21-02-2023 8 TO 9 AM

Create a structure named "Book" to store book details such as title, author, and price.

```
#include<stdio.h>
struct book
{
  char title[40];
  int pages;
  float price;
  int ratings;
};
int main()
{
  struct book book1;
  printf("Enter Book Title: ");
  gets(book1.title);
  printf("Enter Book Pages:");
  scanf("%d", &book1.pages);
  printf("Enter Book Price:");
  scanf("%f", &book1.price);
  printf("Enter Book Ratings:");
  scanf("%d", &book1.ratings);
  printf("\nBook using simple structure variable.\n");
  printf("Book Title: %s\n", book1.title);
  printf("Book Pages: %d\n", book1.pages);
  printf("Book Price: %f\n", book1.price);
  printf("Book Ratings: %i\n", book1.ratings);
}
```

OUTPUT:

```
Book using simple structure variable.
Book Title: KGF
Book Pages: 300
Book Price: 450.000000
Book Ratings: 5
```

Create a structure named "Employee" to store employee details such as name, employee ID, and salary

```
#include <stdio.h>
#include <stdlib.h>
 typedef struct{
    char name[30];
    int id;
    double salary;
} Employee;
int main()
{
  int n=2;
  Employee employees[n];
  printf("Enter %d Employee Details \n \n",n);
    for(int i=0; i<n; i++){
        printf("Employee %d:- \n",i+1);
      printf("Name: ");
        scanf("%[^\n]s",employees[i].name);
      printf("Id: ");
        scanf("%d",&employees[i].id);
      printf("Salary: ");
        scanf("%lf",&employees[i].salary);
 char ch = getchar();
      printf("\n"); }
 for(int i=0; i<n; i++){
        printf("Name \t: ");
        printf("%s \n",employees[i].name);
      printf("Id \t: ");
        printf("%d \n",employees[i].id);
      printf("Salary \t: ");
        printf("%.2lf \n",employees[i].salary);
      printf("\n"); }
  return 0;}
```

OUTPUT:

```
Name  : K.YERRISWAMY
Id        : 192225106
Salary : 20000.00
Name  : P.HEMASUNDAR RAO
Id        : 192225102
Salary : 25000.00
```

Create a structure named "Student" to store student details

```
#include <stdio.h>
struct student {
    char name[50];
    int roll;
    float marks;
} s;

int main() {
    printf("Enter information:\n");
    printf("Enter name: ");
    fgets(s.name, sizeof(s.name), stdin);

    printf("Enter roll number: ");
    scanf("%d", &s.roll);
    printf("Enter marks: ");
    scanf("%f", &s.marks);

    printf("Displaying Information:\n");
    printf("Name: ");
    printf("%s", s.name);
    printf("Roll number: %d\n", s.roll);
    printf("Marks: %.1f\n", s.marks);

    return 0;
}
```

OUTPUT

```
Name: K.YERRISWAMY
Roll number: 19
Marks: 90.0
```

9 TO 10 AM

Define a structure named "Date" to represent a date (day, month, and year)

```
#include <stdio.h>
struct Date {
    int day;
    int month;
    int year;
};
int compareDates(struct Date d1, struct Date d2) {
    if (d1.year < d2.year)
        return -1;
    else if (d1.year > d2.year)
        return 1;
    else {
        if (d1.month < d2.month)
            return -1;
        else if (d1.month > d2.month)
            return 1;
        else {
            if (d1.day < d2.day)
                return -1;
            else if (d1.day > d2.day)
                return 1;
            else
                return 0;
        } }
}
int main() {
    struct Date date1, date2;
  printf("Enter first date (DD MM YYYY): ");
    scanf("%d %d %d", &date1.day, &date1.month, &date1.year);
  printf("Enter second date (DD MM YYYY): ");
    scanf("%d %d %d", &date2.day, &date2.month, &date2.year);
  int result = compareDates(date1, date2);
  if (result < 0)
        printf("First date comes before second date.\n");
    else if (result > 0)
        printf("Second date comes before first date.\n");
    else
        printf("Both dates are the same.\n");
  return 0;
}
```

OUTPUT:

```
Enter first date (DD MM YYYY): 25-10-2023
Enter second date (DD MM YYYY): 20-10-2023
Second date comes before first date.
```

Define a structure named "Point" to represent a point in a 2D coordinate system.

```
#include <stdio.h>
#include <math.h>
struct Point {
    double x;
    double y;
};
double distance(struct Point p1, struct Point p2) {
    double dx = p2.x - p1.x;
    double dy = p2.y - p1.y;
    return sqrt(dx * dx + dy * dy);
}
int main() {
    struct Point point1, point2;
    printf("Enter coordinates of first point (x y): ");
    scanf("%lf %lf", &point1.x, &point1.y);
    printf("Enter coordinates of second point (x y): ");
    scanf("%lf %lf", &point2.x, &point2.y);
    double dist = distance(point1, point2);
    printf("Distance between the points: %.2lf\n", dist);
 return 0;
}
```

