**Algorithm**

**Problem 1:**

**Step 1**: Define a function that converts the numbers to characters in hexadecimal numbers because (A = 10), (B = 11), (C = 12), (D = 13), (E = 14), (F = 15).

**Step 2**: Define a function to use it when we check in octal numbers.

**Step 3**: Define a function to use it when we check in binary numbers.

**Step 4**: Check decimal numbers if decimal number equals one of the characters from A to F or the decimal number is less than zero stop the program but else complete the program normally.

**Step 5**: A condition to verify that the user won’t enter a number except 0s and 1s.

**Step 6**: Check in octal numbers that the user won’t enter characters from A to F or negative numbers or the two numbers 8 and 9.

**Step 7**: Check in hexadecimal numbers if the user entered a negative number stop the program but if the user entered a number from 1 to 9 or from A to F it’s okay.

**Step 8**: Converting from decimal to binary by making a while loop and if (decimal! = 0) go into the while loop and get the reminder and if the reminder equals the integer of the reminder (temp\_binary += “0”) but if the reminder doesn’t equal the integer of the reminder (temp\_binary += “1”) and then defining the decimal number to equal the integer of the reminder and then for loop to reverse the binary number to be correct

**Step 9**: Converting from decimal to octal by checking if decimal doesn’t equal zero then go into the loop and get the reminder and then temp\_octal equals itself plus the string od integer of the reminder minus the integer of the reminder multiplied by eight and then giving initial values for decimal and octal then for loop to reverse.

**Step 10**: Converting from decimal to hexadecimal by making a loop that checks if decimal doesn’t equal zero then go into the loop and get the reminder then convert numbers characters and then reverse the numbers.

**Step 11**: Decimal to decimal will return the same number.

**Step 12**: Define a new function to check the cases from decimal to anything and prints error for invalid operations.

**Step 13**: Define a function that converts from octal to octal by returning the same number.

**Step 14**: Define a function that converts from octal to decimal by using for loop to check every digit and multiply it by 8 powers (i) and then divide the octal number by 10 then return decimal.

**Step 15**: Define a function to convert from octal to binary by writing the eight possible solutions and then return binary.

**Step 16**: Define a function that converts from octal to hexadecimal by converting from octal to binary and then converting from binary to hexadecimal.

**Step 17**: Define a function from octal to anything and put the menus it and check if the user will enter a valid number or operation or not.

**Step 18**: Define a function that converts from binary to binary by returning the same number.

**Step 19**: Define a function that converts from binary to hexadecimal by giving initial values for some variables and then go into a condition if the user entered zero the binary equivalence will be zero but if the user doesn’t enter zero the program will go into a loop to get the reminder and make a condition if the user entered a number except 0s and 1s the program will make the user enter a new number until he enters a number contains 0s and 1s only and them doing some operations to get the hexadecimal number and if the user entered from 10 to 15 this will be converted to (A, B, C, D, E, F) and then return the reverse of hexadecimal.

**Step 20**: Define a function from binary to octal by giving initial values to some variables and make all possible inputs that the user might entera and then return the octal number.

**Step 21**: From binary do decimal by multiplying the number with 2 powers 0 then power 1 and so on.

**Step 22**: Define from binary to anything to get all possible possibilities in all conversions.

**Step 23**: Define a function from hexadecimal to hexadecimal that returns the same number.

**Step 24**: Define a function from hexadecimal to decimal and giving initial values for 10, 11, 12, 13, 14, 15 by converting them to letters from A to F and then go into a loop to convert 10 to A, 11 to B, 12 to C, 13 to D, 14 to E and 15 to F and then getting the end result.

**Step 25**: Define from hexadecimal to binary by going into a while loop and test all possible possibilities and then return binary.

**Step 26**: From hexadecimal to octal by giving initial values for some variables and go into a loop to do some operations and then if statement to return the octal number at the end.

**Step 27**: Define From hexadecimal to anything to put in this function all possible possibilities that the user might enter and find solutions for invalid inputs.

**Step 28**: Define menu1 that says to the user to enter a number or exit the program and put all possible possibilities that the user might enter.

**Step 29**: Define menu2 to select the base the user wants to convert from and put all possible possibilities that the user might enter

**Step 30**: Define menu3 to select the base the user wants to convert to and put all possible possibilities that the user might enter.

**Problem 2:**

**Step 1:** Define a function called stage to check that the user will enter 0s and 1s only and if the user entered another number the program won’t go to the next step until the user enters a valid number.

**Step 2:** Define menu1 to tell the user if he / she wants to insert a number or exit the program and if the user entered a character except A and B the program will make him enter a character again until he enters a valid character.

**Step 3:** Define menu2 to select the operation whether it’s one’s complement, two’s complement, addition, subtraction or multiplication and if the user selected one of these operations the program will go automatically to its part and do the operations of it but if the user entered a letter except (A, B, C, D and M) the program would make him enter again one of the valid characters.

**Step 4:** if the user wants to do the operation of one’s complement first the program will check if the input is valid or not and if the input is invalid the user will enter another number until the input become valid but if the input is valid the for loop will convert any 0 to 1 and any 1 to 0 and then the program will return the result.

**Step 5:** if the user wants to do the operation of two’s complement the program will convert the binary number to one’s complement and then will add one to get the two’s complement by going into a for loop and give initial values for some variables and add one to the one’s complement of the number. when the program adds one to one’s complement of the number if the sum is 1 and 1 the program will write 0 and carry one and if the sum is 1 and 1 and 1 the program will write 1 and carry 1 and so on until the program get the two’s complement and return it.

**Step 6:** If the user wants to get the sum of two numbers the user will choose the letter (C) and will enter the two binary numbers and then the program will go into a while loop and get the sum digit by digit from right to left and the carry with the two digits and the program will go to the second position until the two binary numbers are added.

**Step** **7:** If the user wants to get the subtraction of two binary numbers the user will choose the letter (D) and then the program will ask the user to enter the two binary numbers and then will go into a while loop and subtract them digit by digit from right to left and if the first digit is 0 and the second digit is 1 the first digit will borrow from the number next to it and the program will end when the two binary numbers are subtracted.

**Step 8:** Define a function to multiply the binary numbers and check if the input is a binary number or not. If the input is a binary number the program will exit this condition and complete in the program but if it’s not, the user will enter a number that contains 0s and 1s only and then the program will multiply every digit in the second binary number by the whole first binary number and after multiplying the first digit the program will go to the tens position and start multiplying the second digit and so on. Then the program will get the sum of these numbers at the end and this will be the result of multiplying two numbers.