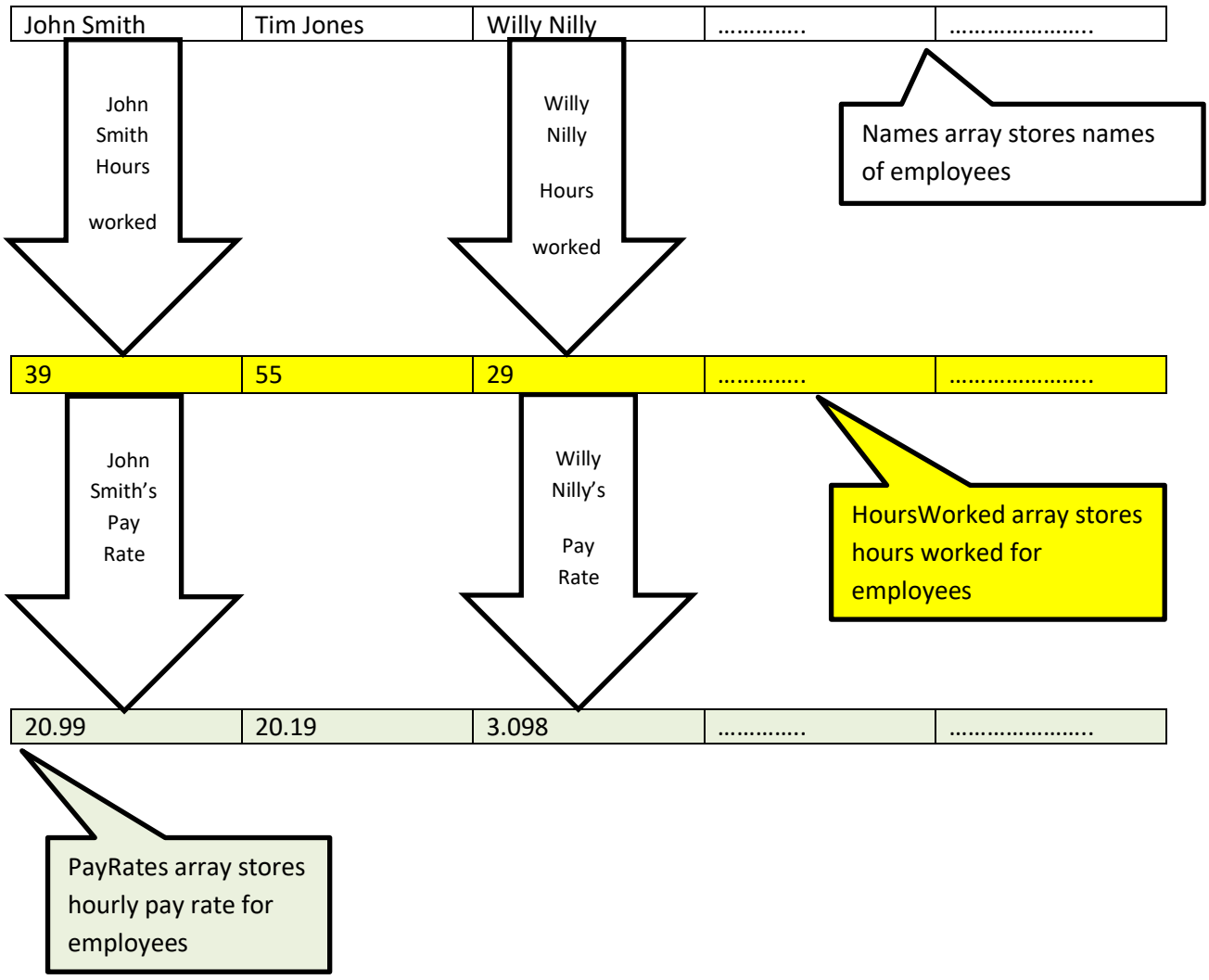
CS52 – Test 1 [Arrays and Functions]

Note: 1. Programs done without required functions will get zero. That means that even main function code gets zero credit if reasonable effort is not seen in writing two user defined functions. That also means that even if your program works with all code only in main function, your grade in this test is zero.
2. Submitted program cannot have compile errors. Your grade is zero if program has even a single compile error. Do not bother submitting programs that do not compile. Comment out lines that give you compile errors.
3. Please read the entire test paper thoroughly before attempting to code. There are no accommodations for errors resulting from information that is not read.

In this program user will be asked to enter multiple records of employee names, hours worked, and hourly pay rates. The single typical record may look like below:

John Smith 39 20.99

In above record after name are hours worked (39), and hourly pay rate in dollars (20.99). Function FillArrays (described soon), takes three arrays as arguments, one array stores the name, second stores the hours worked, and third stores the pay rate. Such arrays are called parallel arrays and schematic representation is shown in Figure 1 below:



A function called FillArrays whose prototype is given below will take three arrays as arguments. User will be prompted to enter one record at a time and function will fill all three arrays and return their logical length as a return statement from the function. Logical length will be number of employees whose records have been entered into three parallel arrays. The prototype of FillArrays is below:

size_t FillArrays (string Names [ ], int HoursWorked[ ], double PayRates[ ]);

size_t is simply a short form of C++ data type unsigned int. (Savitch describes unsigned int on page 492 of 10the edition).

main function calls the function FillArrays and supplies it the necessary actual arguments. After three arrays are filled by function FillArrays and their logical length is returned by return value, the main function does the following:

Iterates through all three arrays using a loop and array logical length returned by FillArrays function.

