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() {

int x,y,z;

printf("Enter value of x: ");

scanf("%d",&x);

printf("Enter value of y: ");

scanf("%d",&y);

printf("Enter value of z: ");

scanf("%d",&z);

//y is the middle value

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

printf("2");

//x is the middle value

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

printf("1");

//z is the middle value

else

printf("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\_lowercase(char c) {

return (c >= 'a' && c <= 'z');

}

int is\_uppercase(char c) {

return (c >= 'A' && c <= 'Z');

}

int is\_digit(char c) {

return (c >= '0' && c <= '9');

}

int is\_special\_char(char c) {

return (c == '!' || c == '@' || c == '#' || c == '$' || c == '%' || c == '^' || c == '&' || c == '\*' || c == '(' || c == ')' || c == '-' || c == '+');

}

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

int length = strlen(password);

if (length < 8) {

return 0;

}

int has\_lowercase = 0, has\_uppercase = 0, has\_digit = 0, has\_special\_char = 0;

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

if (is\_lowercase(password[i])) {

has\_lowercase = 1;

} else if (is\_uppercase(password[i])) {

has\_uppercase = 1;

} else if (is\_digit(password[i])) {

has\_digit = 1;

} else if (is\_special\_char(password[i])) {

has\_special\_char = 1;

}

}

return (has\_lowercase && has\_uppercase && has\_digit && has\_special\_char);

}

int main() {

char password[50];

printf("Enter your password: ");

scanf("%s", password);

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

printf("The password is strong.\n");

} else {

printf("The password is not strong. Please follow the specified criteria.\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.

**CODE:**

#include <stdio.h>

int main() {

int totalHoursNeeded, numberOfDays, numberOfWorkers;

printf("Enter total number of hours needed: ");

scanf("%d", &totalHoursNeeded);

printf("Enter number of days: ");

scanf("%d", &numberOfDays);

printf("Enter number of workers: ");

scanf("%d", &numberOfWorkers);

int totalWorkingDays = numberOfDays - (numberOfDays / 10);

int totalWorkingHours = totalWorkingDays \* numberOfWorkers \* 8;

int totalOvertimeHours = totalWorkingDays \* numberOfWorkers \* 2;

int totalHoursAvailable = totalWorkingHours + totalOvertimeHours;

if (totalHoursAvailable >= totalHoursNeeded) {

int hoursLeft = totalHoursAvailable - totalHoursNeeded;

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

} else {

int additionalHoursNeeded = totalHoursNeeded - totalHoursAvailable;

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

}

return 0;

}