\N", BASIC\_PAY);

PRINTF("HRA: %.2LF\N", HRA);

PRINTF("TA: %.2LF\N", TA);

PRINTF("TOTAL SALARY: %.2LF\N", SALARY);

RETURN 0;

}

Q13. WRITE A PROGRAM TO FIND THE SLOPE OF A LINE AND ANGLE OF INCLINATION THAT PASSES THROUGH TWO POINTS P AND Q WITH COORDINATES (XP, YP) AND (XQ, YQ) RESPECTIVELY.

ANS 13 #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

DOUBLE XP, YP, XQ, YQ;

// INPUT COORDINATES OF POINTS P AND Q

PRINTF("ENTER THE COORDINATES OF POINT P (XP YP): ");

SCANF("%LF %LF", &XP, &YP);

PRINTF("ENTER THE COORDINATES OF POINT Q (XQ YQ): ");

SCANF("%LF %LF", &XQ, &YQ);

DOUBLE SLOPE = (YQ - YP) / (XQ - XP);

DOUBLE ANGLE\_RAD = ATAN(SLOPE);

DOUBLE ANGLE\_DEG = ANGLE\_RAD \* (180.0 / M\_PI);

PRINTF("THE SLOPE OF THE LINE IS: %.2LF\N", SLOPE);

PRINTF("THE ANGLE OF INCLINATION (IN DEGREES) IS: %.2LF\N", ANGLE\_DEG);

RETURN 0;

}

Q14. THE SPI (SEMESTER PERFORMANCE INDEX) IS A WEIGHTED AVERAGE OF THE GRADE POINTS EARNED BY A STUDENT IN ALL THE COURSES HE REGISTERED FOR IN A SEMESTER. IF THE GRADE POINTS ASSOCIATED WITH THE LETTER GRADES AWARDED TO A STUDENT ARE G1, G2, G3,…….GK ETC. AND THE CORRESPONDING CREDITS ARE C1, C2, C3,.…..CK, THE SPI IS GIVEN BY:

A

ANS 14 #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

DOUBLE GRADEPOINTS[5] = {3.5, 4.0, 3.0, 3.7, 3.2};

DOUBLE CREDITS[5] = {3.0, 4.0, 3.0, 2.0, 3.5};

INT NUMCOURSES = 5; // NUMBER OF COURSES

DOUBLE TOTALGRADEPOINTS = 0.0;

DOUBLE TOTALCREDITS = 0.0;

FOR (INT I = 0; I < NUMCOURSES; I++) {

TOTALGRADEPOINTS += GRADEPOINTS[I] \* CREDITS[I];

TOTALCREDITS += CREDITS[I];

}

IF (TOTALCREDITS > 0) {

DOUBLE SPI = TOTALGRADEPOINTS / TOTALCREDITS;

PRINTF("SPI (SEMESTER PERFORMANCE INDEX) IS: %.2LF\N", SPI);

} ELSE {

PRINTF("TOTAL CREDITS CANNOT BE ZERO.\N");

}

RETURN 0;

}

Q 15. WRITE A PROGRAM TO CALCULATE THE FREQUENCY (F) OF A GIVEN WAVE WITH WAVELENGTH (Λ) AND SPEED (C), WHERE C=Λ\*F.

ANS15. #INCLUDE <STDIO.H>

#INCLUDE<MATH.H>

INT MAIN()

{

DOUBLE WAVELENGTH, SPEED, FREQUENCY;

// INPUT WAVELENGTH (IN METERS) AND SPEED (IN METERS PER SECOND)

PRINTF("ENTER THE WAVELENGTH (IN METERS): ");

SCANF("%LF", &WAVELENGTH);

PRINTF("ENTER THE SPEED (IN METERS PER SECOND): ");

SCANF("%LF", &SPEED);

IF (WAVELENGTH > 0 && SPEED > 0) {

FREQUENCY = SPEED / WAVELENGTH;

PRINTF("THE FREQUENCY OF THE WAVE IS: %.2LF HZ\N", FREQUENCY);

} ELSE {

PRINTF("INVALID INPUT. WAVELENGTH AND SPEED MUST BE POSITIVE VALUES.\N");

}

RETURN 0;

}

Q 16. A CAR TRAVELLING AT 30 M/S ACCELERATES STEADILY AT 5 M/S2 FOR A DISTANCE OF 70 M. WHAT IS THE FINAL VELOCITY OF THE CAR? [HINT: V2 = U2 + 2AS]

ANS16. #INCLUDE <STDIO.H>

#INCLUDE <MATH.H>

INT MAIN() {

DOUBLE INITIALVELOCITY = 30.0; // INITIAL VELOCITY IN M/S

DOUBLE ACCELERATION = 5.0; // ACCELERATION IN M/S^2

DOUBLE DISTANCE = 70.0; // DISTANCE IN METERS

DOUBLE FINALVELOCITY = SQRT(INITIALVELOCITY \* INITIALVELOCITY + 2 \* ACCELERATION \* DISTANCE);

PRINTF("THE FINAL VELOCITY OF THE CAR IS %.2LF M/S\N", FINALVELOCITY);

RETURN 0;

}

Q 17.A HORSE ACCELERATES STEADILY FROM REST AT 4 M/S2 FOR 3S. (A) WHAT IS ITS FINAL VELOCITY? (B) HOW FAR HAS IT TRAVELLED? [HINT: (A) V = U + AT (B) S = UT + ½AT2 ]

ANS 17. (A) #INCLUDE <STDIO.H>

INT MAIN() {

DOUBLE INITIALVELOCITY = 0.0; // INITIAL VELOCITY IN M/S

DOUBLE ACCELERATION = 4.0; // ACCELERATION IN M/S^2

DOUBLE TIME = 3.0; // TIME IN SECONDS

// CALCULATE THE FINAL VELOCITY USING THE EQUATION: V = U + AT

DOUBLE FINALVELOCITY = INITIALVELOCITY + ACCELERATION \* TIME;

PRINTF("THE FINAL VELOCITY OF THE HORSE IS %.2LF M/S\N", FINALVELOCITY);

RETURN 0;

}

(B) HOW FAR HAS IT TRAVELLED?

