**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 three distinct values for x, y, and z: ");

scanf("%d %d %d", &x, &y, &z);

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

{

printf("1\n");

}

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

{

printf("2\n");

}

else

{

printf("3\n");

}

return 0;

}

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 main() {

char password[100];

printf("Enter a password: ");

scanf("%s", password);

int length = strlen(password);

int hasLower = 0, hasUpper = 0, hasDigit = 0, hasSpecial = 0;

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

{

if (islower(password[i]))

{

hasLower = 1;

}

else if (isupper(password[i]))

{

hasUpper = 1;

}

else if (isdigit(password[i]))

{

hasDigit = 1;

}

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

{

hasSpecial = 1;

}

}

if (length >= 8 && hasLower && hasUpper && hasDigit && hasSpecial) {

printf("Strong password\n");

} else {

printf("Weak password\n");

printf("Reasons:\n");

if (length < 8) printf("- Length should be at least 8 characters.\n");

if (!hasLower) printf("- Missing lowercase characters.\n");

if (!hasUpper) printf("- Missing uppercase characters.\n");

if (!hasDigit) printf("- Missing digits.\n");

if (!hasSpecial) printf("- Missing special characters (!@#$%^&\*()-+).\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 total\_hours\_needed, number\_of\_days, number\_of\_workers;

printf("Enter total hours needed, number of days, and number of workers: ");

scanf("%d %d %d", &total\_hours\_needed, &number\_of\_days, &number\_of\_workers);

int training\_days = number\_of\_days / 10; // 10% of days for training

int available\_days = number\_of\_days - training\_days;

int available\_hours = available\_days \* 8 \* number\_of\_workers;

int overtime\_hours = number\_of\_days \* 2 \* number\_of\_workers;

if (available\_hours + overtime\_hours >= total\_hours\_needed) {

int remaining\_hours = available\_hours + overtime\_hours - total\_hours\_needed;

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

} else

{

int additional\_hours\_needed = total\_hours\_needed - (available\_hours + overtime\_hours);

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

}

return 0;

}