

PF ASSIGNMENT#01

NAME: MUHAMMAD AZMAT

ROLL NO: 23i-2651

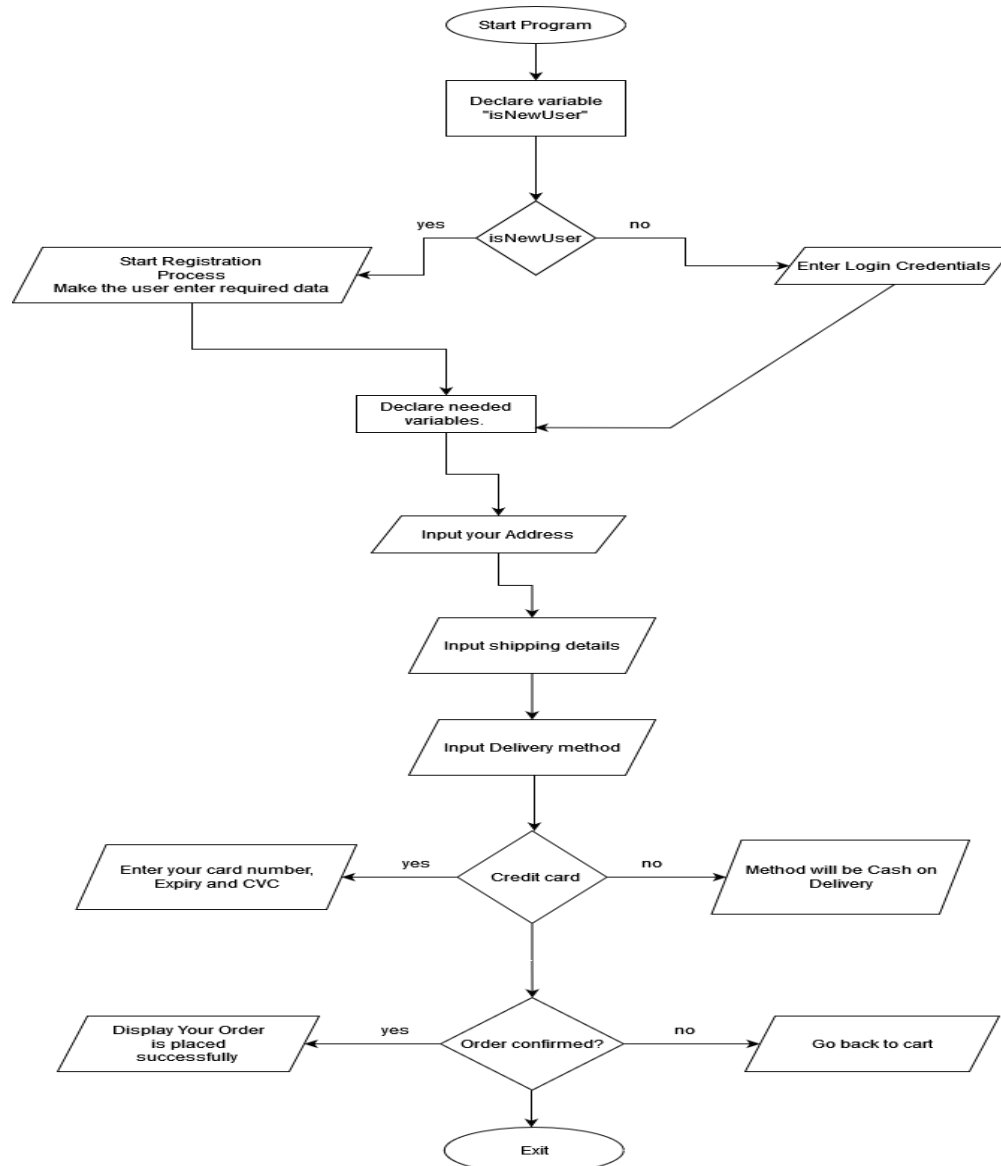
DEPARTMENT: BS-DS-C

QUESTION#01:

1. Start the program.
2. Declare a Boolean variable, “isNewUser”.
3. If yes, then go to the registration process.
4. Make user enter their full name, cell number, email address etc.
5. Otherwise ask the user to enter their login credentials.
6. After the signup/login process,
7. Declare 3 variables related to Product Specs which will display the order summary in cart.
8. Ask the user to provide their address.
9. Confirm the shipping details of user.
10. Declare the variable “selectedPaymentMethod” now.
11. Ask the user for payment method.
12. If user selects Credit Card option

13. Input card number, Expiry date and CVC.
14. Otherwise it will be cash on delivery.
15. Declare a Boolean variable, “order Confirmed”.
16. If true, then display your order is placed successfully.
17. Otherwise go back
18. Exit the program.

FLOWCHART:



QUESTION#02:

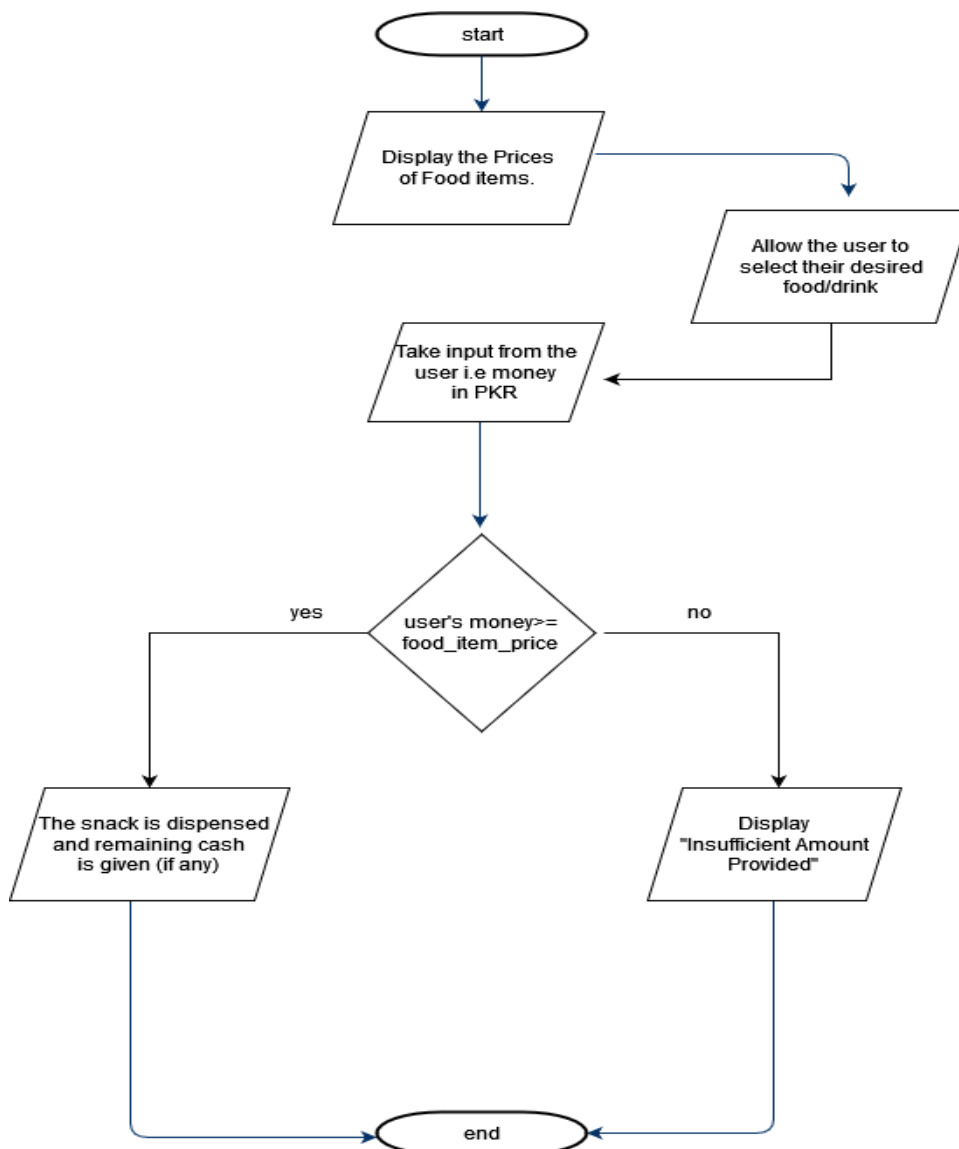
1. Start the program.
2. Declare variables as needed.
3. Display output telling users to input their Credit Scores (CS), Monthly Income (MI) and Loan Amount (LA).
4. if $CS \geq 700 \ \&\& \ MI \geq 500 \ \&\& \ LA \leq 5000$, then the Loan Amount will be Approved.
5. Otherwise, if one or more conditions don't match, then the Loan Amount will be denied.
6. Display message "Loan Amount Decision: LA"
7. Exit the program.

QUESTION#03:

1. Start the program.
2. Display different food items along with their associated prices to the user.
3. Declare as many food variables as needed assigning them specific traits. For instance:
4. `int lays = 30;`
5. `int juice = 20;`
6. `int drink = 100;`
7. The vending machine should be able to take input from the user i.e. money in PKR.
8. If user's money \geq food_item_price

9. Then the snack should be dispensed to the user and the remaining amount (if any) will be handed back.
10. If user's money < food_item_price
11. Then the program should display a statement saying, "Insufficient Amount provided."
12. End.

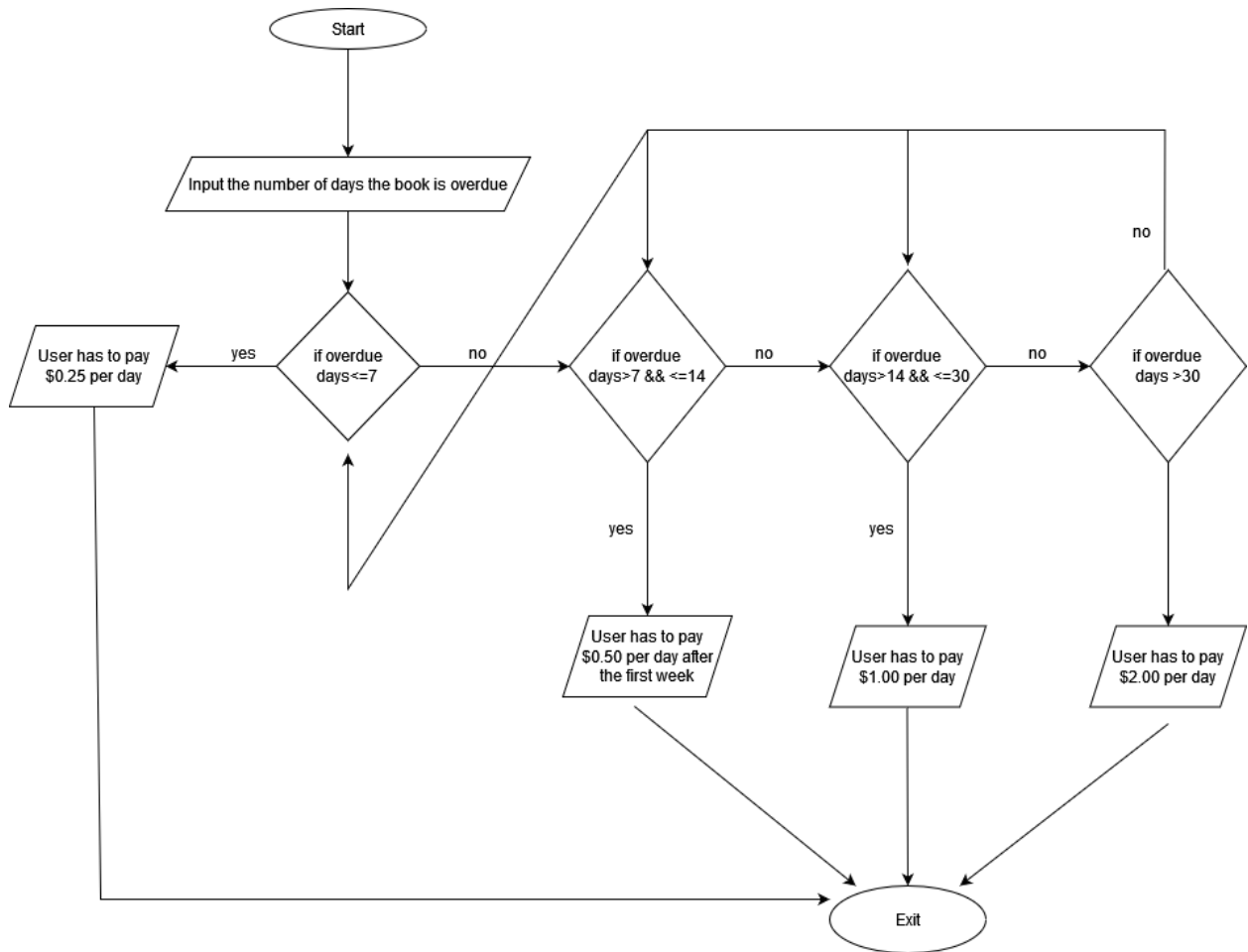
FLOWCHART:



QUESTION#04:

1. Start the Program.
2. Ask the user to input the number of days the book is overdue.
3. if overdue days ≤ 7 , then \$0.25 per day
4. Multiply the penalty fee with the number of days, e.g. for 6 days, it'll be 0.25×6
5. If overdue days > 7 & ≤ 14 , then \$0.50 per day after the first 7 days in which the penalty will be \$0.25
6. Multiply the penalty fee with no. of days, e.g. for 10 days, the charge will be $(0.25 \times 7) + (0.50 \times 3)$
7. If overdue days > 14 & ≤ 30 , then \$1.00 per day, in the first 7 days the fee will be \$0.25. From day 8-14, it'll be \$0.50. Afterwards, it'll be \$1.00
8. Multiply the penalty fee with the number of days, e.g. for 27 days, it'll be $(0.25 \times 7) + (0.50 \times 7) + (1.00 \times 12)$
9. If overdue days > 30 , then the penalty will be \$2.00 per day. The charge in first week is \$1.00, in 2nd week, it is \$0.50 and after the 14th day till 30th, it's \$1.00. After the 30th day has passed, the amount will be \$2.00 per day
10. Multiply the specified penalty fee with no. of days, e.g. for 40 days, the charge will be $(0.25 \times 7) + (0.50 \times 7) + (1.00 \times 15) + (2.00 \times 10)$
11. Exit the Program.

FLOWCHART:

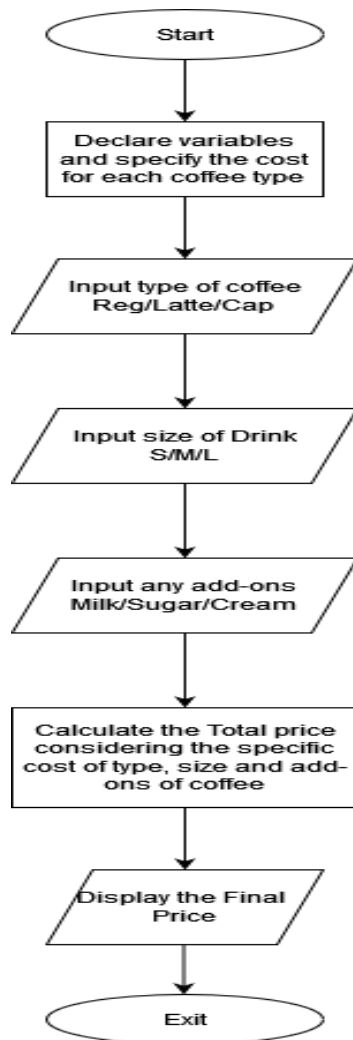


QUESTION#05:

1. Start the Program.
2. Declare variables and specify the cost for each coffee type along with add-ons.
3. Ask the user to input the type of coffee they want (Regular or Latte or Cappuccino)
4. Ask the user to input the size of drink they require (S/M/L)
5. Input from the user any add-ons they might like (Milk, Sugar or Whipped Cream)

6. Calculate the price and display the total to the user, e.g. If a user demands a “Small Cappuccino” along the “Whipped Cream” add-on, his/her total will be $\$4.00 + \$0.75 = \$4.75$
7. Exit the Program.

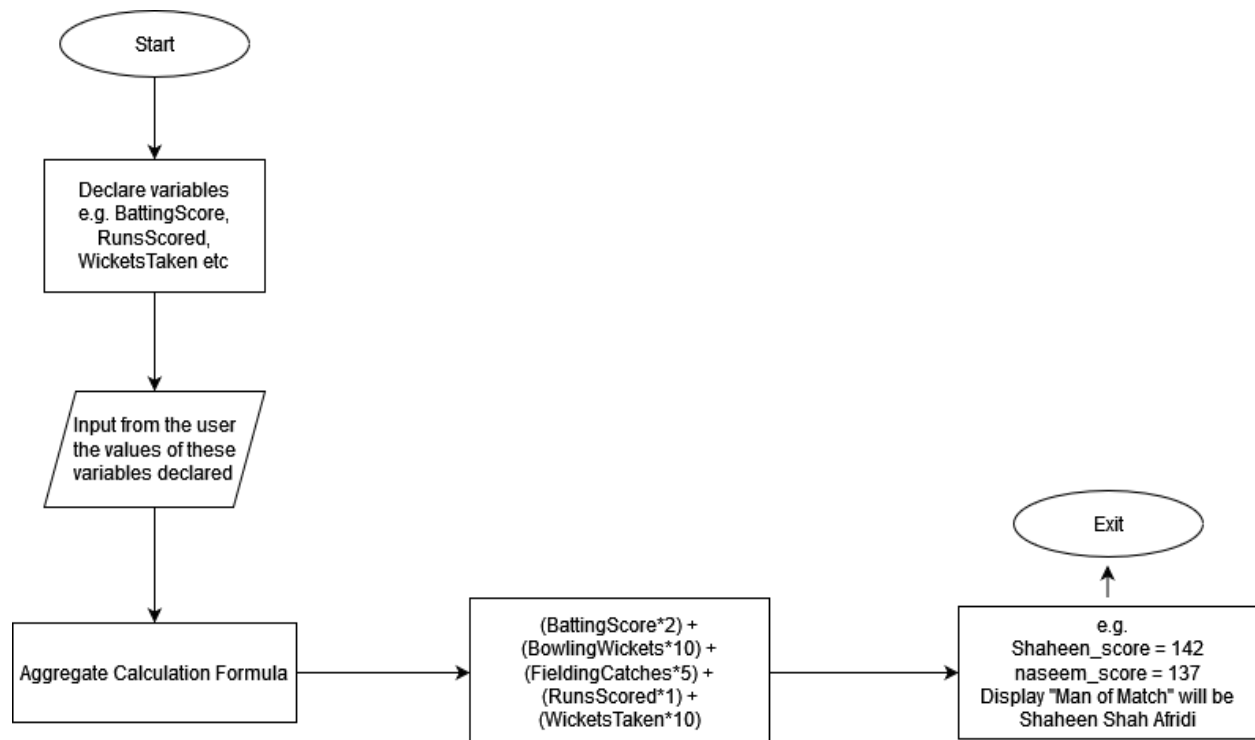
FLOWCHART:



QUESTION#06:

1. Start the Program.
2. Declare Variables, e.g. BattingScore, RunsScored, WicketsTaken.
3. Ask the user to input the values of all the variables which decides who the “Man of Match” will be.
4. Each Player’s score will be calculated on the following criteria.
5. $\text{Player_Score_Naseem} = (\text{BattingScore} * 2) + (\text{BowlingWickets} * 10) + (\text{FieldingCatches} * 5) + (\text{RunsScored} * 1) + (\text{WicketsTaken} * 10)$
6. $\text{Player_Score_Shaheen} = (\text{BattingScore} * 2) + (\text{BowlingWickets} * 10) + (\text{FieldingCatches} * 5) + (\text{RunsScored} * 1) + (\text{WicketsTaken} * 10)$
7. By this composition formula, calculate and compare the scores of the given 2 players.
8. Player with the highest score is the “Man of Match”
9. Display the player’s name along with his score calculated earlier (Player_Score)
10. Exit the Program.

FLOWCHART:



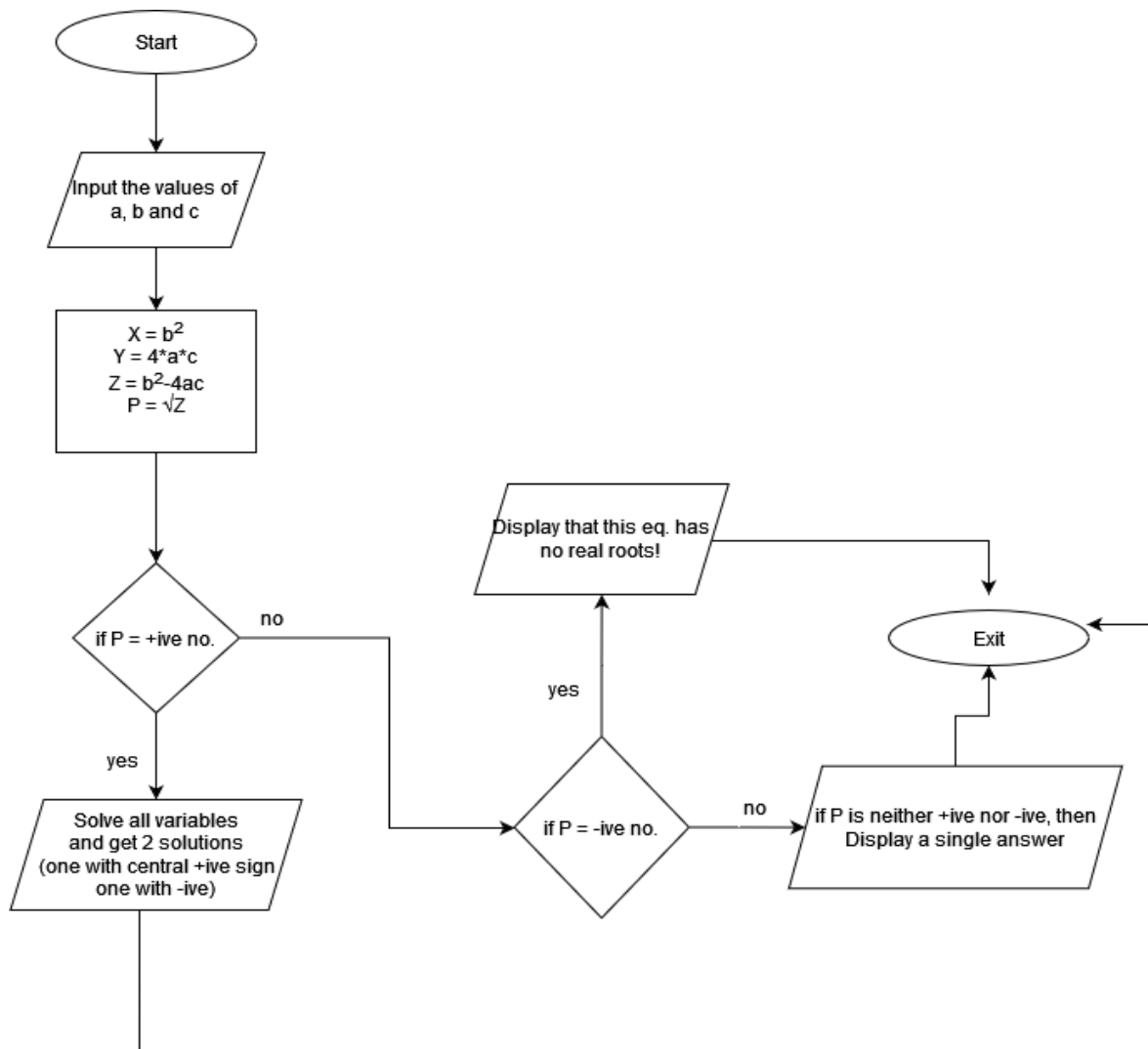
QUESTION#07:

(A):

1. Start the Program.
2. Ask the user to input the values of a, b and c.
3. Declare a variable X which stores the value of b^2
4. Declare a variable Y which stores the value of $4*a*c$
5. Declare a variable Z which calculates and then stores the value of $(b^2 - 4ac)$
6. Declare another variable P which takes the square root of Z, i.e. \sqrt{Z}
7. If $P ==$ a negative number then display that this equation has no real roots and Exit Program.

8. If $P = 0$, then display a single answer and Exit Program.
9. If $P > 0$, Solve one time with positive sign in between and second with the central sign as negative.
10. Display both answers.
11. Exit the Program.