#INCLUDE <STDIO.H>

INT MAIN() {

DOUBLE INITIALVELOCITY = 0.0; // INITIAL VELOCITY IN M/S

DOUBLE ACCELERATION = 4.0; // ACCELERATION IN M/S^2

DOUBLE TIME = 3.0; // TIME IN SECONDS

// CALCULATE THE DISTANCE USING THE EQUATION: S = UT + 0.5 \* AT^2

DOUBLE DISTANCE = INITIALVELOCITY \* TIME + 0.5 \* ACCELERATION \* TIME \* TIME;

PRINTF("THE HORSE HAS TRAVELED A DISTANCE OF %.2LF METERS\N", DISTANCE);

RETURN 0

}

Q 18. WRITE A PROGRAM TO FIND THE SUM OF YOUR FOUR LAST DIGIT OF YOUR UNIVERSITY ROLL NUMBER .

ANS 18. INT MAIN()

{

INT ROLLNUMBER = 2504

INT LASTFOURDIGITS = ROLLNUMBER % 10000;

INT SUM = 0;

// CALCULATE THE SUM OF THE FOUR LAST DIGITS

WHILE (LASTFOURDIGITS > 0) {

SUM += LASTFOURDIGITS % 10;

LASTFOURDIGITS /= 10;

}

PRINTF("THE SUM OF THE FOUR LAST DIGITS OF THE UNIVERSITY ROLL NUMBER IS %D\N", SUM);

RETURN 0;

}

//(OUTPUT- 11)

Q19. WRITE A PROGRAM TO INITIALIZE YOUR HEIGHT AND WEIGHT IN CM. AND KGS RESPECTIVELY DEMONSTRATING COMPILE TIME INITIALIZATION AND CONVERT THEM IN FEETS AND POUNDS RESPECTIVELY. **NOTE :- 1 CM = 0.393701INCH , 1 KG = 2.20462**

**ANS 19. #INCLUDE <STDIO.H>**

**INT MAIN() {**

**DOUBLE HEIGHTINCM = 165.0;**

**DOUBLE WEIGHTINKG = 60;**

**DOUBLE CMTOINCH = 0.393701;**

**DOUBLE KGTOPOUND = 2.20462;**

**// CONVERT HEIGHT TO FEET AND WEIGHT TO POUNDS**

**DOUBLE HEIGHTINFEET = HEIGHTINCM \* CMTOINCH / 12.0;**

**DOUBLE WEIGHTINPOUNDS = WEIGHTINKG \* KGTOPOUND;**

**PRINTF("HEIGHT: %.2LF CM = %.2LF FEET\N", HEIGHTINCM, HEIGHTINFEET);**

**PRINTF("WEIGHT: %.2LF KG = %.2LF POUNDS\N", WEIGHTINKG, WEIGHTINPOUNDS);**

**RETURN 0;**

**}**

**OUTPUT- HEIGHT : 165CM = 5.41FEET**

**WEIGHT – 60.00 KG = 132.28 POUNDS.**

Q 20 . CODE THE VARIABLE DECLARATIONS FOR EACH OF FOLLOWING:

1. A CHARACTER VARIABLE NAMED OPTION.
2. AN INTEGER VARIABLE SUM INITIALIZED TO 0
3. A FLOATING POINT VARIABLE, PRODUCT, INITIALIZED TO 1

**ANS (A) #INCLUDE <STDIO.H>**

**INT MAIN() {**

**CHAR OPTION;**

**RETURN 0;**

**}**

(B) **. #INCLUDE <STDIO.H>**

**INT MAIN() {**

INT SUM = 0;

RETURN 0;

}

© **. #INCLUDE <STDIO.H>**

**INT MAIN() {**

FLOAT PRODUCT = 1.0;

RETURN 0;

}

Q21. WRITE A PROGRAM THAT READS NINE INTEGERS. DISPLAY THESE NUMBERS BY PRINTING THREE NUMBERS IN A LINE SEPARATED BY COMMAS.

ANS 21. **#INCLUDE <STDIO.H>**

**INT MAIN() {**

INT NUMBERS[9];

// READ NINE INTEGERS

PRINTF("ENTER NINE INTEGERS:\N");

FOR (INT I = 0; I < 9; I++) {

SCANF("%D", &NUMBERS[I]);

}

FOR (INT I = 0; I < 9; I++) {

PRINTF("%D", NUMBERS[I]);

IF ((I + 1) % 3 == 0) {

// PRINT A COMMA AND NEWLINE AFTER EVERY THIRD NUMBER

PRINTF(",\N");

} ELSE {

// PRINT A COMMA AFTER THE NUMBER IF IT'S NOT THE THIRD IN THE GROUP

PRINTF(", ");

}

RETURN 0;

}

Q22. WHAT ARE HEADER FILES AND WHAT ARE ITS USES IN C PROGRAMMING?

