This is a program that records and displays sales data for a company with 4 salespeople and 5 products. The data is stored in a file named "sales.txt". The program has several functions to write, read, and display the data.

Variables:

- `sales`: A structure that contains an array of 5 floats representing the sales of each product, a float for the total sales, and a float for the bonus. The `product` array stores the total sales for each product, `s_total` stores the total sales for all products, and `bonus` stores the bonus calculated as 5% of the total sales.

- `emp`: An array of 4 `sales` structures representing the sales data for each salesperson. Each element of the array corresponds to a salesperson and contains their sales data.

- `sale`: A file stream object used to read from and write to the "sales.txt" file. This object is used by the `write()` and `read_and_display()` functions to write and read data to and from the file.

- `sales_p`: An integer representing the salesperson number. This variable is used in several functions to store the user's input for the salesperson number.

- `prod`: An integer representing the product number. This variable is used in several functions to store the user's input for the product number.

- `amount`: A double representing the total value of a product sold. This variable is used in several functions to store the user's input for the total value of a product sold.

Functions:

- `write()`: This function prompts the user to enter sales data and writes it to the "sales.txt" file. The user can enter multiple records by entering the salesperson number, product number, and total value of the product sold. The function ends when the user enters 0 for the salesperson number. The function uses a while loop to repeatedly prompt the user for input until they enter 0 for the salesperson number. The input is then written to the file using the `sale` file stream object.

- `grand_sum(sales e[])`: This function takes an array of `sales` structures as an argument and returns the grand total of all sales. The function uses a for loop to iterate over each element of the array (representing each salesperson) and adds their total sales (`s_total`) to a running total (`tot`). The final total is then returned.

- `product_sum(sales e[], int num)`: This function takes an array of `sales` structures and an integer representing a product number as arguments and returns the total sales for that product. The function uses a for loop to iterate over each element of the array (representing each salesperson) and adds their sales for that product (`product[num]`) to a running total (`tot`). The final total is then returned.

- `display_t()`: This function displays a summary table of all sales data. The table shows the total sales for each product and each salesperson, as well as their bonus (calculated as 5% of their total sales). The function uses nested for loops to iterate over each salesperson and product and print their corresponding data from the `emp` array. It also calls the `grand_sum()` and `product_sum()` functions to calculate and print row and column totals.

- `read_and_display()`: This function reads sales data from the "sales.txt" file and updates the `emp` array accordingly. It then prompts the user to choose whether they want to see a summary table or search for a particular product or salesperson record. If they choose to search, they are prompted to enter either a product or salesperson number, and the corresponding data is displayed. The function starts by opening the "sales.txt" file for reading using the `sale` file stream object. It then uses a while loop to read data from the file until the end of the file is reached. For each record read, it updates the corresponding element of the `emp` array by adding the amount to the appropriate product and total sales. Once all data has been read, the file is closed and the user is prompted to choose how they want to display the data. If they choose to see a summary table, the `display_t()` function is called. If they choose to search for a particular record, they are prompted to enter either a product or salesperson number, and the corresponding data is displayed using either the `display_product()` or `display_sales_p()` function.

- `display_product(int num)`: This function takes an integer representing a product number as an argument and displays the total sales for that product by each salesperson. The function uses a for loop to iterate over each salesperson and print their total sales for that product (`product[num-1]`) from the `emp` array.

- `display_sales_p(int sales_p)`: This function takes an integer representing a salesperson number as an argument and displays their total sales for each product, as well as their total sales and bonus. The function uses a for loop to iterate over each product and print the salesperson's total sales for that product (`product[pr]`) from the `emp` array. It also calculates and prints their total sales (`s_total`) and bonus (calculated as 5% of their total sales).

#The program starts by calling the `write()` function to prompt the user to enter sales data. Once all data has been entered, it calls the `read_and_display()` function to read the data from the file, update the `emp` array, and display it according to the user's choice.