

To:
Tenouk

SOME EXTRA THOUGHT!

For the following three C programs and the output samples. Explain what these programs do.

```
#include <stdio.h>

int main()
{
    int count = 0;
    char letter_grade = 'x';
    int a = 0, b = 0, c = 0, d = 0, e = 0;
    int student_ID = 1;
    double test1 = 0.0, test2 = 0.0, final_test = 0.0, semester_average = 0.0,
class_average = 0.0;

    do
    {
        printf("Enter student ID, 0 to terminate: ");
        scanf("%d", &student_ID);

        if(student_ID != 0)
        {
            printf("Enter score #1: ");
            scanf("%lf", &test1);
            printf("Enter score #2: ");
            scanf("%lf", &test2);
            printf("Enter score #3: ");
            scanf("%lf", &final_test);

            printf("%d have the following scores: %lf, %lf and %lf\n", student_ID,
test1, test2, final_test);
            semester_average = (0.20*test1) + (0.30*test2) + (0.50*final_test);
            printf("The semester average score for %d is %lf\n", student_ID,
semester_average);
            if(semester_average >= 80)
                { letter_grade = 'A'; a++; }
            else if (semester_average >= 65)
                { letter_grade = 'B'; b++; }
            else if (semester_average >= 50)
                { letter_grade = 'C'; c++; }
            else if (semester_average >= 40)
                { letter_grade = 'D'; d++; }
            else if(semester_average >= 0)
                { letter_grade = 'E'; e++; }

            printf("The grade for %d is %c\n", student_ID, letter_grade);
            count++;
            printf("Student #: %d\n", count);
        }
    }while(student_ID != 0);

    printf("\nTotal number of student: %d\n", count);
    printf("\nThe number of grade A student is: %d\n", a);
    printf("\nThe number of grade B student is: %d\n", b);
    printf("\nThe number of grade C student is: %d\n", c);
    printf("\nThe number of grade D student is: %d\n", d);
    printf("\nThe number of grade E student is: %d\n", e);

    class_average = ((4*a) + (3*b) + (2*c) + (1*d))/count;

    printf("\nThe class average is: %lf\n", class_average);
    return 0;
}
```

```
Enter student ID, 0 to terminate: 112
Enter score #1: 79.50
Enter score #2: 60.70
Enter score #3: 95.5
112 have the following scores: 79.500000, 60.700000 and 95.500000
The semester average score for 112 is 81.860000
The grade for 112 is A
Student #: 1
Enter student ID, 0 to terminate: 323
Enter score #1: 90.99
Enter score #2: 45.50
Enter score #3: 78.90
323 have the following scores: 90.990000, 45.500000 and 78.900000
The semester average score for 323 is 71.298000
The grade for 323 is B
Student #: 2
Enter student ID, 0 to terminate: 114
Enter score #1: 50.23
Enter score #2: 60.30
Enter score #3: 45.21
114 have the following scores: 50.230000, 60.300000 and 45.210000
The semester average score for 114 is 50.741000
The grade for 114 is C
Student #: 3
Enter student ID, 0 to terminate: 512
Enter score #1: 91.12
Enter score #2: 85.23
Enter score #3: 81.30
512 have the following scores: 91.120000, 85.230000 and 81.300000
The semester average score for 512 is 84.443000
The grade for 512 is A
Student #: 4
Enter student ID, 0 to terminate: 812
Enter score #1: 34.20
Enter score #2: 70.23
Enter score #3: 10.16
812 have the following scores: 34.200000, 70.230000 and 10.160000
The semester average score for 812 is 32.989000
The grade for 812 is E
Student #: 5
Enter student ID, 0 to terminate: 0

Total number of student: 5

The number of grade A student is: 2
The number of grade B student is: 1
The number of grade C student is: 1
The number of grade D student is: 0
The number of grade E student is: 1

The class average is: 2.000000
Press any key to continue . . .
```

Using the do-while loop, this program keeps reading student number and their marks for three subjects until 0 is entered. Then an average mark will be calculated and be assigned appropriate grade for the student using the if-else statement. When the input is terminated, a total number for every grade obtained will be displayed and finally the average grade for the whole class will be determined.

```
#include <stdio.h>

// Function prototype
void process(int []);
void display(int [], int);

int main()
{
    // define an int array variable with 30 integer values
    int myarray[30] = {3, -4, 1, 7, 9, -10, 12, -11, -21, 15,
-5, 9, 33, -41, 31, 17, -18, 54, 2, -13,
13, -27, -31, 44, 40, 51, 62, -12, -17, -37};

    // call process() function, bringing the pointer to the
// first array element...
    process(myarray);
}
```

```

return 0;
}

// the array name is the pointer to the first element
// of the array. This function process an array. Find the positive number and
// store it in positive[] array, and negative number in negative[] array
// the sort those array in ascending
void process(int bringarray[30])
{
    // define an index
    int i, j = 0, k = 0;
    int positive[30];
    int negative[30];

    printf("The original array is:\n");
    for(i=0; i<=29;i++)
    {
        printf("%d ", bringarray[i]);
        if(bringarray[i] >= 0)
        {
            positive[j] = bringarray[i];
            // Final value of j is the limit/max though
            // given the array size is 30
            j = j + 1;
        }
        // else (negative number) put it in negative[] array
        else
        {
            negative[k] = bringarray[i];
            // Final k value is the limit/max
            k = k + 1;
        }
    }
    printf("\n");
    printf("Positive sorted array elements:\n");
    display(positive, j);
    printf("Negative sorted array elements:\n");
    display(negative, k);
    printf("\n");
}