ANS22. HEADER FILES IN C PROGRAMMING ARE FILES THAT CONTAIN DECLARATIONS, DEFINITIONS, AND OTHER INFORMATION ABOUT FUNCTIONS, VARIABLES, MACROS, AND DATA TYPES USED IN A C PROGRAM. THEY SERVE SEVERAL IMPORTANT PURPOSES IN C PROGRAMMING:

1. **DECLARATION OF FUNCTIONS AND VARIABLES**: HEADER FILES PROVIDE DECLARATIONS (PROTOTYPES) FOR FUNCTIONS AND VARIABLES THAT ARE DEFINED IN OTHER SOURCE FILES (USUALLY IMPLEMENTATION FILES). THESE DECLARATIONS ALLOW OTHER PARTS OF THE PROGRAM TO USE THESE FUNCTIONS AND VARIABLES WITHOUT HAVING TO KNOW THEIR INTERNAL DETAILS. THIS SEPARATION OF INTERFACE (DECLARATION) FROM IMPLEMENTATION (DEFINITION) IS A KEY PRINCIPLE IN SOFTWARE ENGINEERING AND PROMOTES CODE MODULARITY.
2. **INCLUDE GUARD**: HEADER FILES OFTEN INCLUDE AN "INCLUDE GUARD" TO PREVENT MULTIPLE INCLUSIONS OF THE SAME HEADER FILE WITHIN A SINGLE COMPILATION UNIT. THIS GUARD TYPICALLY CONSISTS OF CONDITIONAL PREPROCESSOR DIRECTIVES (E.G., **#IFNDEF**, **#DEFINE**, AND **#ENDIF**) TO ENSURE THAT THE CONTENT OF THE HEADER IS INCLUDED ONLY ONCE. THIS PREVENTS ISSUES WITH REDEFINITION AND REDUCES COMPILATION TIME.
3. **DOCUMENTATION**: HEADER FILES OFTEN CONTAIN COMMENTS AND DOCUMENTATION THAT DESCRIBE THE FUNCTIONS, VARIABLES, MACROS, AND THEIR USAGE. THIS DOCUMENTATION HELPS PROGRAMMERS UNDERSTAND HOW TO USE THESE CONSTRUCTS CORRECTLY AND PROVIDES VALUABLE INFORMATION ABOUT THE PURPOSE AND BEHAVIOR OF THE CODE.
4. **MODULARITY**: HEADER FILES ALLOW YOU TO BREAK YOUR PROGRAM INTO SMALLER, MANAGEABLE PIECES. YOU CAN DEFINE FUNCTIONS, DATA STRUCTURES, AND CONSTANTS IN SEPARATE SOURCE FILES (IMPLEMENTATION FILES) AND PROVIDE THEIR DECLARATIONS IN HEADER FILES. THIS PROMOTES CODE MODULARITY, REUSABILITY, AND MAINTAINABILITY, AS DIFFERENT PARTS OF THE PROGRAM CAN BE WORKED ON INDEPENDENTLY.
5. **LIBRARY INTEGRATION**: HEADER FILES ARE ESSENTIAL FOR INTEGRATING EXTERNAL LIBRARIES OR APIS INTO YOUR C PROGRAMS. THESE FILES PROVIDE THE INTERFACE TO THE LIBRARY, ALLOWING YOUR CODE TO USE ITS FUNCTIONALITY. BY INCLUDING THE APPROPRIATE HEADER FILES, YOU CAN ACCESS AND USE FUNCTIONS AND DATA TYPES PROVIDED BY LIBRARIES.

COMMONLY USED C STANDARD LIBRARY HEADER FILES INCLUDE **<STDIO.H>** FOR INPUT AND OUTPUT, **<STDLIB.H>** FOR GENERAL UTILITIES, **<STRING.H>** FOR STRING MANIPULATION, **<MATH.H>** FOR MATHEMATICAL FUNCTIONS, AND MORE.

PROGRAMMERS CAN ALSO CREATE THEIR OWN HEADER FILES TO ENCAPSULATE CUSTOM FUNCTIONS, DATA STRUCTURES, AND MACROS THEY DEFINE IN THEIR PROGRAMS, MAKING IT EASIER TO REUSE CODE AND COLLABORATE ON LARGER PROJECTS. THESE HEADER FILES ARE TYPICALLY PAIRED WITH IMPLEMENTATION FILES (E.G., **.C** FILES) THAT CONTAIN THE ACTUAL CODE FOR THE FUNCTIONS AND VARIABLES DECLARED IN THE HEADER.

Q23. WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM?

#INCLUDE<STDIO.H>

INT MAIN()

{ INT NUM=070;

PRINTF(“%D\T%O\T%X”,NUM,NUM,NUM);

}

Top of Form

ANS 23.

#INCLUDE<STDIO.H>

INT MAIN()

// DISPLAY THE NUMBERS IN GROUPS OF THREE

FOR (INT I = 0; I < 9; I++) {

PRINTF("%D", NUMBERS[I]);

IF ((I + 1) % 3 == 0) {

// PRINT A COMMA AND NEWLINE AFTER EVERY THIRD NUMBER

PRINTF(",\N");

} ELSE {

// PRINT A COMMA AFTER THE NUMBER IF IT'S NOT THE THIRD IN THE GROUP

PRINTF(", ");

}

}

RETURN 0;

}

Q 24. WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM?

#INCLUDE <STDIO.H>

VOID MAIN()

{

INT X = PRINTF("GLA UNIVERSITY");

PRINTF("%D", X);

}

ANS **24.** #INCLUDE<STDIO.H>

INT MAIN()

**INT X = PRINTF("GLA UNIVERSITY");**

**PRINTF("%D", X);**

**RETURN 0;**

**}**

Q25. WHAT ARE LIBRARY FUNCTIONS? LIST ANY FOUR LIBRARY FUNCTIONS.

