Derick James M Espinosa

BSIT-S-T-1A-T

Unit 2

(30 pts) Create a program that will input 3 quizzes, 3 unit tests, 4 machine problems, and a term exam. The total number of items per exam is fixed at 100. For each examination, the lowest grade that the instructor could give is 20 pts.

Condition:

If the student misses three exams he will be given a grade of “INC” and a remark of “incomplete”

If the student misses 4 or more exams he will be given a grade of “DRP” and a remark of “dropped”

Otherwise, compute his grade. If his grade is less than 50 then the remark is “failed” otherwise “passed”

Computation

Total exam score = (score/100) * percent * 100

Q - 20%

U – 25%

M – 35%

T – 20%

Weighted Grade is the summation of all weighted grades of the examination.

For the Final Grade follow the following condition

Weighted Grade	Final Grade
98-100	1.0
91 -97	1.25
85-90	1.50
79-84	1.75
73-78	2.00
67-72	2.25
61-66	2.50
55-60	2.75
50-54	3.00
Below 60	5.00

Count the number of students who got passed, the number of students who got failed, the number of students who got dropped, and the number of students who got incomplete.

Display on the screen all inputted quizzes, unit tests, machine problem and term tests, each examination's weighted grade, weighted grade, and final grade. You are warned to input data as long as you want.

CODE:

```c
#include <stdio.h>
#include <windows.h>
#include <conio.h>

void main(){
    /* Variables */
    int totalExams = 11;
    int totalItems = 100;
    char *remark=0;
    char *grade=" ";
    // QUIZESS
    int quiz1=0, quiz2=0, quiz3=0;
    // UNIT TESTS
    int unit1=0, unit2=0, unit3=0;
    // MACHINE PROBLEMS
    int mp1=0, mp2=0, mp3=0, mp4=0;
    // Term Test
    int term_test = 0;

    // Counters
    int count_missing = 0;

    // Weighted
    float gquiz = 0;
    float gunit = 0;
    float gmp = 0;
    float gterm_test = 0;
    int weightedGrade = 0;

    float fgrade = 0;

ask_score:
    /* INPUTs */
    // QUIZESS
    printf("\n[ENTER QUIZ SCORES]\n");
    printf("Quiz 1: ");
```