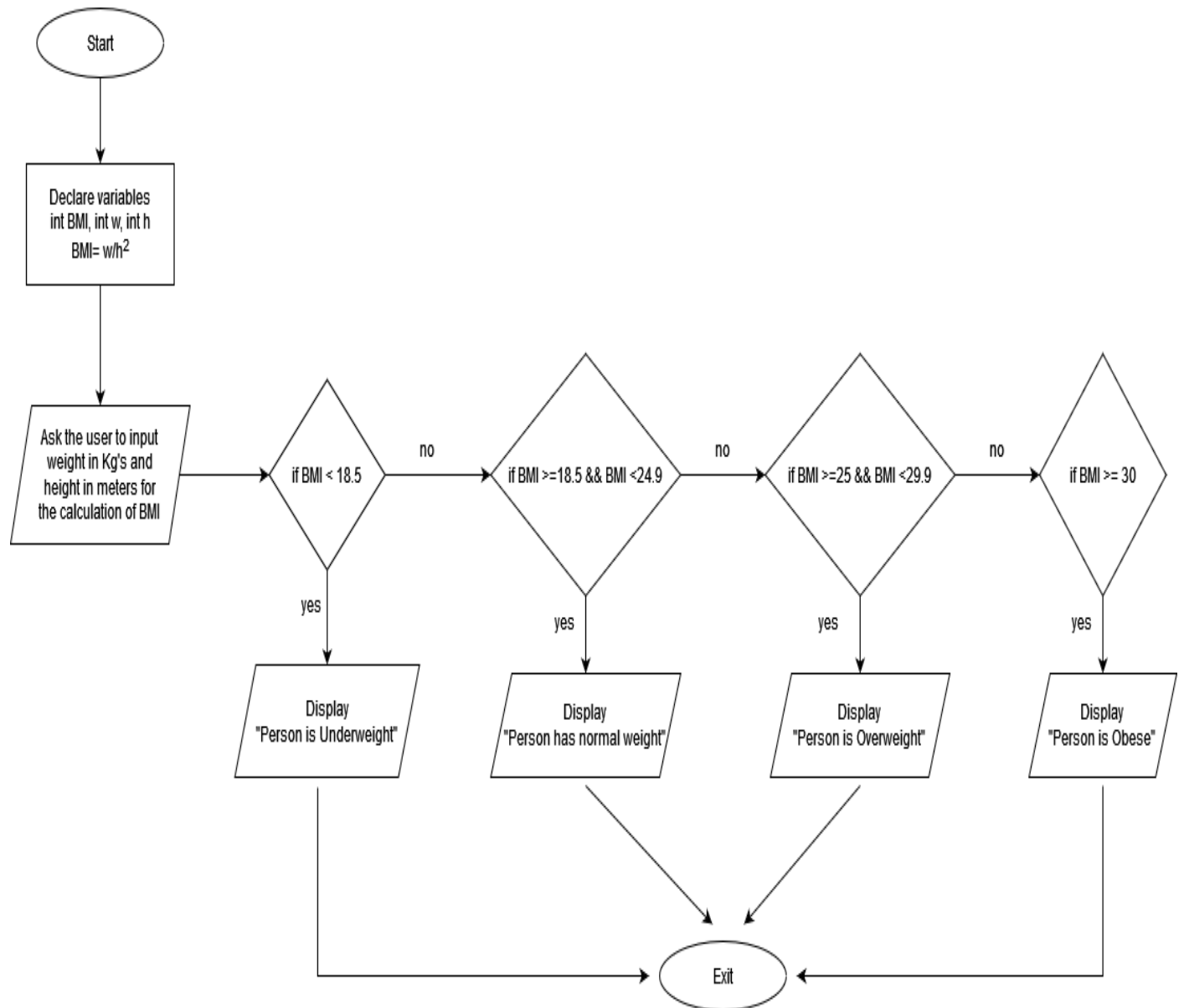
FLOWCHART:



(B):

1. Start the Program.
2. Declare the variables, int BMI, int w, int h.
3. $BMI = w/h^2$
4. Ask the user to input their weight in “Kg’s” and their height in “meters”
5. If answer < 18.5 , then display that the person is ‘Underweight’ along with BMI value.
6. If answer ≥ 18.5 & < 24.9 , then display that person has ‘Normal weight’ with BMI value.
7. If answer ≥ 25 & < 29.9 , display that person is ‘Overweight’ alongside BMI value.
8. If answer ≥ 30 , then display that person is ‘Obese’ in addition to their BMI value
9. Exit the Program.

FLOWCHART:

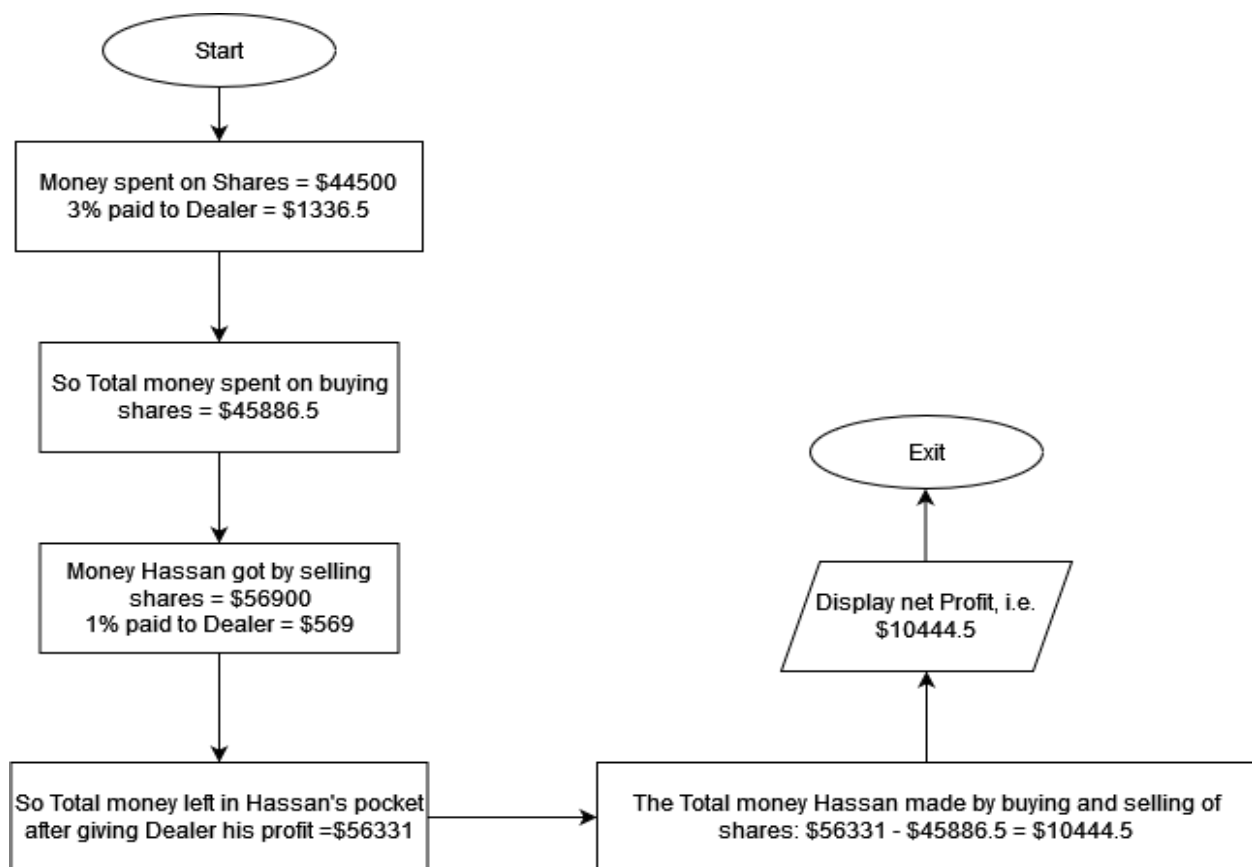


QUESTION#08:

1. Start the Program.
2. Total number of money Hassan spent on shares $\rightarrow \$49.50 * 900 = \44550

3. 3% paid to stockbroker will be = \$1336.5
4. Total money gone from Hassan's pocket = \$45886.5
5. Money Hassan got by selling shares $\rightarrow \$56.90 * 1000 = \56900
6. Money paid to stockbroker = 1% of total = \$569
7. Total money left in Hassan's pocket = $56900 - 569 = \$56331$
8. So Hassan Grand Total profit will be: $\$56331 - \$45886.5 = \$10444.5$
9. Exit the Program

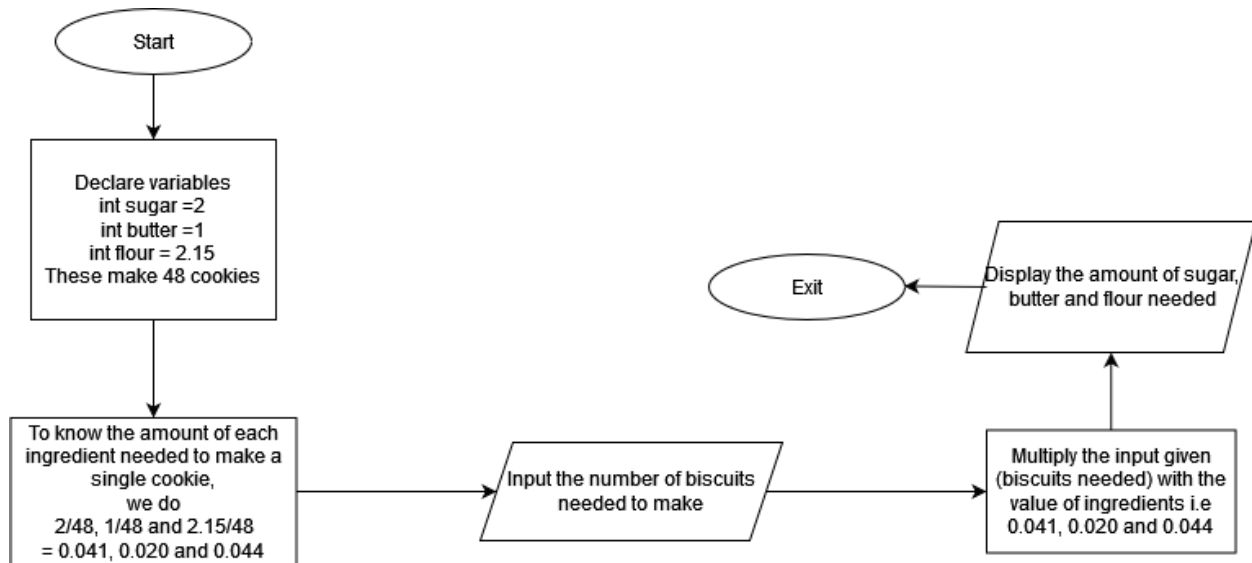
FLOWCHART:



QUESTION#09:

1. Start the Program.
2. Declare variables and store in them, the ingredients quantity required for baking of cookies.
3. $\text{int sugar} = 2$, $\text{int butter} = 1$, $\text{int flour} = 2.15 \rightarrow$ they produce 48 cookies.
4. $\text{int cookies} = \text{sugar} + \text{butter} + \text{flour}$
5. Ask the user to input the number of Cookies they want to make.
6. Now find the number of ingredients needed to make a single cookie
7. i.e. $2/48$ and $1/48$ and $2.15/48$
8. Multiply each value with x (user inputted value for the number of biscuits) to get required answer.
9. According to the number given in Input, Display the quantity of each ingredient needed.
10. Exit the Program.