ANS 25. LIBRARY FUNCTIONS ARE PRE-WRITTEN FUNCTIONS PROVIDED BY STANDARD LIBRARIES OR EXTERNAL LIBRARIES THAT CAN BE USED IN A C PROGRAM TO PERFORM COMMON TASKS. THESE FUNCTIONS ARE DESIGNED TO BE REUSABLE AND SAVE PROGRAMMERS TIME BY PROVIDING WELL-TESTED AND EFFICIENT IMPLEMENTATIONS OF FREQUENTLY USED OPERATIONS. HERE ARE FOUR EXAMPLES OF LIBRARY FUNCTIONS:

1. **PRINTF**: A STANDARD C LIBRARY FUNCTION USED FOR FORMATTED OUTPUT TO THE CONSOLE. IT ALLOWS YOU TO PRINT DATA IN VARIOUS FORMATS, SUCH AS INTEGERS, FLOATING-POINT NUMBERS, AND STRINGS.
2. **SCANF**: ANOTHER STANDARD C LIBRARY FUNCTION USED FOR FORMATTED INPUT FROM THE CONSOLE. IT ALLOWS YOU TO READ DATA OF VARIOUS TYPES FROM THE USER.
3. **STRLEN**: THIS FUNCTION, FOUND IN THE **<STRING.H>** HEADER, CALCULATES THE LENGTH OF A NULL-TERMINATED STRING (I.E., A C STRING).
4. **SQRT**: A MATHEMATICAL FUNCTION FROM THE **<MATH.H>** HEADER THAT COMPUTES THE SQUARE ROOT OF A NUMBER.

Q26. WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM?

#INCLUDE <STDIO.H>

VOID MAIN()

{

INT X = PRINTF("C IS PLACEMENT ORIENTED LANGUAGE") – PRINTF(“HI”);

PRINTF("%D %O %X", X,X,X);

}

**ANS 26. #INCLUDE <STDIO.H>**

**INT MAIN()**

**{**

**INT X = PRINTF("C IS PLACEMENT ORIENTED LANGUAGE");**

**PRINTF("%D", X,X,X);**

**}**

**RETURN 0;**

Q27. WHAT IS THE MEANING OF FOLLOWING STATEMENT? PRINTF(“%D”,SCANF(“%D%D”,&A,&B));

ANS 27. THE STATEMENT **PRINTF("%D", SCANF("%D%D", &A, &B));** INVOLVES BOTH THE **SCANF** AND **PRINTF** FUNCTIONS. LET'S BREAK IT DOWN:

1. **SCANF("%D%D", &A, &B);**: THIS PART OF THE STATEMENT IS A CALL TO THE **SCANF** FUNCTION. IT IS ATTEMPTING TO READ TWO INTEGER VALUES FROM THE USER AND STORE THEM IN THE VARIABLES **A** AND **B**. THE FORMAT STRING **"%D%D"** SPECIFIES THAT TWO INTEGER VALUES ARE EXPECTED TO BE ENTERED.
2. **PRINTF("%D", ...);**: THIS PART OF THE STATEMENT IS A CALL TO THE **PRINTF** FUNCTION. IT IS ATTEMPTING TO PRINT A SINGLE INTEGER USING THE FORMAT SPECIFIER **%D**. HOWEVER, WHAT GETS PRINTED DEPENDS ON THE RETURN VALUE OF THE **SCANF** FUNCTION.

NOW, THE KEY THING TO UNDERSTAND IS THAT **SCANF** RETURNS THE NUMBER OF SUCCESSFULLY MATCHED AND ASSIGNED INPUT ITEMS. IN THIS CASE, IT WILL RETURN EITHER **0**, **1**, OR **2** BASED ON WHETHER ZERO, ONE, OR BOTH INTEGER VALUES WERE SUCCESSFULLY READ AND ASSIGNED.

SO, THE **PRINTF** STATEMENT WILL PRINT THE RETURN VALUE OF **SCANF**, WHICH CAN BE **0**, **1**, OR **2**, DEPENDING ON THE USER'S INPUT. FOR EXAMPLE, IF THE USER ENTERS TWO INTEGERS, IT WILL PRINT **2**. IF THE USER ENTERS ONLY ONE INTEGER, IT WILL PRINT **1**. IF THE USER ENTERS NON-INTEGER INPUT OR NOTHING, IT WILL PRINT **0**.

Q28. WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM?

#INCLUDE <STDIO.H>

VOID MAIN()

{

PRINTF(" \"C %% FOR %% PLACEMENT\"");

}

**ANS 28. #INCLUDE <STDIO.H>**

**INT MAIN() {**

**PRINTF(" \"C %% FOR %% PLACEMENT\"");**

**}**

**RETURN 0;**

**}**

Q29. SUPPOSE DISTANCE BETWEEN GLA UNIVERSITY AND DELHI IS M KM (TO BE ENTERED BY USER), BY BUS YOU CAN REACH DELHI IN 4 HOURS. DEVELOP A ‘C’ PROGRAM TO CALCULATE SPEED OF BUS.

**ANS 29. INT MAIN() {**

**FLOAT DISTANCE; // DISTANCE IN KILOMETERS**

**FLOAT TIME; // TIME IN HOURS**

**FLOAT SPEED; // SPEED IN KILOMETERS PER HOUR**

**PRINTF("ENTER THE DISTANCE BETWEEN GLA UNIVERSITY AND DELHI (IN KM): ");**

**SCANF("%F", &DISTANCE);**

**PRINTF("ENTER THE TIME TAKEN TO REACH DELHI BY BUS (IN HOURS): ");**

**SCANF("%F", &TIME);**

**SPEED = DISTANCE / TIME;**

**PRINTF("THE SPEED OF THE BUS IS %.2F KILOMETERS PER HOUR.\N", SPEED);**

**RETURN 0;**

**}**

Q30. IN AN EXAM SATYAM GOT 50 MARKS, SUMAN GOT 70 MARKS AND SHYAM GOT 80 MARKS, WRITE A ‘C’ PROGRAM TO FIND AVERAGE MARKS OF THESE THREE PARTICIPANTS.