number1.c number2.c number3.c

```c
    // QUIZESS
    printf("\n[ENTER QUIZ SCORES]\n");
    printf("Quiz 1: ");
    scanf("%d", &quiz1);
    if (quiz1==0) count_missing++;
    printf("Quiz 2: ");
    scanf("%d", &quiz2);
    if (quiz2==0) count_missing++;
    printf("Quiz 3: ");
    scanf("%d", &quiz3);
    if (quiz3==0) count_missing++;

    // UNIT TESTS
    printf("\n[ENTER QUIZ SCORES]\n");
    printf("Unit Test 1: ");
    scanf("%d", &unit1);
    printf("Unit Test 2: ");
    scanf("%d", &unit2);
    printf("Unit Test 3: ");
    scanf("%d", &unit3);

    // MACHINE PROBLEMS
    printf("\n[ENTER QUIZ SCORES]\n");
    printf("Machine Problem 1: ");
    scanf("%d", &mp1);
    printf("Machine Problem 2: ");
    scanf("%d", &mp2);
    printf("Machine Problem 3: ");
    scanf("%d", &mp3);
    printf("Machine Problem 4: ");
    scanf("%d", &mp4);

    // Term Test
    printf("\n[ENTER Tert Term score]\n");
    printf("Test Term Score: ");
    scanf("%d", &term_test);
```

```c
    printf("\n[ENTER Tert Term score]\n");
    printf("Test Term Score: ");
    scanf("%d", &term_test);


    /* PROCESS */

    if (quiz1==0) count_missing++;
    if (quiz2==0) count_missing++;
    if (quiz3==0) count_missing++;
    if (unit1==0) count_missing++;
    if (unit2==0) count_missing++;
    if (unit3==0) count_missing++;
    if (mp1==0) count_missing++;
    if (mp2==0) count_missing++;
    if (mp3==0) count_missing++;
    if (mp4==0) count_missing++;
    if (term_test==0) count_missing++;

    gquiz = ((quiz1+quiz2+quiz3)/3.0);
    gunit = ((unit1+unit2+unit3)/3.0);
    gmp= ((mp1+mp2+mp3+mp4)/4.0);
    gterm_test = term_test;

    weightedGrade = (int)(((gquiz*0.2)+(gunit*0.25)+(gmp*0.35)+(gterm_test*0.20)));
    switch(weightedGrade){
        case 98 ... 100:
            fgrade = 1.00;
            break;
        case 91 ... 97:
            fgrade = 1.25;
            break;
        case 85 ... 90:
            fgrade = 1.50;
            break;
        case 79 ... 84:
```

```c
        case 79 ... 84:
            fgrade = 1.75;
            break;
        case 73 ... 78:
            fgrade = 2.00;
            break;
        case 67 ... 72:
            fgrade = 2.25;
            break;
        case 61 ... 66:
            fgrade = 2.50;
            break;
        case 55 ... 60:
            fgrade = 2.75;
            break;
        case 50 ... 54:
            fgrade = 3.00;
            break;
        default:
            fgrade = 5.00;
            break;
    }
    if(count_missing==3){
        grade = "INC";
        remark = "Incomplete";
    }
    else if(count_missing>=4){
        grade = "DRP";
        remark = "Dropped";
    }
    else if(weightedGrade<50){
        remark = "Failed";
    }
    else {
        remark = "Passed";
    }
```

```c
    /* OUTPUT */
    printf("\n[SCORES]\n");
    printf("Quiz 1: %d\n", quiz1);
    printf("Quiz 2: %d\n", quiz2);
    printf("Quiz 3: %d\n", quiz3);
    printf("Unit Test 1: %d\n", unit1);
    printf("Unit Test 2: %d\n", unit2);
    printf("Unit Test 3: %d\n", unit3);
    printf("Machine Problem 1: %d\n", mp1);
    printf("Machine Problem 2: %d\n", mp2);
    printf("Machine Problem 3: %d\n", mp3);
    printf("Machine Problem 4: %d\n", mp4);
    printf("Term Test: %d\n", term_test);

    printf("\n[Grades]\n");
    printf("Quiz Weighted Grade: %.2f\n", gquiz);
    printf("Unit Test Weighted Grade:  %.2f\n", gunit);
    printf("Machine Problem Weighted Grade: %.2f\n", gmp);
    printf("Term Test Weighted Grade:  %.2f\n", gterm_test);

    printf("\n[Final Grade]\n");
    printf("Weighted Grade: %d\n", weightedGrade);
    printf("Final Grade: %.2f\n", fgrade);
    printf("Remarks: %s\n", remark);
    printf("Grade: %.2f\n", grade);

    ask:
        printf("\nDo you want to continue? (y/n) ");
        char ans = tolower(getch());
        if(ans=='y'){
            system("cls");
            goto ask_score;
        }
        else if(ans=='n'){
            printf("\nProgram ended.\n");
            exit(0);
        }
        else {
            goto ask;
        }
```

OUTPUT:

```
[ENTER QUIZ SCORES]
Quiz 1: 32
Quiz 2: 34
Quiz 3: 35

[UNIT TESTS SCORES]
Unit Test 1: 32
Unit Test 2: 31
Unit Test 3: 37

[ENTER MACHINE PROBLEMS SCORES]
Machine Problem 1: 37
Machine Problem 2: 30
Machine Problem 3: 32
Machine Problem 4: 33

[ENTER Tert Term score]
Test Term Score: 33

[SCORES]
Quiz 1: 32
Quiz 2: 34
Quiz 3: 35
Unit Test 1: 32
Unit Test 2: 31
Unit Test 3: 37
Machine Problem 1: 37
Machine Problem 2: 30
Machine Problem 3: 32
Machine Problem 4: 33
Term Test: 33

[Grades]
Quiz Weighted Grade: 33.67
Unit Test Weighted Grade:  33.33
Machine Problem Weighted Grade: 33.00
Term Test Weighted Grade:  33.00

[Final Grade]
Weighted Grade: 33
Final Grade: 5.00
Remarks: Failed
Grade: 0.00

Do you want to continue? (y/n)
```

```
[ENTER QUIZ SCORES]
Quiz 1: 09
Quiz 2: 09
Quiz 3: 09

[UNIT TESTS SCORES]
Unit Test 1: 09
Unit Test 2: 09
Unit Test 3: 09

[ENTER MACHINE PROBLEMS SCORES]
Machine Problem 1: 09
Machine Problem 2: 09
Machine Problem 3: 09
Machine Problem 4: 09

[ENTER Tert Term score]
Test Term Score: 09

[SCORES]
Quiz 1: 09
Quiz 2: 09
Quiz 3: 09
Unit Test 1: 09
Unit Test 2: 09
Unit Test 3: 09
Machine Problem 1: 09
Machine Problem 2: 09
Machine Problem 3: 09
Machine Problem 4: 09
Term Test: 09

[Grades]
Quiz Weighted Grade: 09.00
Unit Test Weighted Grade:  09.00
Machine Problem Weighted Grade: 09.00
Term Test Weighted Grade:  09.00

[Final Grade]
Weighted Grade: 09
Final Grade: 2.25
Remarks: Passed
Grade: 0.00

Do you want to continue? (y/n)
```