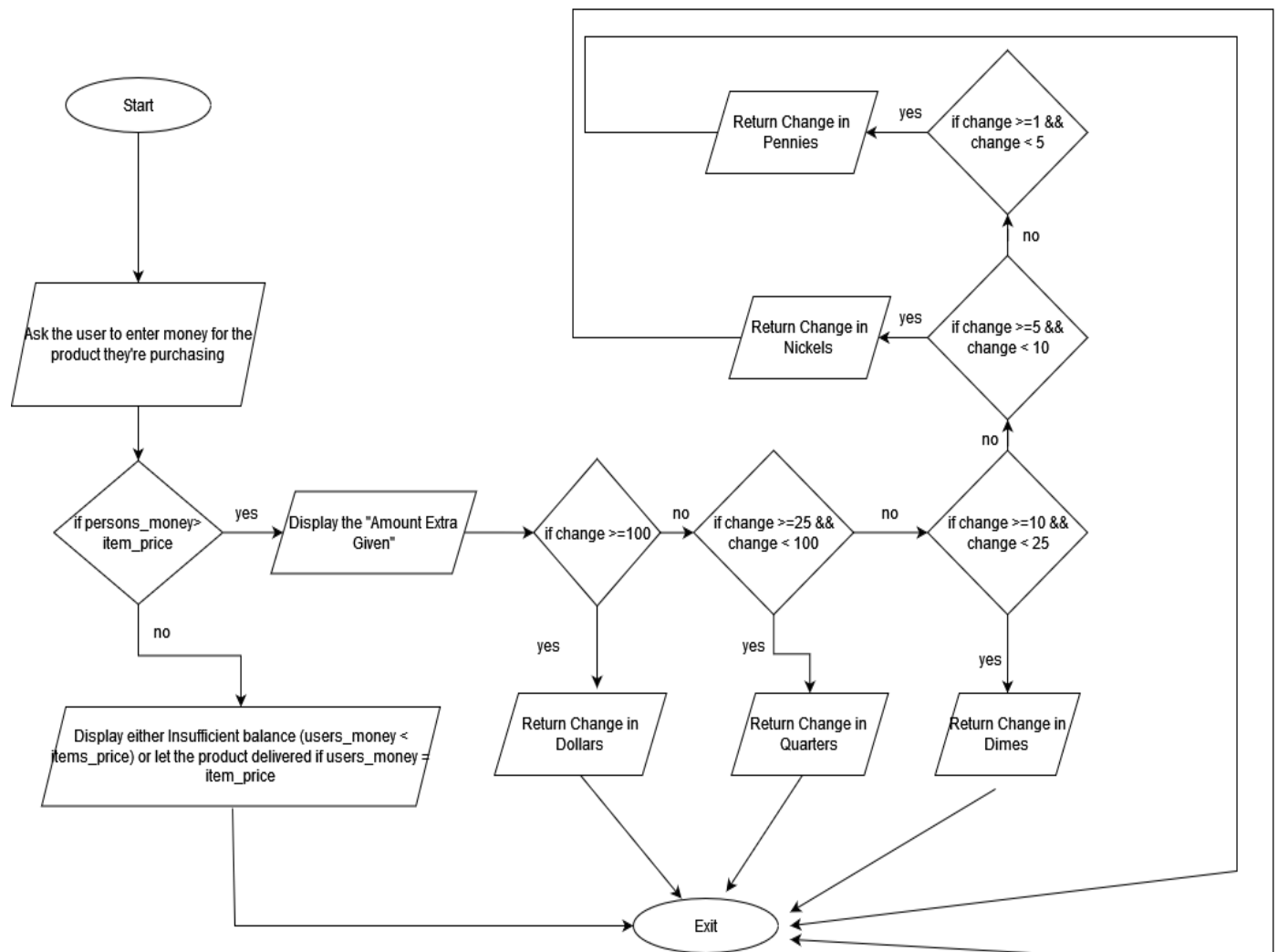
FLOWCHART:



QUESTION#10:

1. Start the Program.
2. Ask the user to enter the money for the product they are going to buy
3. If persons_money = item_price, then display, “Amount is received”
4. If persons_money > item_price, display the “Amount extra given” and Program has to deliver the left charge.
5. If change >= 100, then return in Dollars.
6. If change >=25 && change <100, return in Quarters.
7. If change >=10 && change < 25, return in Dimes.
8. If change >=5 && change < 10, return in Nickels.
9. If change >= 1 && change < 5, return in Pennies.
10. E.g. if the person has paid \$2 (200 cents) and the item price was only 150 cents, then the Program should return 2 quarters i.e. (50 cents = 2 *25 cents)
11. Exit the Program.

FLOWCHART:



QUESTION#11:

(A): 10

1. Start the Program.
2. Set $dN = 0$, $bR = ""$ and $oR = ""$
3. Now enter the number i.e 10
4. So $dN = 10$ and as temp is equal to dN so $temp = 10$
5. If ($temp > 0$) i.e ($10 > 0$)
6. Set $remainder = temp \% 2 = 0$
7. Set $character = char remainder = char (0)$
8. Set $temp = temp/2 = 10/2 = 5$
9. $bR.append(character)$, we know bR is empty and $char = 0$, so
10. $bR.append(character) = 0$
11. Now go again to if statement i.e now ($5 > 0$)
12. Set $remainder = temp \% 2 = 1$
13. Set $character = char (remainder) = char (1)$
14. Set $temp = temp/2 = 5/2 = 2.5 = 2$ (as int data type is used)
15. $bR.append(character) = 01$
16. Again go to if statement i.e now ($2 > 0$) so
17. Set $remainder = temp \% 2 = 0$
18. Set $character = char (remainder) = char (0)$
19. Set $temp = temp/2 = 2/2 = 1$

20. `bR.append(character) = 010`
21. Head back to if statement i.e. now $(1 > 0)$
22. Set `remainder = temp % 2 = 1 % 2 = 1`
23. Set `character = char (remainder) = char (1)`
24. Set `temp = temp/2 = 1/2 = 0.5 = 0` (as data type is integer)
25. `bR.append(character) = 0101`
26. Now if we head back to the if statement, temp is not greater than 0
27. Now move downwards. Here another if situation comes, set `temp = dN` , means
 $dN = 10$ and so `temp = 10` as well.
28. `Temp > 0` means $10 > 0$ and this is true so
29. Set `remainder = temp % 8 = 10 % 8 = 2`
30. Set `character = char (remainder) = char (2)`
31. Set `temp = temp/8 = 10/8 = 1.25 = 1` (as int is datatype)
32. `oR.append(character) = 2`
33. Go back to if statement, here again `temp > 0` i.e. $1 > 0$
34. Set `remainder = temp % 8 = 1 % 8 = 1`
35. Set `character = char (remainder) = char (1)`
36. Set `temp = temp/8 = 1/8 = 0.125 = 0` (datatype is int)
37. `oR.append(character) = 21`
38. Again go to if statement, and now the condition is false, so move down,
39. `bR = reverse(bR) = reverse(0101) = 1010`
40. `oR = reverse(oR) = reverse(021) = 12`
41. Exit