**ANS 30. #INCLUDE <STDIO.H>**

**INT MAIN() {**

**INT SATYAM\_MARKS = 50;**

**INT SUMAN\_MARKS = 70;**

**INT SHYAM\_MARKS = 80;**

**FLOAT AVERAGE\_MARKS;**

**AVERAGE\_MARKS = (SATYAM\_MARKS + SUMAN\_MARKS + SHYAM\_MARKS) / 3.0;**

**PRINTF("THE AVERAGE MARKS OF SATYAM, SUMAN, AND SHYAM IS: %.2F\N", AVERAGE\_MARKS);**

**RETURN 0;**

**}**

Q31. ONE DAY, MOHAN CALLED SAURAV AND SAJAL AND GAVE SOME MONEY TO THEM, LATER HE REALIZED THAT MONEY THAT WAS GIVEN TO SAURAV SHOULD BE GIVEN TO SAJAL AND VICE-VERSA. DEVELOP A ‘C’ PROGRAM TO HELP MOHAN SO THAT HE CAN RECTIFY HIS MISTAKE.

**ANS 31.** **#INCLUDE <STDIO.H>**

**INT MAIN() {**

**FLOAT SAURAV\_MONEY; // AMOUNT GIVEN TO SAURAV**

**FLOAT SAJAL\_MONEY; // AMOUNT GIVEN TO SAJAL**

**PRINTF("ENTER THE AMOUNT GIVEN TO SAURAV: ");**

**SCANF("%F", &SAURAV\_MONEY);**

**PRINTF("ENTER THE AMOUNT GIVEN TO SAJAL: ");**

**SCANF("%F", &SAJAL\_MONEY);**

**FLOAT TEMP = SAURAV\_MONEY;**

**SAURAV\_MONEY = SAJAL\_MONEY;**

**SAJAL\_MONEY = TEMP;**

**PRINTF("AFTER RECTIFICATION:\N");**

**PRINTF("AMOUNT GIVEN TO SAURAV: %.2F\N", SAURAV\_MONEY);**

**PRINTF("AMOUNT GIVEN TO SAJAL: %.2F\N", SAJAL\_MONEY);**

**RETURN 0;**

**}**

Q32. ONE DAY WHEN I WAS GOING FOR A LUNCH, SUDDENLY RAIN STARTED, I WAS VERY HUNGRY SO STARTED RUNNING WITH SPEED OF 4KM/H AND IT TOOK 3 MIN TO REACH MESS. HELP ME TO DEVELOP A ‘C’ PROGRAM TO CALCULATE DISTANCE TRAVELLED BY ME.

ANS32. **#INCLUDE <STDIO.H>**

**INT MAIN() {**

DOUBLE SPEED\_KMPH = 4.0; // SPEED IN KILOMETERS PER HOUR

DOUBLE TIME\_MIN = 3.0; // TIME IN MINUTES

DOUBLE TIME\_HR = TIME\_MIN / 60.0;

DOUBLE DISTANCE\_KM = SPEED\_KMPH \* TIME\_HR;

PRINTF("DISTANCE TRAVELED: %.2F KILOMETERS\N", DISTANCE\_KM);

RETURN 0;

}

Q33. CAN TWO OR MORE ESCAPE SEQUENCES SUCH AS \N AND \T BE COMBINED IN A SINGLE LINE OF PROGRAM CODE?

ANS 33. NO, YOU CANNOT COMBINE MULTIPLE ESCAPE SEQUENCES LIKE **\N** AND **\T** IN A SINGLE LINE OF PROGRAM CODE. EACH ESCAPE SEQUENCE SHOULD BE USED SEPARATELY.

Q34. WHAT ARE COMMENTS AND HOW DO YOU INSERT IT IN A C PROGRAM?

ANS 34. COMMENTS IN C ARE USED TO PROVIDE EXPLANATIONS AND NOTES WITHIN THE SOURCE CODE FOR HUMANS TO UNDERSTAND. THEY ARE IGNORED BY THE COMPILER. IN C, YOU CAN INSERT COMMENTS USING TWO METHODS:

* SINGLE-LINE COMMENTS ARE CREATED USING

T

Q35. WHAT IS WRONG IN THIS STATEMENT? SCANF(“%D”,NUMBER);

ANS35 THE STATEMENT SCANF("%D", NUMBER); IS INCORRECT. IT SHOULD BE:

SCANF("%D", &NUMBER);

RETURN 0;

}

IN SCANF, YOU NEED TO PROVIDE THE MEMORY ADDRESS WHERE THE SCANNED VALUE SHOULD BE STORED. THEREFORE, YOU SHOULD USE THE & OPERATOR BEFORE THE VARIABLE NAME TO PASS ITS ADDRESS TO SCANF.

Q36. WHAT WILL BE THE OUTPUT?

#INCLUDE <STDIO.H>

INT MAIN()

{

IF (SIZEOF(INT) > -1)

PRINTF("YES");

ELSE

PRINTF("NO");

RETURN 0;

}

ANS 36. THE OUTPUT OF THE GIVEN PROGRAM WILL BE "YES." IN C, **SIZEOF(INT)** RETURNS THE SIZE (IN BYTES) OF THE **INT** DATA TYPE, WHICH IS GUARANTEED TO BE GREATER THAN OR EQUAL TO 1. THEREFORE, THE CONDITION **SIZEOF(INT) > -1** IS ALWAYS TRUE, AND "YES" WILL BE PRINTED.

