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.

```
#include <stdio.h>

int main() {

    int x, y, z;

     printf("Enter the value of x: ");

    scanf("%d", &x);

    printf("Enter the value of y: ");

    scanf("%d", &y);

    printf("Enter the value of z: ");

    scanf("%d", &z);

    if ((x < y && y < z) || (z < y && y < x)) {

        printf("2\n");  // y is the middle value

    } else if ((y < x && x < z) || (z < x && x < y)) {

        printf("1\n");  // x is the middle value

    } else {

        printf("3\n");  // z is the middle value

    }

return 0;

    }
```

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.**

```
#include <stdio.h>

#include <string.h>

#include <stdbool.h>

 bool isSpecialChar(char ch) {
    char specialChars[] = "!@#$%^&*()-+";
    for (int i = 0; i < strlen(specialChars); i++) {
        if (ch == specialChars[i]) {
            return true;
        }
    }
    return false;
}
bool isStrongPassword(char password[]) {
    int length = strlen(password);
 if (length < 8) {
        return false;
    }
bool hasLower = false;
bool hasUpper = false;
 bool hasSpecial = false;
 bool hasDigit = false;
for (int i = 0; i < length; i++) {
        if (islower(password[i])) {
            hasLower = true;
        } else if (isupper(password[i])) {
            hasUpper = true;
        } else if (isdigit(password[i])) {
            hasDigit = true;
        } else if (isSpecialChar(password[i])) {
            hasSpecial = true;
        }
    }
   return hasLower && hasUpper && hasDigit && hasSpecial;
}
```

```
 int main() {

 char password[100];

 printf("Enter the password: ");

 scanf("%s", password);

 if (isStrongPassword(password)) {

printf("Password is strong.\n");

   } else {

      printf("Password is not strong.\n");

   }

 return 0;

}
```

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.

```
#include <stdio.h>
 int main() {
   int totalHoursNeeded, numDays, numWorkers;
   printf("Enter the total number of hours needed: ");
   scanf("%d", &totalHoursNeeded);
   printf("Enter the number of days: ");
   scanf("%d", &numDays);
   printf("Enter the number of workers: ");
   scanf("%d", &numWorkers);


   int totalWorkDays = numDays - (numDays * 0.1); // 10% of days are training days
   int totalWorkHours = totalWorkDays * numWorkers * 8; // 8 hours per worker per day
   int totalOvertimeHours = totalWorkDays * numWorkers * 2;
   int totalAvailableHours = totalWorkHours + totalOvertimeHours;
    if (totalAvailableHours >= totalHoursNeeded) {
    int hoursLeft = totalAvailableHours - totalHoursNeeded;
    printf("Yes! %d hours left.\n", hoursLeft);
   } else {
    int additionalHoursNeeded = totalHoursNeeded - totalAvailableHours;
    printf("Not enough time! %d hours needed.\n", additionalHoursNeeded);
   }
    return 0;
   }
```