(B): 60

1. Start the Program.
2. Set $dN = 0$, $bR = ""$ and $oR = ""$
3. Now enter the number i.e 60
4. So $dN = 60$ and as temp is equal to dN so $temp = 60$
5. If ($temp > 0$) i.e ($60 > 0$)
6. Set $remainder = temp \% 2 = 60 \% 2 = 0$
7. Set $character = char remainder = char (0)$
8. Set $temp = temp/2 = 60/2 = 30$
9. $bR.append(character)$, we know bR is empty and $char = 0$, so
10. $bR.append(character) = 0$
11. Now go again to if statement i.e now ($30 > 0$)
12. Set $remainder = temp \% 2 = 30 \% 2 = 0$
13. Set $character = char (remainder) = char (0)$
14. Set $temp = temp/2 = 30/2 = 15$
15. $bR.append(character) = 00$
16. Again go to if statement i.e now ($15 > 0$) so
17. Set $remainder = temp \% 2 = 15 \% 2 = 1$
18. Set $character = char (remainder) = char (1)$
19. Set $temp = 15/2 = 7.5 = 7$ (as data type is integer)
20. $bR.append(character) = 001$
21. Head back to if statement i.e. now ($7 > 0$)

22. Set remainder = temp % 2 = 7 % 2 = 1

23. Set character = char (remainder) = char (1)

24. Set temp = temp/2 = 7/ 2 = 3.5 = 3 (as data type is integer)

25. bR.append(character) = 0011

26. Now if we head back to the if statement, temp is still greater than 0 i.e. (3 > 0)

27. Set remainder = temp % 2 = 3 % 2 = 1

28. Set character = char(remainder) char (1)

29. Set temp = temp / 2 = 3/2 = 1.5 = 1 (datatype is integer)

30. bR.append(character) = 00111

31. again go to if statement, and it will execute as 1 > 0

32. set remainder = temp % 2 = 1 % 2 = 1

33. set character = char (remainder) = char (1)

34. set temp = temp/2 = 1/ 2 = 0.5 = 0 (as data type is integer)

35. bR.append(character) = 001111

36. now if we go to back to if statement, it wont be true as 0 is not greater than 0.

37. Now move downwards. Here another if situation comes, set temp = dN , means dN = 60
and so temp = 60 as well.

38. Temp > 0 means 60 > 0 and this is true so

39. Set remainder = temp % 8 = 60 % 8 = 4

40. Set character = char (remainder) = char (4)

41. Set temp = temp/8 = 60/8 = 7.5 = 7 (as int is datatype)

42. oR.append(character) = 4

43. Go back to if statement, here again temp > 0 i.e. 7 > 0

44. Set remainder = temp % 8 = 7 % 8 = 7
45. Set character = char (remainder) = char (7)
46. Set temp = temp/8 = 7/8 = 0.875 = 0 (datatype is int)
47. oR.append(character) = 47
48. Again go to if statement, and now the condition is false, so move down,
49. bR = reverse(bR) = reverse(001111) = 111100
50. oR = reverse(oR) = reverse(47) = 74
51. Exit

(C) : 65

1. Start the Program.
2. Set dN = 0, bR = “ ” and oR = “ ”
3. Now enter the number i.e 65
4. So dN = 65 and as temp is equal to dN so temp = 65
5. If (temp > 0) i.e (65 > 0)
6. Set remainder = temp % 2 = 65 % 2 = 1
7. Set character = char remainder = char (1)
8. Set temp = temp/2 = 65/2 = 32.5 = 32 (as data type is integer)
9. bR.append (character), we know bR is empty and char = 1, so
10. bR.append(character) = 1
11. Now go again to if statement i.e now (32 > 0)

12. Set remainder = temp % 2 = 32 % 2 = 0
13. Set character = char (remainder) = char (0)
14. Set temp = temp/2 = 32/2 = 16
15. bR.append(character) = 10
16. Again go to if statement i.e now (16 > 0) so
17. Set remainder = temp % 2 = 16 % 2 = 0
18. Set character = char (remainder) = char (0)
19. Set temp = 16/2 = 8
20. bR.append(character) = 100
21. Head back to if statement i.e. now (8 > 0)
22. Set remainder = temp % 2 = 8 % 2 = 0
23. Set character = char (remainder) = char (0)
24. Set temp = 8/2 = 4
25. bR.append(character) = 1000
26. Now if we head back to the if statement, temp is still greater than 0 i.e. (4 > 0)
27. Set remainder = temp % 2 = 4 % 2 = 0
28. Set character = char(remainder) char (0)
29. Set temp = temp / 2 = 4/2 = 2
30. bR.append(character) = 10000
31. again go to if statement, and it will execute as 2 > 0
32. set remainder = temp % 2 = 2 % 2 = 0
33. set character = char (remainder) = char (0)
34. set temp = temp/2 = 2/ 2 = 1

35. `bR.append(character) = 100000`

36. again go back to if statement and it will be true as $1 > 0$

37. `set remainder = temp % 2 = 1 % 2 = 1`

38. `set character = char (remainder) = char (1)`

39. `set temp = temp/2 = 1/ 2 = 0.5 = 0` (as data type is integer)

40. `bR.append(character) = 1000001`

41. now if we go to back to if statement, it wont be true as 0 is not greater than 0.

42. Now move downwards. Here another if situation comes, `set temp = dN` , means `dN = 65`

and so `temp = 65` as well.

43. `Temp > 0` means $65 > 0$ and this is true so

44. `Set remainder = temp % 8 = 65 % 8 = 1`

45. `Set character = char (remainder) = char (1)`

46. `Set temp = temp/8 = 65/8 = 8.125 = 8` (as int is datatype)

47. `oR.append(character) = 1`

48. Go back to if statement, here again `temp > 0` i.e. $8 > 0$

49. `Set remainder = temp % 8 = 8 % 8 = 0`

50. `Set character = char (remainder) = char (0)`

51. `Set temp = temp/8 = 8/8 = 1`

52. `oR.append(character) = 10`

53. Again, Go back to if statement, here again `temp > 0` i.e. $1 > 0$

54. `Set remainder = temp % 8 = 1 % 8 = 1`

55. `Set character = char (remainder) = char (1)`

56. `Set temp = temp/8 = 1/8 = 0.125 = 0` (as data type is integer)

57. `oR.append(character) = 101`

58. Again go to if statement, and now the condition is false, so move down,

59. `bR = reverse(bR) = reverse(1000001) = 1000001`

60. `oR = reverse(oR) = reverse(101) = 101`

61. Exit the program.

THE END!