Q37. POINT OUT WHICH OF THE FOLLOWING VARIABLE NAMES ARE INVALID:

GROSS-SALARY INTEREST , SALARY OF EMP , AVG. , THEREISBOOKINMYSOUP

ANS37. Q37. VARIABLE NAMES IN C SHOULD FOLLOW CERTAIN RULES:

* VARIABLE NAMES CANNOT CONTAIN SPACES, SO "SALARY OF EMP" AND "AVG." ARE INVALID.
* VARIABLE NAMES CAN CONTAIN LETTERS, DIGITS, AND UNDERSCORES, BUT THEY CANNOT START WITH A DIGIT. SO, "GROSS-SALARY" AND "INTEREST" ARE VALID VARIABLE NAMES.
* VARIABLE NAMES ARE CASE-SENSITIVE, SO "THEREISBOOKINMYSOUP" IS A VALID VARIABLE NAME.

SO, THE INVALID VARIABLE NAMES ARE "SALARY OF EMP" AND "AVG."

Q38. TOM WORKS AT AN AQUARIUM SHOP ON SATURDAYS. ONE SATURDAY, WHEN TOM GETS TO WORK, HE IS ASKED TO CLEAN A 175-GALLON REEF TANK. HIS FIRST JOB IS TO DRAIN THE TANK. HE PUTS A HOSE INTO THE TANK AND STARTS A SIPHON. TOM WONDERS IF THE TANK WILL FINISH DRAINING BEFORE HE LEAVES WORK. HE MEASURES THE AMOUNT OF WATER THAT IS DRAINING OUT AND FINDS THAT 12.5 GALLONS DRAIN OUT IN 30 MINUTES. SO, HE FIGURES THAT THE RATE IS 25 GALLONS PER HOUR. DEVELOP A ‘C’ PROGRAM TO HELP TOM TO CALCULATE TIME REQUIRED TO COMPLETELY CLEAN TANK.

ANS38.   
TO HELP TOM CALCULATE THE TIME REQUIRED TO COMPLETELY CLEAN THE TANK, YOU CAN USE THE FORMULA:

TIME (IN HOURS)=TOTAL GALLONS IN THE TANKRATE (GALLONS PER HOUR)TIME (IN HOURS)=RATE (GALLONS PER HOUR)TOTAL GALLONS IN THE TANK​

HERE, THE TOTAL GALLONS IN THE TANK IS 175 GALLONS, AND THE RATE IS 25 GALLONS PER HOUR. YOU CAN CREATE A 'C' PROGRAM TO CALCULATE THIS AS FOLLOWS:

#INCLUDE <STDIO.H>

INT MAIN()

{

DOUBLE TOTALGALLONS = 175.0; // TOTAL GALLONS IN THE TANK

DOUBLE RATE = 25.0; // RATE OF DRAINING (GALLONS PER HOUR)

// CALCULATE THE TIME REQUIRED IN HOURS

DOUBLE TIMEREQUIRED = TOTALGALLONS / RATE;

PRINTF("TIME REQUIRED TO COMPLETELY CLEAN THE TANK: %.2F HOURS\N", TIMEREQUIRED);

RETURN 0;

}

Q39. THE PERCENT Y (IN DECIMAL FORM) OF BATTERY POWER REMAINING X HOURS AFTER YOU TURN ON A LAPTOP COMPUTER IS Y = −0.2 X + 1. DEVELOP A ‘C’ PROGRAM TO CALCULATE AFTER HOW MANY HOURS THE BATTERY POWER IS AT 75%?

ANS 39. INT MAIN() {

DOUBLE BATTERYPOWER = 0.75; // 75% IN DECIMAL FORM

DOUBLE X; // NUMBER OF HOURS

// SOLVE FOR X USING THE GIVEN EQUATION Y = -0.2X + 1

// REARRANGE THE EQUATION: -0.2X = Y - 1

// THEN, X = (1 - Y) / 0.2

X = (1 - BATTERYPOWER) / 0.2;

PRINTF("THE BATTERY POWER IS AT 75%% AFTER %.2F HOURS\N", X);

RETURN 0;

}

Q40.WHICH OF THE FOLLOWING IS USED TO CONVERT THE HIGH LEVEL LANGUAGE IN MACHINE LANGUAGE IN A SINGLE GO?

A. COMPILER B.INTERPRETER

C. LINKER D.ASSEMBLER

ANS40. (A). COMPILER

A COMPILER IS USED TO CONVERT A HIGH-LEVEL PROGRAMMING LANGUAGE CODE INTO MACHINE CODE (ALSO KNOWN AS OBJECT CODE) IN A SINGLE GO

Q 41. WHAT IS THE FORMAT SPECIFIER FOR AN OCTAL NUMBER?

A.%0 B.%D

C. %O D. %E

ANS41. C. %O

THE FORMAT SPECIFIER FOR AN OCTAL NUMBER IN C IS %O. WHEN YOU USE %O IN A PRINTF OR SCANF STATEMENT, IT REPRESENTS THAT THE CORRESPONDING ARGUMENT IS AN OCTAL NUMBER.

Q 42. WHICH FORMAT SPECIFIER IS USED TO PRINT THE EXPONENT VALUE UPTO 2 DECIMAL PLACES.

A. %E B.%.2F C. %F D.%.2E

ANS42. D. %.2E

THE FORMAT SPECIFIER %.2E IS USED TO PRINT THE EXPONENT VALUE (IN SCIENTIFIC NOTATION) WITH TWO DECIMAL PLACES.

Q 43. WHICH OF THE FOLLOWING IS NOT A BASIC DATA TYPE?

