BITP 1113: PROGRAMMING TECHNIQUES
1BITS
Lab 7
Function Part 2

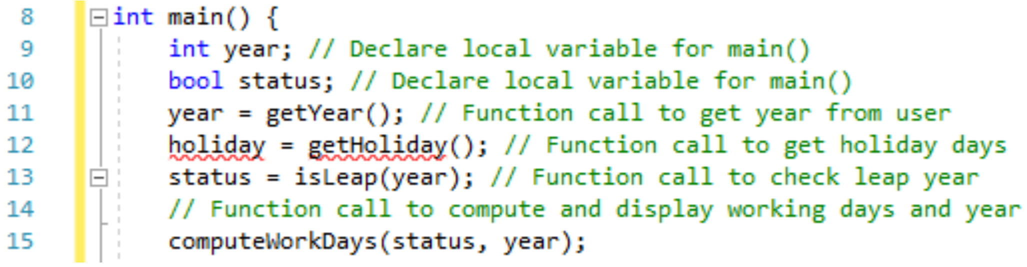
Exercise 1

A company decides to upgrade the previous chart of the number of working days in a year. Besides Saturday and Sunday, the employees also have national holidays off. Write a program to compute and display a number of working days in a year. User will provide the year and number of national holidays. Modify a program from Lab 6: Function Part 1.

1. Analyse input, process and output.

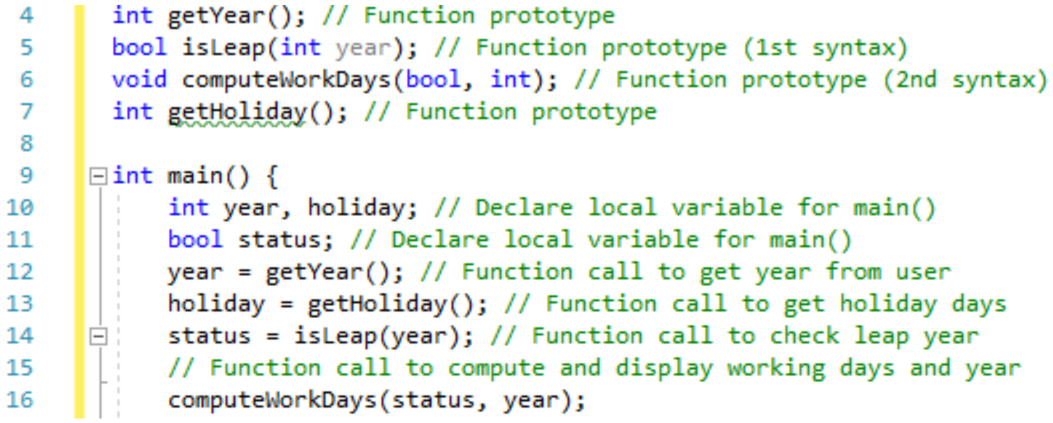
Input : `year, nationalHolidays`
Process : if leap year, `workingDays = 366 – 52 * 2 - nationalHolidays;`
`Else, workingDays = 365 – 52 * 2 - nationalHolidays;`
Output : `year, workingDays`

2. Open Lab 6: Function Part 1 project named `Days`
3. Open file `WorkingDays.cpp.` (You can copy-paste the code and modify it in a different file).
4. Write function call for `getHoliday()` to call a function to get a number of national holidays (`holiday`) from a user. This function should return the `holiday` because the working days calculation depends on this `holiday` value (Line 12).



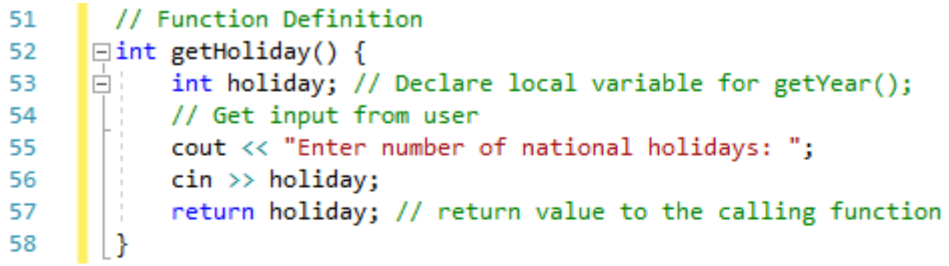
```
8   int main() {
9       int year; // Declare local variable for main()
10      bool status; // Declare local variable for main()
11      year = getYear(); // Function call to get year from user
12      holiday = getHoliday(); // Function call to get holiday days
13      status = isLeap(year); // Function call to check leap year
14      // Function call to compute and display working days and year
15      computeWorkDays(status, year);
```

