**Psudocode**

1. Start

2. Declare variables each\_item\_price, starting\_inflation\_rate, total\_price, rateof\_inflation, sarting\_year, budget\_endyear, budget\_year, numberof\_pencile, choice, permmision

3. Ask user for permission to set a budget (permmision)

4. If permmision is 1, ask user for the following requirements:

- Starting year (sarting\_year)

- End year (budget\_endyear)

- Each item price in the starting year (each\_item\_price)

- Estimated inflation rate (starting\_inflation\_rate)

- Quantity of pencils (numberof\_pencile)

5. Convert starting\_inflation\_rate from percent to decimal

6. Calculate total\_price as numberof\_pencile multiplied by each\_item\_price

7. Calculate budget\_year as budget\_endyear minus sarting\_year

8. Declare an array estimated\_price of size budget\_year and a pointer p\_price pointing to the first element of the array

9. Use a for loop to calculate the estimated price for each year and store it in the estimated\_price array using the formula:

- estimated\_price[i] = total\_price + (total\_price \* rateof\_inflation);

- rateof\_inflation \*= 2;

10. Set rateof\_inflation back to starting\_inflation\_rate

11. Clear the screen

12. Ask user for the year they want to see the pencil price for (choice)

13. If choice is not 1, 2 or 3, display an error message and go back to step 12

14. If choice is 1, display a table with the following columns:

- Year

- Number of pencils

- Inflation rate

- Price of each pencil (ETB)

- Estimated price (ETB)

For each year between sarting\_year and budget\_endyear, calculate and display the corresponding values using the formula:

- cout << sarting\_year << setw(15) << numberof\_pencile << setw(15) << rateof\_inflation << setw(20) << each\_item\_price << setw(30) << \*(p\_price+i) << endl;

- sarting\_year++;

- each\_item\_price += each\_item\_price \* rateof\_inflation;

- rateof\_inflation \*= 2;

15. If choice is 2, ask user for the year they want to see the budget for (choiseYear)

16. Use a for loop to search for the year in the estimated\_price array. When the year is found, display a table with the following columns:

- Year

- Number of pencils

- Inflation rate

- Price of each pencil (ETB)

- Estimated price (ETB)

17. If choice is 3, display a thank you message and end the program

18. If permmision is 2, display a thank you message and end the program

19. If permmision is not 1 or 2, display an error message and go back to step 3

20. End

**psudocode**

1. Start the program

2. Create a label "enter\_again"

3. Prompt the user to enter a number greater than 0 or -1 to quit the program

4. Read the input from the user and store it in variable n

5. Create a variable "input\_validation" to store the user's input for validation

6. Check if n is equal to 0, if so, go to validation\_for\_0 label

7. Use a for loop to create the first half of the triangle, starting from n and decrementing until 0

8. Use a nested for loop to print out the numbers in each row, and add spacing where necessary

9. Check if j is equal to i, if so, use another nested for loop to print out the second half of the row

10. Print a new line after each row is printed

11. Use another for loop to create the second half of the triangle, starting from 1 and incrementing until n

12. Use a nested for loop to print out the numbers in each row, and add spacing where necessary

13. Check if j is equal to i, if so, use another nested for loop to print out the second half of the row

14. Print a new line after each row is printed

15. Create a label "validation\_for\_0"

16. Check if n is less than or equal to 0, if so, enter a while loop until n is equal to -1

17. Inside the while loop, inform the user of their input and go back to "enter\_again" label

18. If n is not equal to -1, prompt the user if they want to try again

19. Read the input from the user and store it in "input\_validation" variable

20. Use a switch statement to check if the user wants to try again or not, if so, go back to "enter\_again" label, if not, terminate the program

21. End the program