A. CHAR

B. ARRAY

C. FLOAT

D. INT

ANS43. B. ARRAY

AN ARRAY IS NOT A BASIC DATA TYPE. IT IS A COMPOSITE OR DERIVED DATA TYPE IN C AND MANY OTHER PROGRAMMING LANGUAGES. ARRAYS ARE USED TO STORE COLLECTIONS OF ELEMENTS OF THE SAME DATA TYPE, BUT THEY ARE NOT CONSIDERED ONE OF THE BASIC DATA TYPES LIKE CHAR, INT, OR FLOAT.

Q 44. WHAT IS THE OUTPUT OF FOLLOWING CODE?

#INCLUDE<STDIO.H>

VOID MAIN()

{

INT X=0;

X= PRINTF("\"HELLO\B\"");

PRINTF(“%D”,X);

}

A. HELLO7 B. “HELLO”7 C. “HELL”8 D. HELL8

ANS44. LET'S ANALYZE BOTH CODE SNIPPETS:

THE CODE SNIPPET CONTAINS A COUPLE OF ISSUES:

- THE `PRINTF` STATEMENT WITHIN `X` ASSIGNMENT IS ENCLOSED IN DOUBLE QUOTES, WHICH SHOULD BE SINGLE QUOTES FOR CHARACTER CONSTANTS.

#INCLUDE <STDIO.H>

INT MAIN() {

INT X = 0;

X = PRINTF("\"HELLO\\B\"");

PRINTF("%D", X);

}

NOW, LET'S BREAK IT DOWN:

- `PRINTF("\"HELLO\\B\"")` WILL PRINT "HELLO\B" BECAUSE `\B` REPRESENTS A BACKSPACE CHARACTER.

- `X = PRINTF("\"HELLO\\B\"")` SETS `X` TO THE NUMBER OF CHARACTERS PRINTED BY THE `PRINTF` STATEMENT, WHICH IS 8 (INCLUDING THE BACKSPACE CHARACTER).

- FINALLY, `PRINTF("%D", X);` PRINTS THE VALUE OF `X`,

WHICH IS 8. ANSWER IS D. `HELL8`.

Q 45. WHAT IS THE OUTPUT OF FOLLOWING CODE?

#INCLUDE<STDIO.H>

VOID MAIN()

{

INT B,C=5 ;

INT(“%D , %D”, B,C);

}

A. 5, 5 B. 5, 5.000000

C. GARBAGE, 5.000000 D. GARBAGE, 5

ANS45. THE CODE SNIPPET CONTAINS MULTIPLE ISSUES:

- THE `INT` KEYWORD BEFORE THE `PRINTF` STATEMENT IS INCORRECT.

THE FORMAT STRING IS INCORRECTLY PLACED INSIDE `INT`.

HERE'S THE CORRECTED CODE:

#INCLUDE <STDIO.H>

INT MAIN() {

INT B, C = 5;

PRINTF("%D , %D", B, C);

}

NOW, LET'S BREAK IT DOWN:

- `INT B` DECLARES AN INTEGER VARIABLE `B` BUT DOES NOT INITIALIZE IT, SO IT CONTAINS A GARBAGE VALUE.

- `C` IS INITIALIZED TO 5.

- `PRINTF("%D , %D", B, C);` PRINTS THE VALUES OF `B` AND `C`, WHICH ARE GARBAGE FOR `B` AND 5 FOR `C`.

CORRECT ANSWER IS D. "GARBAGE, 5".

Q46. WHICH OF THE FOLLOWING IS AN IDENTIFIER?

A. &FACT B. BASIC\_PAY C. ENUM D. 1SUM

ANS46. B. BASIC\_PAY

Q 47. WHAT IS THE OUTPUT OF THE FOLLOWING PROGRAM?

#INCLUDE<STDIO.H>

VOID MAIN()

{

CHAR X, A=’C’;

X=PRINTF("%C",A);

PRINTF(“%D”,X);

}

A. C1 B. CGARBAGE

C. 1 C. C

ANS47. #INCLUDE<STDIO.H>

INT MAIN()

{

CHAR X, A = 'C';

X = PRINTF("%C", A);

PRINTF("%D", X);

}

RETURN 0;

}

SO, THE CORRECT ANSWER IS: C. 1

Q48. PERFORM THE FOLLOWING CONVERSION FROM DECIMAL TO OTHER NUMBER AS DIRECTED-

1. (365.55)10 = (?)2
2. (453.65)10 = (?)8
3. (5164.12)10 = (?)16
4. (23.65)10 = (?)5
5. (772)10 = (?)7

ANS 48. CONVERSION FROM DECIMAL TO OTHER NUMBER SYSTEMS:

A) (365.55)10 = (?)2

IN BINARY (BASE 2): 101101101.11

B) (453.65)10 = (?)8

IN OCTAL (BASE 8): 705.52

C) (5164.12)10 = (?)16

IN HEXADECIMAL (BASE 16): 142C.2A

D) (23.65)10 = (?)5

IN BASE 5: 113.3

E) (772)10 = (?)7

IN BASE 7: 1544

Q49. COVERT THE FOLLOWING NUMBERS TO DECIMAL NUMBER SYSTEM-

1. (325.54)6 = (?)10
2. (1001010110101.1110101)2 = (?)10
3. (742.72)8 = (?)10
4. (AC94.C5)16 = (?)10

ANS 49. CONVERSION TO DECIMAL FROM OTHER NUMBER SYSTEMS:

A) (325.54)6 = (?)10

IN DECIMAL: 3\*6^2 + 2\*6^1 + 5\*6^0 + 5\*6^-1 + 4\*6^-2 = 183.91666666666666 (ROUNDED TO 6 DECIMAL PLACES)

