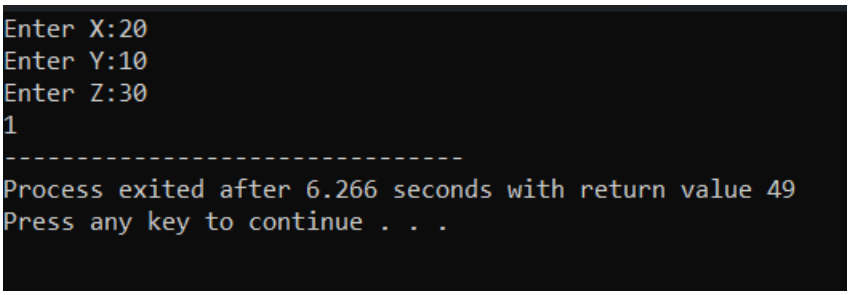
1 .Get three values x, y, z and write a program to print 1 if x is the middle value, 2 if y is the middle value and 3 if z is the middle value. Assume that all three variables (x, y, z) are distinct and have different values

Code

```
#include<stdio.h>
#include<conio.h>
void middle(int x,int y,int z)
{
        if((y<x && x<z)||(z<x && x<y))
         {
                printf("1");
         }
         else if((x<y && y<z)||(z<y && y<z))
         {
                printf("2");
         }
         else
         {
                printf("3");
         }
}
void main()
{
         int x,y,z;
         printf("Enter X:");
         scanf("%d",&x);
         printf("Enter Y:");
         scanf("%d",&y);
         printf("Enter Z:");
         scanf("%d",&z);
         middle(x,y,z);

}
```

Output



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 <stdbool.h>
#include <string.h>

void strength(char password[10]) {
   int i;
   bool haslower = false, hasupper = false, hasspecial = false;

   for (i = 0; i < strlen(password); i++) {
      if (islower(password[i])) {
         haslower = true;
      }
      if (isupper(password[i])) {
         hasupper = true;
      }
```

```
if (strchr("!@#$%^&*()-+", password[i]) != NULL) {
            hasspecial = true;
        }
    }

    if (haslower && hasupper && hasspecial && (strlen(password) > 8)) {
        printf("Strong\n");
    } else {
        printf("Weak\n");
    }
}

int main() {
    char password[10];
    printf("Enter password:\n");
    scanf("%s", password);
    strength(password);
    return 0;
}
```

Output

```
Enter password:
Shobi@)12
Strong
--------------------------------
Process exited after 7.666 seconds with return value 0
Press any key to continue . . .
```

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>
#include <math.h>
void calculate(int totalDays,int projectHours,int totalWorkers,int normalWorkingHours,int overtimeHours
,int trainingDaysPercentage)
{
int effectiveWorkingDays = totalDays - (totalDays * trainingDaysPercentage / 100);
int totalAvailableHours = effectiveWorkingDays * totalWorkers * (normalWorkingHours + overtimeHours);
int hoursLeft = totalAvailableHours - projectHours;
if (hoursLeft >= 0)
{
 printf("The project can be finished on time!\n");
 printf("Hours left: %d\n", hoursLeft);
}
else
{
 printf("The project cannot be finished on time.\n");
 printf("Additional hours needed: %d\n", abs(hoursLeft));
}
}
int main()
{
    int normalWorkingHours = 8;
    int trainingDaysPercentage = 10;
    int overtimeHours = 2;
    int totalDays, projectHours, totalWorkers;
    printf("Enter the total number of days available: ");
    scanf("%d", &totalDays);
    printf("Enter the total number of hours needed for the project: ");
    scanf("%d", &projectHours);
    printf("Enter the total number of workers: ");
    scanf("%d", &totalWorkers);
```

```
calculate(totalDays,projectHours,totalWorkers,normalWorkingHours,overtimeHours,trainingDaysPercentage);

    return 0;

}
```

```
Enter the total number of days available: 40
Enter the total number of hours needed for the project: 180
Enter the total number of workers: 5
The project can be finished on time!
Hours left: 1620

--------------------------------
Process exited after 8.61 seconds with return value 0
Press any key to continue . . .
```