5. Declare local variables `holiday` for `main()` (Line 10) and write a function prototype for `getHoliday()` (Line 7).



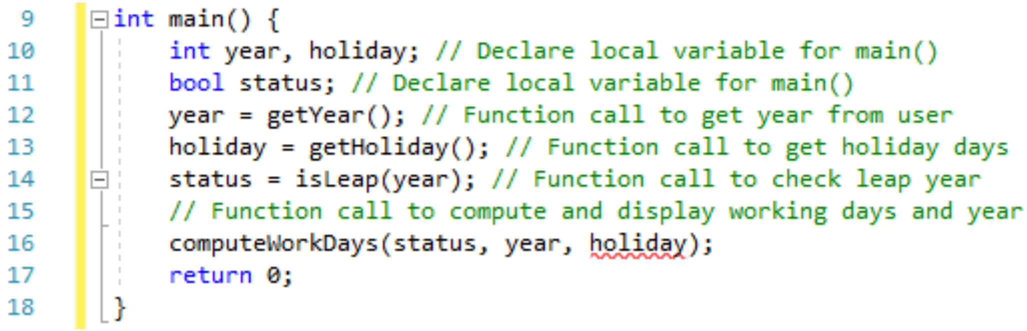
```
4   int getYear(); // Function prototype
5   bool isLeap(int year); // Function prototype (1st syntax)
6   void computeWorkDays(bool, int); // Function prototype (2nd syntax)
7   int getHoliday(); // Function prototype
8
9   int main() {
10      int year, holiday; // Declare local variable for main()
11      bool status; // Declare local variable for main()
12      year = getYear(); // Function call to get year from user
13      holiday = getHoliday(); // Function call to get holiday days
14      status = isLeap(year); // Function call to check leap year
15      // Function call to compute and display working days and year
16      computeWorkDays(status, year);
```

6. Copy-paste `getYear()` function definition and modify it to suit the `getHoliday()` function definition (Line 51 to 58).



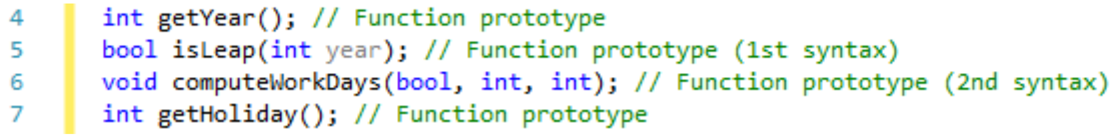
```
51      // Function Definition
52    int getHoliday() {
53          int holiday; // Declare local variable for getYear();
54          // Get input from user
55          cout << "Enter number of national holidays: ";
56          cin >> holiday;
57          return holiday; // return value to the calling function
58      }
```

7. Modify `computeWorkDays()` function call to pass `holiday,` which will be used to calculate the working days (Line 16).



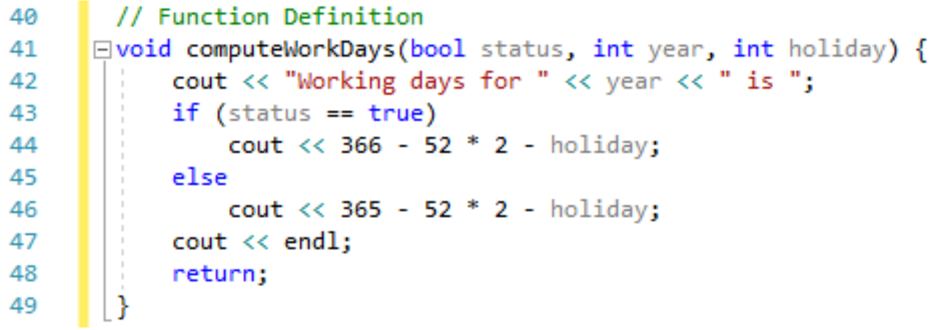
```
 9    int main() {
10          int year, holiday; // Declare local variable for main()
11          bool status; // Declare local variable for main()
12          year = getYear(); // Function call to get year from user
13          holiday = getHoliday(); // Function call to get holiday days
14          status = isLeap(year); // Function call to check leap year
15          // Function call to compute and display working days and year
16          computeWorkDays(status, year, holiday);
17          return 0;
18      }
```

8. Modify function prototype for `computeWorkDays()` to match the modified function call (Line 6).



```
4     int getYear(); // Function prototype
5     bool isLeap(int year); // Function prototype (1st syntax)
6     void computeWorkDays(bool, int, int); // Function prototype (2nd syntax)
7     int getHoliday(); // Function prototype
```

