QUIZ-2

1. Get three values x, y, z and write a program to print 1 if x is the middle value, 2 if y is the middle value and 3 if z is the middle value. Assume that all three variables (x, y, z) are distinct and have different values.

   CODE:

```
#include <stdio.h>

int main() {
   float x, y, z;
   printf("Enter the first value (x): ");
   scanf("%f", &x);
   printf("Enter the second value (y): ");
   scanf("%f", &y);
   printf("Enter the third value (z): ");
   scanf("%f", &z);
if ((y < x && x < z) || (z < x && x < y)) {
      printf("1\n");  // x is the middle value
   } else if ((x < y && y < z) || (z < y && y < x)) {
      printf("2\n");  // y is the middle value
   } else {
      printf("3\n");  // z is the middle value
   }

   return 0;
}
```

   OUTPUT:

```
Enter the first value (x): 5
Enter the second value (y): 8
Enter the third value (z): 6
3
```

2. A password is said to be strong if it satisfies the following criteria:

It contains at least one lowercase English character.

It contains at least one uppercase English character.

It contains at least one special character.

The special characters are: !@#$%^&*()-+

Its length is at least 8.

It contains at least one digit. **Given a string, find its strength.**

CODE:

```
#include <stdio.h>
#include <string.h>

int is_strong_password(const char *password) {
    int has_lowercase = 0, has_uppercase = 0, has_special_char = 0, has_digit = 0;
    int length_at_least_eight = (int)strlen(password) >= 8;

    for (int i = 0; password[i] != '\0'; i++) {
        if ('a' <= password[i] && password[i] <= 'z') {
            has_lowercase = 1;
        } else if ('A' <= password[i] && password[i] <= 'Z') {
            has_uppercase = 1;
        } else if (strchr("!@#$%^&*()-+", password[i]) != NULL) {
            has_special_char = 1;
        } else if ('0' <= password[i] && password[i] <= '9') {
            has_digit = 1;
        }
    }

    if (has_lowercase && has_uppercase && has_special_char && has_digit &&
length_at_least_eight) {
        return 1;  // Strong password
    } else {
        return 0;  // Weak password
    }
}

int main() {
    char password[100];

    printf("Enter the password: ");
    scanf("%s", password);
```

```
    if (is_strong_password(password)) {
        printf("The password is strong.\n");
    } else {
        printf("The password is weak.\n");
    }

    return 0;
}
```

OUTPUT:

```
Enter the password: NATCHA_03
The password is weak.
```

3. A firm creates projects for which a certain number of hours are needed. The firm has a certain number of days. During 10% of the days, the workers are being trained and cannot work on the project. A normal working day is 8 hours long. The project is important for the firm and every worker must work on it with overtime of 2 hours per day. The hours must be rounded down to the nearest integer (for example, 6.98 hours are rounded to 6 hours). Write a program that calculates whether the firm can finish the project on time and how many hours more are needed or left.

Input:

Accept three integers as input(total number of hours needed,number of days,number of workers).

Output:

If the time is enough,print "Yes!{the hours left} hours left.".
If the time is NOT enough, print "Not enough time!{additional hours} hours needed.

CODE:

```
#include <stdio.h>
#include <math.h>
int calculate_hours_needed(int hours_needed, int days, int workers) {
    double total_days = days - (0.10 * days);  // Accounting for training days
    int normal_working_hours = 8;
    int overtime_hours = 2;
 double total_hours_available = total_days * workers * (normal_working_hours
+ overtime_hours);
    int hours_left = total_hours_available - hours_needed;
 if (hours_left >= 0) {
        printf("Yes! %d hours left.\n", hours_left);
    } else {
        int additional_hours_needed = abs(hours_left);
```

```
        printf("Not enough time! %d hours needed.\n", additional_hours_needed);
    }

    return 0;
}
int main() {
  int hours_needed, days, workers;
    printf("Enter the total number of hours needed: ");
    scanf("%d", &hours_needed);

    printf("Enter the number of days: ");
    scanf("%d", &days);

    printf("Enter the number of workers: ");
    scanf("%d", &workers);
    calculate_hours_needed(hours_needed, days, workers);
    return 0;
}
```

OUTPUT:

```
Enter the total number of hours needed: 150
Enter the number of days: 20
Enter the number of workers: 5
Yes! 750 hours left.
```