```

```

-----
3 -4 1 7 9 -10 12 -11 -21 15 -5 9 33 -41 31 17 -18 54 2 -13 13
Positive sorted array elements:
1 2 3 7 9 9 12 13 15 17 31 33 40 44 51 54
Negative sorted array elements:
-41 -37 -31 -27 -21 -18 -17 -13 -12 -11 -10 -5
Press any key to continue . . .

```

This program declare myarray array of 30 integers. This array then, sent to process () function for processing. Firstly the function displays the array content horizontally using for loop. In the for loop also, the array was iterated to separate positive and negative integers. The positive integers were stored in positive[] array and negative integers were stored in negative[] array. Finally, the positive and negative arrays were sent to display() function for printing.

```

// function to display the array
void display(int signarray[30], int p)
{
    // i will be used for index, q is the array size,
    // pass and hold will be use as comparison
    // count and temporary storage respectively
    // in arranging the array element ascending
    int i, q = p, pass, hold;

    // do the ascending sorting...
    for(pass = 1; pass <= q-1; pass++)
    //for every 2 array elements comparison do
    //the comparison and swap...
    for(i = 0; i <= q-2; i++)
    //set the condition...
    if(signarray[i] > signarray[i + 1])
    {
        // put the a[i] in temporary variable hold...
        hold = signarray[i];
        // put the a[i + 1] in a[i]
        signarray[i] = signarray[i + 1];
        // put the hold in a[i + 1], one swapping is
        // completed and repeat for other elements...
        signarray[i + 1] = hold;
    }

    // Finally display the supposedly sorted array...
    for(i=0; i < q - 1; i++)
    printf("%d ", signarray[i]);
    printf("\n");
}

```

```

#include <stdio.h>
#define day 4
#define maxsize 10

```

```

void calculatesaltax(float);

```

```

int main()
{
    // array index
    int i;
    // total working hour and array to store user input
    float sumhour = 0.0, list[maxsize];

    // getting and storing user inputs
    for(i=0; i<= day; i++)
    {
        printf("Enter your day working hours (hours), day #%d: ", i);
        // scanf("%f", &list[i]);
        scanf_s("%f", &list[i], sizeof(list));
        // calculate the total working hours
        sumhour = sumhour + list[i];
    }
}

```

```

}

// just to re-confirm the input by displaying the input values
printf("\nConfirm your working hours per week:\n");
for(i=0;i<=day; i++)
    printf("%.2f ", list[i]);

// displaying the total working hours
printf("\nYour total working hours for this week is: %.2f\n", sumhour);

// calling calculatesaltax() function, bringing along
// the total working hour
calculatesaltax(sumhour);

return 0;
}

void calculatesaltax(float sumhour)
{
    // declaring and initializing the local variables...
    float sum1 = 0.0, sum2 = 0.0, sum3 = 0.0, hourly_rate = 300.00;

    // calculating the gazetted/basic total salary, that is < 40 working hours
    if (sumhour <= 40)
    {
        sum1 = sumhour * hourly_rate;
        printf("Total salary for gazetted hours = %f", sumhour * hourly_rate);
    }

    // calculating the total overtime salary...
    if (sumhour > 40)
    {
        // simple type conversion, may lose some precision
        sum2 = (float)((sumhour-40) * 1.5 * hourly_rate);
    }

    // total up the basic salary plus overtime...
    sum3 = sum1 + sum2;
    printf("Your total salary in this week for %.2f working hours is: USD%.2f\n",
sumhour, sum3);

    // checking and classifying the tax payment category...
    if(sum3 < 0)
        printf("You don't have to pay any tax!\n");

    if((0 <= sum3) & (sum3 <= 300))
        printf("Your tax is = USD%.2f\n", (2/100)*sum3);

    if(sum3 >= 300)
        printf("Your tax is = USD%.2f\n", (100 + ((2.5/100)*sum3)));
}

```

```

-----
Enter your day working hours (hours), day #0: 15.20
Enter your day working hours (hours), day #1: 10.24
Enter your day working hours (hours), day #2: 17.50
Enter your day working hours (hours), day #3: 13.37
Enter your day working hours (hours), day #4: 14.25

Confirm your working hours per week:
15.20 10.24 17.50 13.37 14.25
Your total working hours for this week is: 70.56
Your total salary in this week for 70.56 working hours is: USD13752.00
Your tax is = USD443.80
Press any key to continue . . .

```

This program prompts user for working hours per week. Then the total working hours per week was calculated. Based on the total working hours per week, a total income per week was calculated by calculatesaltax() function. Based on the total working hours also, a different tax need to be paid was determined based on a different tax rate also done in calculatesaltax() function.

www.tenouk.com

[| Main](#) | [Site Index](#) | [Download](#) |