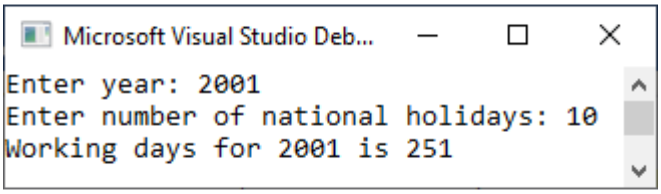
9. Modify function header for `computeWorkDays()` to match the modified function call and function prototype (Line 41). Then modify the mathematical expression used to find the working days (Line 44 and Line 46).



```
40      // Function Definition
41    void computeWorkDays(bool status, int year, int holiday) {
42          cout << "Working days for " << year << " is ";
43          if (status == true)
44              cout << 366 - 52 * 2 - holiday;
45          else
46              cout << 365 - 52 * 2 - holiday;
47          cout << endl;
48          return;
49      }
```

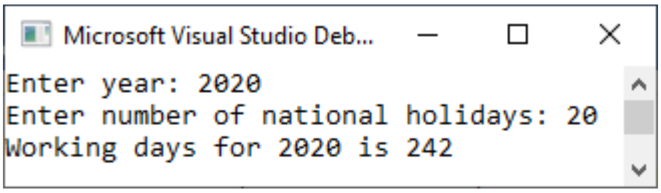
10. Compile and run

Example Output 1



```
Enter year: 2001
Enter number of national holidays: 10
Working days for 2001 is 251
```

Example Output 2

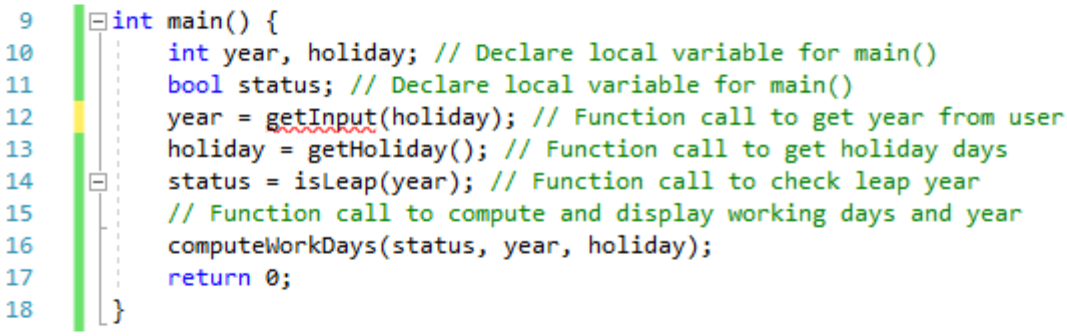


```
Enter year: 2020
Enter number of national holidays: 20
Working days for 2020 is 242
```

TIPS!

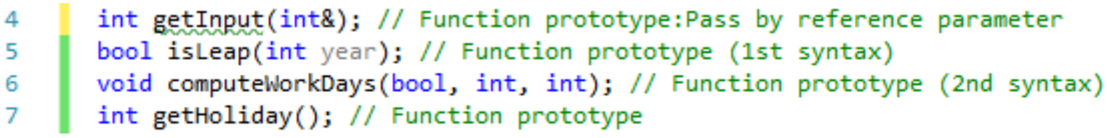
Instead of defining two functions for input (`getYear()` and `getHoliday()`), we can combine both functions tasks into one. The problem is function does not allow returning two values (a year and a number of national holidays) at the same time. So, to overcome this, we will use pass by reference.

11. At function call, rename `getYear()` to `getInput()` and modify the function call passing variable by adding `holiday` as argument (Line 12).



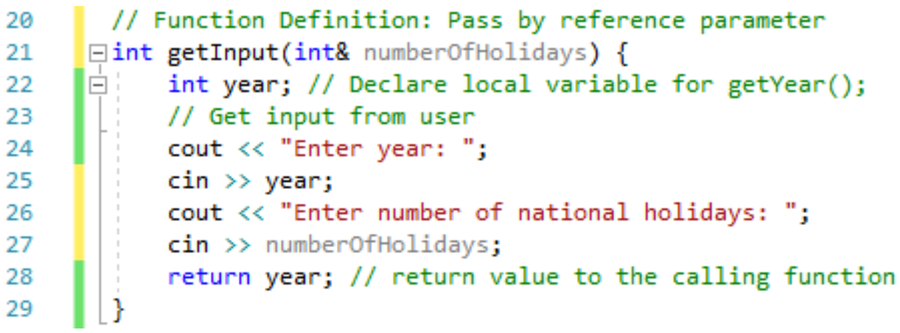
```
9   int main() {
10      int year, holiday; // Declare local variable for main()
11      bool status; // Declare local variable for main()
12      year = getInput(holiday); // Function call to get year from user
13      holiday = getHoliday(); // Function call to get holiday days
14      status = isLeap(year); // Function call to check leap year
15      // Function call to compute and display working days and year
16      computeWorkDays(status, year, holiday);
17      return 0;
18  }
```

