HW 4: Functions & Arrays using the `make` tool

Refactor the code in the previous homework using the tool `make`. Again, write a program to perform a simple income tax calculation. It should prompt the user for data like income, expense, deductions, etc, and calculate the income tax for a given year, just as before. This time however, put each function in its separate file as described below. Create and configure a `Makefile` that minimizes compilation upon changes as discussed in lecture.

Do not use STL container variables (`vector`, `deque`, etc) or variables of type `string`. Rather, use integer and character arrays and value returning and/or non-returning functions which are passed value and/or reference parameters. Furthermore, its structure must comply with the following guidelines:

1. Use a header file (`hw04.h`) for assignment and your personal information as usual
2. Use a separate header file, `hw04f.h`, for declarations of functions (listed below) and include it in the file that contains your `main() (hw04.cpp)`
3. Perform **I/O** for tax payer name (`taxPayer[]`) and tax year (`year[]`) in `main()`
4. **Input** and processing (eg, calculate total, keep track of number of entries) for income items should be performed in function `getInc()`; **output** in function `prInc()`
5. Similarly, **input** and processing of expense items should be performed in function `getExp()`; **output** in function `prExp()`
6. Same scheme for dependents: perform **input** (name, relationship) in function `getDep()`; **output** in function `prDep()`
7. Finally, write a function, `prSumm()` to calculate and summarize the tax owed using the following formula:

$$tax = \{[incTot - expTot - (depTot * 3000.00)] * 0.18\} \qquad (1)$$

Use the following variables (in `main()`):

```
char taxPayer[40];            // name of tax payer
char year[5];                 // year in question (notice type char)
double incItem[10];           // gross income from each source (10 max)
char incDesc[10][40];         // gross income item description (10 max)
```

```
int isiz;                  // to hold number of income items entered
double incTot;             // total gross income for the year
double expItem[10];        // expense items (mortgage interest,medical)
char expDesc[10][40];      // expense item description
int esiz;                  // to hold number of expense items entered
double expTot;             // total expense for the year
char depNam[10][40];       // name of each dependent
char depRel[10][40];       // relationship of each dependent
int dsiz;                  // number of dependents entered if any
```

Notice that `incItem[]/incDesc[][]`, `expItem[]/expDesc[][]`, and `depNam[][]/depRel[][]` are pairs of parallel arrays. (Use them as such in your functions.) `isiz`, `esiz`, and `dsiz` are the corresponding number of entries.

Declare the following functions, filling in for the formal parameter lists, in the header file `hw04.h` and define each of them in separate function files as following:

________ `getInc` (__________ , __________ , ____________); // hw04f1.cpp

________ `getExp` (__________ , __________ , ____________); // hw04f2.cpp

________ `getDep` (__________ , __________ , ____________); // hw04f3.cpp

________ `prInc` (__________ , __________ , ____________); // hw04f4.cpp

________ `prExp` (__________ , __________ , ____________); // hw04f5.cpp

________ `prDep` (__________ , __________ , ____________); // hw04f6.cpp

________ `prSumm` (__________ , __________ , ____________); // hw04f7.cpp

Run `script` and compile/interact with the program as usual (which implies you now also have `hw04.scr`). Package a `tar` file that includes `Makefile, hw04.h, hw04f.h, hw04.cpp, hw04f1.cpp hw04f2.cpp, hw04f3.cpp, hw04f4.cpp, hw04f5.cpp, hw04f6.cpp, hw04f7.cpp`, and `hw04.scr`. Submit it to canvas by the due date on top of this page.