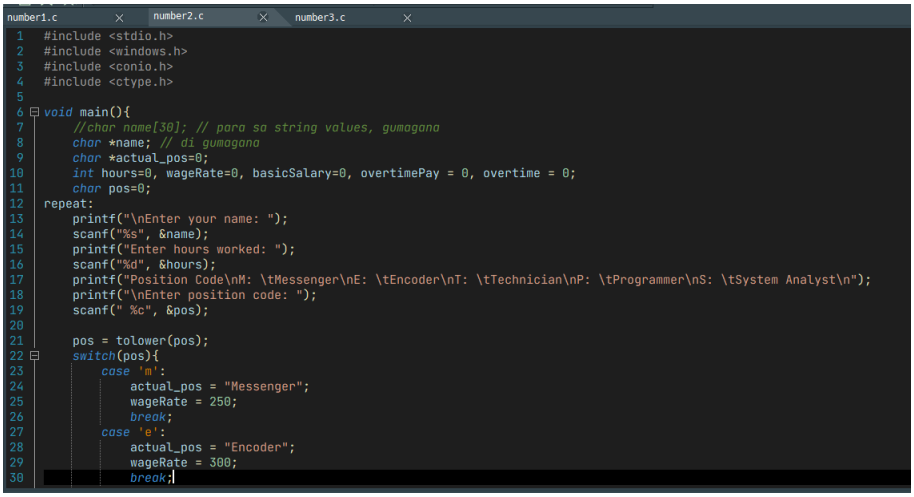
(15pts) Write a program that allows the user to compute the weekly salary of an employee based on the given condition.

Position	Description	Wage Rate
M	Messenger	250
E	Encoder	300
T	Technician	350
P	Programmer	500
S	System Analyst	600

Regular working hours for one week is 40 hours. Beyond 40 hours is paid 1.5 times the regular wage rate. Input the name, the number of hours worked, and the position of an employee and compute the weekly employee salary. Display the name, the number of hours worked, the actual employee description, weekly employee salary, and overtime if any. The user is warned to input data as long as he wants.

Basic Salary = Number hours * wage

CODE:



```c
#include <stdio.h>
#include <windows.h>
#include <conio.h>
#include <ctype.h>

void main(){
    //char name[30]; // para sa string values, gumagana
    char *name; // di gumagana
    char *actual_pos=0;
    int hours=0, wageRate=0, basicSalary=0, overtimePay = 0, overtime = 0;
    char pos=0;
repeat:
    printf("\nEnter your name: ");
    scanf("%s", &name);
    printf("Enter hours worked: ");
    scanf("%d", &hours);
    printf("Position Code\nM: \tMessenger\nE: \tEncoder\nT: \tTechnician\nP: \tProgrammer\nS: \tSystem Analyst\n");
    printf("\nEnter position code: ");
    scanf(" %c", &pos);

    pos = tolower(pos);
    switch(pos){
        case 'm':
            actual_pos = "Messenger";
            wageRate = 250;
            break;
        case 'e':
            actual_pos = "Encoder";
            wageRate = 300;
            break;
```

```c
            wageRate = 300;
            break;
        case 't':
            actual_pos = "Technician";
            wageRate = 350;
            break;
        case 'p':
            actual_pos = "Programmer";
            wageRate = 500;
            break;
        case 's':
            actual_pos = "System Analyst";
            wageRate = 600;
            break;
        default:
            printf("Error.\n");
            break;
    }

    if(hours>40){
        overtime = hours - 40;
        overtimePay = (wageRate * 0.5) * hours;
        wageRate = wageRate * 1.5;
    }
    basicSalary = hours * wageRate;

    printf("\n[OUTPUT]\n");
    printf("Name: %s\n", name);
    printf("Hours work: %d\n", hours);
    printf("Wage rate: %d\n", wageRate);
```

```c
        overtimePay = (wageRate * 0.5) * hours;
        wageRate = wageRate * 1.5;
    }
    basicSalary = hours * wageRate;

    printf("\n[OUTPUT]\n");
    printf("Name: %s\n", name);
    printf("Hours work: %d\n", hours);
    printf("Wage rate: %d\n", wageRate);
    printf("Basic Salary: %d\n", basicSalary);
    printf("\nOvertime: %d\n", overtime);
    printf("Overtime Pay: %d\n", overtimePay);

    ask:
        printf("\nDo you want to continue? (y/n) ");
        char ans = tolower(getch());
        if(ans=='y'){
            system("cls");
            goto repeat;
        }
        else if(ans=='n'){
            printf("\nProgram ended.\n");
            exit(0);
        }
        else {
            goto ask;
        }


}
```