OUTPUT:

```
Enter coordinates of first point (x y): 23
25
Enter coordinates of second point (x y): 35
46
Distance between the points: 24.19
```

Write a function named "findMaxMin" that takes an array of integers

```
#include <stdio.h>
void findMaxMin(int arr[], int length, int *max, int *min) {
    if (length <= 0) {
        printf("Array length should be greater than 0.\n");
        return;
    }
    *max = arr[0]; // Initialize max to first element
    *min = arr[0]; // Initialize min to first element
    for (int i = 1; i < length; i++) {
        if (arr[i] > *max)
            *max = arr[i];
        if (arr[i] < *min)
            *min = arr[i];
    }
}
int main() {
    int length;
    printf("Enter the length of the array: ");
    scanf("%d", &length);
    if (length <= 0) {
        printf("Array length should be greater than 0.\n");
        return 1;
    }
    int arr[length];
    printf("Enter the elements of the array:\n");
    for (int i = 0; i < length; i++) {
        printf("Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
    int max, min;
    findMaxMin(arr, length, &max, &min);
    printf("Maximum value: %d\n", max);
    printf("Minimum value: %d\n", min);
    return 0;
}
```

OUTPUT:

```
Enter the length of the array: 5
Enter the elements of the array:
Element 1: 24
Element 2: 635
Element 3: 3476
Element 4: 454
Element 5: 7566
Maximum value: 7566
Minimum value: 24
```

Write a function named "calculateFactorial" that takes an integer

```
#include <stdio.h>
unsigned long long calculateFactorial(int n) {
    if (n == 0)
        return 1;
    else
        return n * calculateFactorial(n - 1);
}
int main() {
    int num;
    printf("Enter a non-negative integer: ");
    scanf("%d", &num);
    if (num < 0) {
        printf("Factorial is not defined for negative numbers.\n");
        return 1;
    }
    unsigned long long factorial = calculateFactorial(num);
    printf("Factorial of %d is: %llu\n", num, factorial);
    return 0;
}
```

OUTPUT:

```
Enter a non-negative integer: 6
Factorial of 6 is: 720
```

10 TO 11 AM

Write a function named "isPalindrome" that takes a string:

```
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include <ctype.h>
bool isPalindrome(const char *str) {
    int length = strlen(str);
    int i, j;
    i = 0;
    j = length - 1;
    while (i < j) {
        while (!isalnum(str[i]) && i < j)
            i++;
        while (!isalnum(str[j]) && i < j)
            j--;
        if (tolower(str[i]) != tolower(str[j]))
            return false;
        i++;
        j--;
    }
    return true;
}
int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    if (str[strlen(str) - 1] == '\n')
        str[strlen(str) - 1] = '\0';
    if (isPalindrome(str))
        printf("The string is a palindrome.\n");
    else
        printf("The string is not a palindrome.\n");
return 0;
}
```

OUTPUT:

```
Enter a string: 1234
The string is not a palindrome.
```

Write a function named "reverseString" that takes a string

```
#include <stdio.h>
#include <string.h>
char* reverseString(char *str) {
    int length = strlen(str);
    int start = 0;
    int end = length - 1;
    while (start < end) {
        char temp = str[start];
        str[start] = str[end];
        str[end] = temp;
        start++;
        end--;
    }
    return str;
}
int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    if (str[strlen(str) - 1] == '\n')
        str[strlen(str) - 1] = '\0';
    char *reversed = reverseString(str);
    printf("Reversed string: %s\n", reversed);
    return 0;
}
```

OUTPUT:

```
Enter a string: 12345
Reversed string: 54321
```

Write a function named "isPrime" that takes an integer

```
#include <stdio.h>
#include <stdbool.h>
bool isPrime(int num) {
    if (num <= 1)
        return false;
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0)
            return false;
    }
    return true;
}
int main() {
    printf("Prime numbers between 1 and 100:\n");
    for (int i = 1; i <= 100; i++) {
        if (isPrime(i))
            printf("%d ", i);
    }
    printf("\n");
    return 0;
}
```

OUTPUT:

```
Prime numbers between 1 and 100:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

Write a program to recursively calculate the number of ways

```
#include <stdio.h>
int countWays(int numbers[], int numSize, int targetSum) {
    if (targetSum == 0)
        return 1;
    if (targetSum < 0)
        return 0;
  int ways = 0;
    for (int i = 0; i < numSize; i++) {
        ways += countWays(numbers, numSize, targetSum - numbers[i]);
    }
  return ways;
}
int main() {
    int numbers[] = {1, 2, 3};
    int numSize = sizeof(numbers) / sizeof(numbers[0]);
    int targetSum = 4;
  int ways = countWays(numbers, numSize, targetSum);
  printf("Number of ways to reach target sum %d: %d\n", targetSum, ways);
  return 0;
}
```

OUTPUT:

```
Number of ways to reach target sum 4: 7
```