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

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

printf("2\n"); // y 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 <ctype.h>

#include <string.h>

int is\_strong\_password(char \*password) {

int len = strlen(password);

if (len < 8) {

return 0; // Password must be at least 8 characters long

}

int has\_lower = 0, has\_upper = 0, has\_digit = 0, has\_special = 0;

for (int i = 0; i < len; i++) {

if (islower(password[i])) {

has\_lower = 1;

} else if (isupper(password[i])) {

has\_upper = 1;

} else if (isdigit(password[i])) {

has\_digit = 1;

} else if (strchr("!@#$%^&\*()-+", password[i])) {

has\_special = 1;

}

}

return has\_lower && has\_upper && has\_digit && has\_special;

}

int main() {

char password[100];

printf("Enter your password: ");

scanf("%s", password);

if (is\_strong\_password(password)) {

printf("Strong\n");

} else {

printf("Weak\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 for the project: ");

scanf("%d", &totalHoursNeeded);

printf("Enter the number of days available: ");

scanf("%d", &numDays);

printf("Enter the number of workers: ");

scanf("%d", &numWorkers);

// Calculate the total available working hours, considering training days and overtime

int effectiveWorkDays = numDays - (numDays \* 10) / 100; // Subtract 10% for training days

int dailyHoursPerWorker = 8 + 2; // 8 normal hours + 2 overtime hours

int totalAvailableHours = effectiveWorkDays \* dailyHoursPerWorker \* numWorkers;

// Calculate the difference between available hours and needed hours, rounded down

int difference = totalAvailableHours - totalHoursNeeded;

int hoursLeftOrNeeded = difference < 0 ? -difference : difference; // Use absolute value

// Print the result

if (difference >= 0) {

printf("Yes! %d hours left.\n", hoursLeftOrNeeded);

} else {

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

}

return 0;

}