12. At function prototype, rename `getYear()` to `getInput()` and modify the parameter as pass by reference parameter (Line 4).



```
4   int getInput(int&); // Function prototype:Pass by reference parameter
5   bool isLeap(int year); // Function prototype (1st syntax)
6   void computeWorkDays(bool, int, int); // Function prototype (2nd syntax)
7   int getHoliday(); // Function prototype
```

13. At function definition, rename `getYear()` to `getInput()` and modify the parameter at function header (Line 21). Then modify the input statements in the function body to get a year and number of national holidays from a user (Line 22 to 29).



```cpp
// Function Definition: Pass by reference parameter
int getInput(int& numberOfHolidays) {
    int year; // Declare local variable for getYear();
    // Get input from user
    cout << "Enter year: ";
    cin >> year;
    cout << "Enter number of national holidays: ";
    cin >> numberOfHolidays;
    return year; // return value to the calling function
}
```

TIPS!

In this return statement, the `getInput()` will return one value only which is `year`. Although there is no return statement for `numberOfHolidays`, the calling function will get the latest `numberOfHolidays` value because both `holiday` and `numberOfHolidays` are referring to the same reference.

14. Remove function call, function prototype and function definition of `getHoliday()` from the program.
15. Compile and run. The output will be exactly as Step 10.

Lab Attendance Week 9 (Group)

Question

In a population, the birth rate is the percentage increase due to births, and the death rate is the percentage decrease due to deaths. Write a program that asks for the following:

- The starting size of a population (minimum 2)
- The approximate annual number of birth
- The approximate annual number of death
- The number of years to display (minimum 1)

The program should then display the projected new size of population for each year. The formula to calculate the projected new size of population is

$$N = P (1 + B) (1 - D)$$

Where

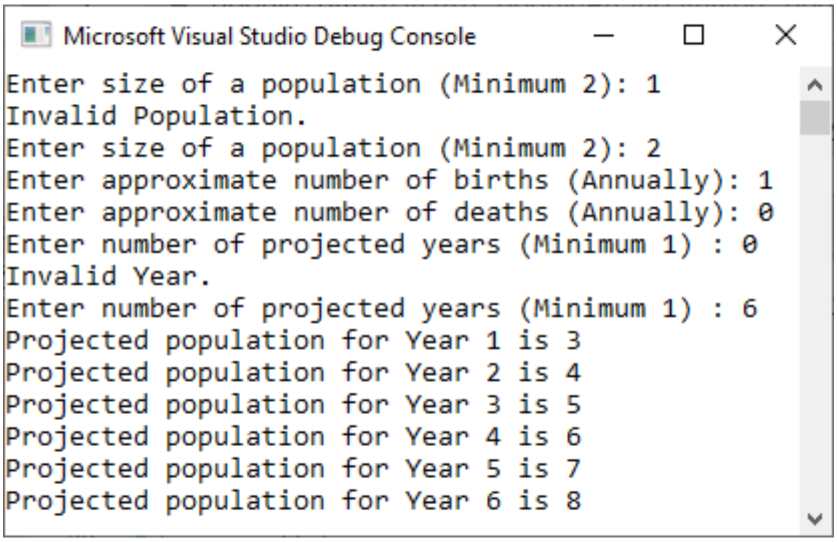
- `N` is the new population size
- `P` is the previous population size
- `B` is the birth rate
- `D` is the death rate

Annual birth rate and death rate are the typical numbers of births and deaths in a year per size of the population. For example, if there usually are about 32 births and 26 deaths per

1000 people in a given population, the birth rate would be 0.32 and the death rate would be 0.26.

Notes: Your program must apply pass by reference.

Example Output



```
Enter size of a population (Minimum 2): 1
Invalid Population.
Enter size of a population (Minimum 2): 2
Enter approximate number of births (Annually): 1
Enter approximate number of deaths (Annually): 0
Enter number of projected years (Minimum 1) : 0
Invalid Year.
Enter number of projected years (Minimum 1) : 6
Projected population for Year 1 is 3
Projected population for Year 2 is 4
Projected population for Year 3 is 5
Projected population for Year 4 is 6
Projected population for Year 5 is 7
Projected population for Year 6 is 8
```

Submit this exercise at ULearn before 12.00 p.m. 14 December 2020 (Monday).

~Sometimes later becomes never. Do it now.~
- Anonymous