1. Calls the getSalary function to get the salary of each employee. Prototype of getSalary function is below:
   double getSalary(int HoursWorked, double PayRate);

   The salary computation rules inside the getSalary function are as follows:

- If worker works up to and equal to 40 hours then their salary is hours worked multiplied by the pay rate.
- If hours worked exceed 40 then overtime hours (hours beyond 40 hours), are paid at the rate of 1.5 times of the regular pay rate.

   You have used and executed above algorithm in Lab2. So, you can review that lab.

2. Then main function greets user by calling their name and prints following data for them
   - Hours Worked
   - Hourly Pay rate
   - Salary

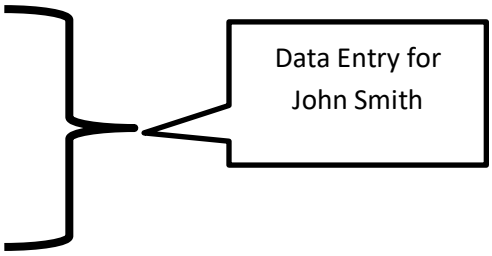
Program also prints:

1. Total number of employees processed.
2. Total payroll, which is sum of the salaries of all employees for whom computation was done.

   The output from the program that I did is shown in appendix 1. I will have a running program available if you wish to run it and see that how it works.

Appendix 1:

Hello, EOF character in windows is Control + Z
EOF character on Mac is Control + D
Enter employee's first name or EOF character to exit:
John
Enter employee's last name:
Smith
Enter hours in the week, worked by employee [whole number only]:
39
Enter employee's hourly pay rate(xx.yy):
10.23



Enter employee's first name or EOF character to exit:
Massoud
Enter employee's last name:
Ghyam
Enter hours in the week, worked by employee [whole number only]:
45
Enter employee's hourly pay rate(xx.yy):
30.45
Enter employee's first name or EOF character to exit:
Willy
Enter employee's last name:
Nilly
Enter hours in the week, worked by employee [whole number only]:
50
Enter employee's hourly pay rate(xx.yy):
25.45
Enter employee's first name or EOF character to exit:
Nilly
Enter employee's last name:
Willy
Enter hours in the week, worked by employee [whole number only]:
60
Enter employee's hourly pay rate(xx.yy):
28.45
Enter employee's first name or EOF character to exit:
Jill
Enter employee's last name:
Quirk
Enter hours in the week, worked by employee [whole number only]:
35
Enter employee's hourly pay rate(xx.yy):
20.18
Enter employee's first name or EOF character to exit:
Quirky
Enter employee's last name:
Jill
Enter hours in the week, worked by employee [whole number only]:
38
Enter employee's hourly pay rate(xx.yy):
12.22

Enter Employees first name or EOF character:
User enters Control + Z or Control + D (Depending upon operating system)

Output of Above Data entry is shown below


```
Welcome
Hello John Smith
You worked 39 hours in the week.
Your hourly pay rate: $10.23
Your weekly salary: $398.97
-----------------------------------
Hello Massoud Ghyam
You worked 45 hours in the week.
Your hourly pay rate: $30.45
Your weekly salary: $1446.38
-----------------------------------
Hello Willy Nilly
You worked 50 hours in the week.
Your hourly pay rate: $25.45
Your weekly salary: $1399.75
-----------------------------------
Hello Nilly Willy
You worked 60 hours in the week.
Your hourly pay rate: $28.45
Your weekly salary: $1991.50
-----------------------------------
Hello Jill Quirk
You worked 35 hours in the week.
Your hourly pay rate: $20.18
Your weekly salary: $706.30
-----------------------------------
Hello Quirky Jill
You worked 38 hours in the week.
Your hourly pay rate: $12.22
Your weekly salary: $464.36
-----------------------------------
Number of employees entered 6 :
Total company pay roll: $6407.26
Good bye.
```

Hints:

The loop in function FillArrays have to be such that user can enter as many records as they like, or they may exit gracefully without entering any records at all. Program when user enters no record at all would have an output like below:

```
Enter employee's first name or EOF character to exit:
User enters Control + Z or Control + D (Depending upon operating system)

Number of employees entered 0 :
Total company pay roll: $0.0
Good bye.
```

Thus, loop in FillArray function MUST be a pretest loop. Do while is NOT a pretest loop. Hence use of do while is not allowed. The loop below is not a pretest loop in the sense that it does not have a prime read before the loop, and is not allowed either:

```
while (true)

{

  //do data processing

//exit using break

}
```

As indicated earlier and is also apparent from the output example, required design would use an EOF controlled loop, which user can get out using Control + Z in windows and Control + D on Mac. I have described such a loop in description of Lab1.

Test grading rubric is such that fairly large number of points are assigned for loop in the FillArray Function.