OUTPUT:

```
Enter your name: Jack
Enter hours worked: 30
Position Code
M:      Messenger
E:      Encoder
T:      Technician
P:      Programmer
S:      System Analyst

Enter position code: M

[OUTPUT]
Name: Jack
Hours work: 30
Wage rate: 250
Basic Salary: 7500

Overtime: 0
Overtime Pay: 0

Do you want to continue? (y/n)
```

```
Do you want to continue? (y/n)
Enter your name: James
Enter hours worked: 69
Position Code
M:      Messenger
E:      Encoder
T:      Technician
P:      Programmer
S:      System Analyst

Enter position code: P

[OUTPUT]
Name: James
Hours work: 69
Wage rate: 750
Basic Salary: 51750

Overtime: 29
Overtime Pay: 17250

Do you want to continue? (y/n)
```

(15 pts.)The tolerance of critical components in a system is determined according to the

following schedule:

Specifications Status	Tolerance
Space exploration	Less than 0.1%
Military grade	Greater than or equal to 0.1% and less than 1%
Commercial grade	Greater than or equal to 1% and less than 10%
Toy grade	Greater than or equal to 10%

Using this information to create an application program that accepts a component's tolerance reading and determine the specification that should assigned to it. Accumulate tolerance of each specification status. Display specification status, tolerance and the accumulated tolerance of each specification status. Note: the user is warned to input data as long as he wants.

CODE:

```c
#include <stdio.h>
#include <windows.h>
#include <conio.h>

void main(){
    float tolerance = 0, aTolerance1=0, aTolerance2=0, aTolerance3=0, aTolerance4=0;
    char *status = 0;
repeat:
    system("cls");
    // INPUT
    printf("\nEnter the tolerance (in %%): ");
    scanf("%f", &tolerance);

    // PROCESS
    if(tolerance<0.1){
        status = "Space exploration";
        aTolerance1 += tolerance;
    }
    else if(tolerance>=0.1&&tolerance<1){
        status = "Military grade";
        aTolerance2 += tolerance;
    }
    else if(tolerance>=1&&tolerance<10){
        status = "Commercial grade";
        aTolerance3 += tolerance;
    }
    else if(tolerance>=10){
        status = "Toy grade";
        aTolerance4 += tolerance;
    }
```

number1.c number2.c number3.c

```c
    }
    else{
        system("cls");
        printf("\nInvalid tolerance value.\n");
        goto repeat;
    }

    // OUTPUT
    printf("\n[OUTPUT]\n");
    printf("Tolerance Perccent: %.2f%%\n", tolerance);
    printf("Tolerance Status: %s\n", status);
    printf("\nAccumulated Values: \n");
    printf("Less than 0.1%%: %.2f%%\n", aTolerance1);
    printf("Greater than or equal to 0.1%% and less than 1%%: %.2f%%\n", aTolerance2);
    printf("Greater than or equal to 1%% and less than 10%%: %.2f%%\n", aTolerance3);
    printf("Greater than or equal to 10%%: %.2f%%\n", aTolerance4);

    ask:
        printf("\nDo you want to continue? (y/n) ");
        char ans = tolower(getch());
        if(ans=='y'){
            goto repeat;
        }
        else if(ans=='n'){
            printf("\nProgram ended.\n");
            exit(0);
        }
        else {
            goto ask;
        }
```

OUTPUT:

```
Enter the tolerance (in %): 0.1

[OUTPUT]
Tolerance Perccent: 0.10%
Tolerance Status: Military grade

Accumulated Values:
Less than 0.1%: 0.00%
Greater than or equal to 0.1% and less than 1%: 0.10%
Greater than or equal to 1% and less than 10%: 0.00%
Greater than or equal to 10%: 0.00%

Do you want to continue? (y/n)
```

```
Enter the tolerance (in %): 0.08

[OUTPUT]
Tolerance Perccent: 0.08%
Tolerance Status: Space exploration

Accumulated Values:
Less than 0.1%: 0.08%
Greater than or equal to 0.1% and less than 1%: 0.10%
Greater than or equal to 1% and less than 10%: 0.00%
Greater than or equal to 10%: 0.00%

Do you want to continue? (y/n)
```