```
Script started on 2023-11-17 19:56:05+00:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUM
NS="103" LINES="97"]
\[\033[01;34m\]\w\[\033[00m\]$ pwd
/home/runner/Project-7-The-Bakery-Problem-kcp3s
\[\033[01;34m\]\w\[\033[00m\]$ ls -la
total 2516
drwxr-xr-x 1 runner runner     250 Nov 17 19:56 .
drwxrwxrwx 1 runner runner     146 Nov 17 19:40 ..
-rwxr-xr-x 1 runner runner   22064 Nov 17 03:33 a.out
-rw-r--r-- 1 runner runner      17 Oct 27 20:51 .breakpoints
drwxr-xr-x 1 runner runner      12 Jan 24  2022 .cache
drwxr-x--- 1 runner runner     446 Nov 16 16:17 .ccls-cache
-rw-r--r-- 1 runner runner    4048 Nov 17 19:31 cla19.cpp
drwxr-xr-x 1 runner runner      68 Nov 17 18:59 .lesson
-rwxr-xr-x 1 runner runner 1254392 Oct 27 20:53 main
-rw-r--r-- 1 runner runner    2669 Nov 17 18:59 main.cpp
-rwxr-xr-x 1 runner runner 1255712 Oct 27 20:53 main-debug
-rw-r--r-- 1 runner runner     449 Oct 27 20:53 Makefile
-rw-r--r-- 1 runner runner       0 Nov 17 19:56 Patel_Lab_19.log
-rw-r--r-- 1 runner runner     171 Nov 17 14:02 products.dat
-rw-r--r-- 1 runner runner    1426 Dec 21  2022 .replit
-rw-r--r-- 1 runner runner     120 Nov 17 19:40 replit.nix
\[\033[01;34m\]\w\[\033[00m\]$ cat -n cla19.cpp
     1  #include <array>
     2  #include <fstream>
     3  #include <iostream>
     4
     5  // Prototypes
     6  void open_file(std::ifstream &file);
     7  void getinfo(std::ifstream file, int &ingredients[][kMaxProducts], int &ingredien
tsprice[], const int &rows, const int &cols);
     8  void calculations(int ingredients[][kMaxProducts], double ingredientsprice[], dou
ble price[]);
     9  void display(int ingredients[][kMaxProducts], double price[], std::string product
name[], const int rows, const int cols);
    10  std::string mostexpensive();
    11
    12  // COnstant global array for product names
    13  // const int Maxsizeproduct = 7;
    14  // const std::array<std::string, Maxsizeproduct> productname = {"Donut", "Bagel",
 "White
    15  // Bread", "Kaiser Roll", "King Cake", "Apple Pie", "Cherry Wafer"}
    16
    17  int main() {
    18    std::string filename;
    19    std::ifstream file;
    20
    21    // using constants
    22    // const int kMaxProducts = 24;
    23    // const int kMaxIngredients = 30;
    24    // const int rows, cols;
    25
    26    // Assigning all the arrays;
    27     // std::array<int<int, cols>, rows> ingredients;
    28      //std::array<double, cols>  ingredientsprice;
    29      // std::array<double, cols> price;
    30
    31      // Asking for the file firsttime
    32    std::cout << "Enter a filename: ";
    33    std::cin >> filename;
    34    file.open(filename);
    35
    36      // Using the first function
    37    open_file(file);
    38
    39  //Here we would call out functions
    40
    41
    42  // Closing the file
    43    file.close();
```

```
    44  }
    45
    46  void open_file(std::ifstream &file) {
    47    //     step 1 open the file in main
    48    //      if file does not exist, in a while loop ask the user what file name he
    49    //       wants to open and check again until he gives a valid file and
    50    // if it is the right one display successfull opened.
    51
    52    // Code for the file
    53    // std::string filename;
    54    // while (!file) {
    55    //   std::cout << "Error opening file\n";
    56    //   std::cout << "Enter a filename: ";
    57    //   std::cin >> filename;
    58    //   file.open(filename);
    59    // }
    60    // if (file) {
    61    //   std::cout << "Successfully opened.";
    62    // }
    63  }
    64
    65  void getinfo(std::ifstream file, int ingredients[][kMaxProducts], double ingredie
ntsprice[], const int &rows, const int &cols) {
    66           // THis function will get input from the file
    67            // As first line assigns rows and columns it would assign the rows and
columns using reference
    68                  // THen using the rows and columns this would fill in a 2D array
for ingredients list
    69          // Then we would return the info back to the 2D array for using referrenc
e ingredients;
    70            // Another double array for ingredients price and this would take the l
ast column and return to ingredients price using referrence
    71  }
    72
    73  void calculations(int ingredients[][kMaxProducts], double ingredientsprice[], dou
ble price[]){
    74      // For this function we will multiply the 2D array for the ingredients list w
ith our double array for the ingredients price to get the final price for each product an
d return it back to the third array for double price;
    75  }
    76
    77  void display(int ingredients[][kMaxProducts], double price[], std::string product
name[], const int rows, const int cols){
    78      // This function should take 3 of the arrays and the calculations function as
 input and format it properly to get it simething like the output
    79      //Array for ingredients list, array for the global constant products name, an
d array for the price
    80      // Firstly we will do the header
    81      //Secondly we would use setw to format and set the ingredientslist and price
and the dotted line after it
    82      // Thirdly we would use the for loop to do the product name and numbers in fr
ont on them
    83          // Then we would print out the ingredients from the ingredient list array
    84          // And finally we would print out the prices and $ in front of it.
    85      // And the dotted line to end it with
    86  }
    87
    88  std::string mostexpensive(std::string productname[], double price[], int size){
    89      // This function would take const string productname array and double price a
rray and size as input and return the mostexpensive product;
    90      // Using a parallel for loop it will iterate through the price list and produ
ct name and whichever is the highest price it will return the index and name for the most
 expensive product.
    91  }
    92
    93
\[\033[01;34m\]\w\[\033[00m\]$ exit

Script done on 2023-11-17 19:56:36+00:00 [COMMAND_EXIT_CODE="0"]
```