B) (1001010110101.1110101)2 = (?)10

IN DECIMAL: 1\*2^12 + 0\*2^11 + 0\*2^10 + 1\*2^9 + 0\*2^8 + 1\*2^7 + 0\*2^6 + 1\*2^5 + 1\*2^4 + 0\*2^3 + 1\*2^2 + 0\*2^1 + 1\*2^0 + 1\*2^-1 + 1\*2^-2 + 1\*2^-3 + 0\*2^-4 + 1\*2^-5 = 4917.84375

C) (742.72)8 = (?)10

IN DECIMAL: 7\*8^2 + 4\*8^1 + 2\*8^0 + 7\*8^-1 + 2\*8^-2 = 498.875

D) (AC94.C5)16 = (?)10

IN DECIMAL: A\*16^4 + C\*16^3 + 9\*16^2 + 4\*16^1 + C\*16^0 + 5\*16^-1 = 44116.7734375

Q50. PERFORM THE FOLLOWING CONVERSION FROM HEXADECIMAL TO OTHER NUMBER AS DIRECTED-

(DB56.CD4)16 = (?)2, (?)8, (?)4

ANS 50. CONVERSION FROM HEXADECIMAL TO OTHER NUMBER SYSTEMS:

(DB56.CD4)16 = (?)2

IN BINARY: 1101101101010110.110011010100

(DB56.CD4)16 = (?)8

IN OCTAL: 333265.6345

(DB56.CD4)16 = (?)4

IN QUATERNARY: 30302321232222.3032030232111

Q51. PERFORM THE FOLLOWING CONVERSION FROM OCTAL TO OTHER NUMBER AS DIRECTED-

(473.42)8 = (?)2, (?)10, (?)16, (?)5

ANS 51. CONVERSION FROM OCTAL TO OTHER NUMBER SYSTEMS:

(473.42)8 = (?)2

IN BINARY: 1001110011.100

(473.42)8 = (?)10

IN DECIMAL: 315.25

(473.42)8 = (?)16

IN HEXADECIMAL: 13B.68

(473.42)8 = (?)5

IN BASE 5: 132.2

Q52. FIND THE VALUE OF A?

1. (23)10 = (17)A
2. (21)16 = (41)A
3. (32)8 = (101)A

ANS 52. TO FIND THE VALUE OF A IN EACH CASE, WE NEED TO CONVERT THE NUMBERS TO THE DECIMAL SYSTEM:

A) (23)10 = (17)A

CONVERT (23)10 TO BASE A:

2A + 3 = 17

2A = 17 - 3

2A = 14

A = 14 / 2

A = 7

SO, (23)10 = (17)7.

B) (21)16 = (41)A

CONVERT (21)16 TO BASE A:

2A + 1 = 4A + 1

2A = 4A

THIS EQUATION DOESN'T HOLD TRUE FOR ANY POSITIVE INTEGER VALUE OF A, SO THERE IS NO VALID VALUE OF A THAT SATISFIES THIS EQUATION.

C) (32)8 = (101)A

CONVERT (32)8 TO BASE A:

3A + 2 = A2 + 1

A2 - 3A - 1 + 2 = 0

A2 - 3A - 1 = 0

SOLVE THIS QUADRATIC EQUATION FOR A. USING THE QUADRATIC FORMULA:

A = [3 ± √(32 - 4 \* 1 \* (-1))] / (2 \* 1)

A = [3 ± √(13)] / 2

A ≈ 2.30 OR A ≈ -1.30

SO, THERE ARE TWO POSSIBLE VALUES FOR A: A ≈ 2.30 AND A ≈ -1.30.

Q53: WHAT WILL BE THE OUTPUT OF FOLLOWING PROGRAM? ASSUME INTEGER IS OF 2 BYTES

VOID MAIN(){

INT A=32770;

PRINTF(“%D”,A);

}

ANS 53. THE OUTPUT OF THE GIVEN PROGRAM WILL BE:

32766

- THE VARIABLE `A` IS OF TYPE `INT`, WHICH IS TYPICALLY 2 BYTES IN SIZE ON MANY SYSTEMS.

- THE VALUE `32770` EXCEEDS THE MAXIMUM POSITIVE VALUE THAT CAN BE REPRESENTED BY A 2-BYTE `INT`, WHICH IS 32,767.

- WHEN YOU EXCEED THE MAXIMUM VALUE, IT CAN LEAD TO INTEGER OVERFLOW, AND THE BEHAVIOR IS UNDEFINED.

- IN THIS CASE, THE OVERFLOW CAUSES `A` TO WRAP AROUND TO A NEGATIVE VALUE, SPECIFICALLY `-32766`.

SO, WHEN YOU PRINT `A`, IT DISPLAYS THE NEGATIVE VALUE `-32766`.

Q54: #INCLUDE <STDIO.H>

INT MAIN()

{

FLOAT C = 5.0;

PRINTF ("TEMPERATURE IN FAHRENHEIT IS %.2F", (9/5)\*C + 32);

RETURN 0;

}

ANS 54. THE PROVIDED C PROGRAM IS DESIGNED TO CONVERT A TEMPERATURE IN CELSIUS TO FAHRENHEIT AND PRINT THE RESULT. HOWEVER, THERE'S AN ISSUE WITH THE CALCULATION BECAUSE THE EXPRESSION (9/5)\*C USES INTEGER DIVISION, WHICH WILL RESULT IN UNEXPECTED OUTPUT. TO FIX THIS, YOU SHOULD MAKE SURE TO USE FLOATING-POINT CONSTANTS FOR THE DIVISION. HERE'S THE CORRECTED PROGRAM:

#INCLUDE <STDIO.H>

INT MAIN() {

FLOAT C = 5.0;

PRINTF("TEMPERATURE IN FAHRENHEIT IS %.2F", (9.0 / 5.0) \* C + 32.0);

RETURN 0;