----------Programs----------

```
//1
#include<stdio.h>
void sum(int,int);
void sum(int a,int b){
   printf("the sum is %d",a+b);
}
int main(){
   int a=10,b=20;
   sum(a,b);
   return 0;
}

#include<stdio.h>
int sum(int,int);
int sum(int a,int b){
   int sum=a+b;
   return sum;
}
int main(){
   int a=10,b=20;
   sum(a,b);
   int result=sum(a,b);
   printf("the result is %d",result);
}

//2
#include<stdio.h>
```

```
void swap_numbers(int,int);
void swap_numbers(int a,int b){
   int temp;
   temp=a;
   a=b;
   b=temp;
   printf("\nThe value after swapping is :");
   printf("a-%d , b-%d\n",a,b);
}

int main(){
   int a,b;
   printf("enter 2 numbers :");
   scanf("%d%d",&a,&b);
   printf("the value before swapping is :");
   printf("a-%d , b-%d",a,b);
   swap_numbers(a,b);
   return 0;
}

#include<stdio.h>
struct pair {
   int first;
   int second;
};

struct pair swap_numbers(int a, int b) {
   struct pair result;
   int temp;
```

```
    temp = a;
    a = b;
    b = temp;
    result.first = a;
    result.second = b;
    return result;
}

int main() {
    int a, b;
    printf("Enter 2 numbers: ");
    scanf("%d%d", &a, &b);
    printf("The value before swapping is: ");
    printf("a-%d , b-%d\n", a, b);

    struct pair result = swap_numbers(a, b);
    printf("The value after swapping is: ");
    printf("a-%d , b-%d\n", result.first, result.second);

    return 0;
}

3//
#include<stdio.h>
void find_max(int,int);
void find_max(int a,int b){
    int max=0;
    if(a>max && a>b){
        printf("%d is max",a);
```

```
    }else if(b>a && b>max){
        printf("%d is max",b);
    }else{}
}

int main(){
    int a=10,b=20;
    find_max(a,b);
}

#include<stdio.h>
int find_max(int,int);
int find_max(int a,int b){
    int max=0;
    if(a>max && a>b){
        return a;
    }else if(b>a && b>max){
        return b;
    }else{}
}

int main(){
    int a=10,b=20;
    int result=find_max(a,b);
    printf("the max element is %d",result);
}

//4
#include<stdio.h>
```

```
void factorial(int n){
   int pro=1;
   for(int i=1;i<n;i++){
      pro*=i;
   }
   printf("the factorial is %d",pro);
}
int main(){
   int n;
   printf("enter a number");
   scanf("%d",&n);
   factorial(n);
   return 0;
}
```

```
#include<stdio.h>
int factorial(int);
int factorial(int n){
   int pro=1;
   for(int i=1;i<n;i++){
      pro*=i;
   }
   return pro;
}
int main(){
   int n;
   printf("enter a number");
   scanf("%d",&n);
   int result=factorial(n);
```

```
    printf("the factorial is %d",result);
    return 0;
}


//5
#include<stdio.h>
void odd_even(int);
void odd_even(int n){
    if(n%2==0){
        printf("the given number is even");
    }else{
        printf("the given number is odd");
    }
}

int main(){
    int n;
    printf("enter a number");
    scanf("%d",&n);
    odd_even(n);
}

#include<stdio.h>
#include<stdbool.h>
bool odd_even(int);
bool odd_even(int n){
    if(n%2==0){
      return false;
```

```
    }else{
      return true;
    }
}
int main(){
    int n;
    printf("enter a number");
    scanf("%d",&n);
    if(odd_even(n)){
       printf("the given number %d is odd",n);
    }else{
       printf("the given number %d is even",n);
    }
}

//6
#include<stdio.h>

void intrest(float p, float r, float t) {
    float si = (p * r * t) / 100;
    printf("The simple interest is %.2f\n", si);
}

int main() {
    float p, r, t;
    printf("Enter the principal amount: ");
    scanf("%f", &p);
    printf("Enter the rate of interest: ");
    scanf("%f", &r);
```

```
    printf("Enter the number of years: ");
    scanf("%f", &t);
    intrest(p, r, t);
    return 0;
}

#include<stdio.h>

int interest(float,float,float);
int intrest(float p, float r, float t) {
    float si = (p * r * t) / 100;
    return si;
}

int main() {
    float p, r, t;
    printf("Enter the principal amount: ");
    scanf("%f", &p);
    printf("Enter the rate of interest: ");
    scanf("%f", &r);
    printf("Enter the number of years: ");
    scanf("%f", &t);
    float result=intrest(p, r, t);
    printf("the simple intrest is %.2f",result);

}

//7
#include<stdio.h>
```

```
void reverse_number(int n) {
    int reversed = 0;
    while (n != 0) {
        reversed = reversed * 10 + n % 10;
        n = n / 10;
    }
    printf("Reversed number: %d\n", reversed);
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Original number before reversal: %d\n", num);
    reverse_number(num);
    printf("Original number after reversal attempt (no change): %d\n", num);

    return 0;
}

#include<stdio.h>
int reverse_number(int);
int reverse_number(int n) {
    int reversed = 0;
    while (n != 0) {
        reversed = reversed * 10 + n % 10;
        n = n / 10;
```

```
    }
    return reversed;
}

int main() {
    int num;
    printf("Enter a number: ");
    scanf("%d", &num);

    printf("Original number before reversal: %d\n", num);
    int result=reverse_number(num);
    printf("Original number after reversal attempt (no change): %d\n", result);

    return 0;
}

//8
#include<stdio.h>
void gcd(int a, int b) {
    int temp;
    while (b != 0) {
        temp = b;
        b = a % b;
        a = temp;
    }
    printf("The GCD is: %d\n", a);
}

int main() {
```

```
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d%d", &num1, &num2);

    printf("Original numbers: %d and %d\n", num1, num2);
    gcd(num1, num2);

    return 0;
}

#include<stdio.h>
int gcd(int,int);
int gcd(int a, int b) {
    int temp;
    while (b != 0) {
        temp = b;
        b = a % b;
        a = temp;
    }
    return a;
}

int main() {
    int num1, num2;
    printf("Enter two numbers: ");
    scanf("%d%d", &num1, &num2);

    printf("Original numbers: %d and %d\n", num1, num2);
```

```
    int result=gcd(num1, num2);
    printf("the gcd of 2 numbers are :%d",result);
    return 0;
}

//9
#include<stdio.h>
void find_sum(int n){
    int sum=0;
    while(n>0){
        int last_digit=n%10;
        sum+=last_digit;
        n=n/10;
    }
    printf("the sum of digits is %d",sum);
}
int main(){
    int n;
    printf("enter a number :");
    scanf("%d",&n);
    find_sum(n);
}

#include<stdio.h>
int find_sum(int);
int find_sum(int n){
    int sum=0;
    while(n>0){
        int last_digit=n%10;
```

```
        sum+=last_digit;
        n=n/10;
    }
    return sum;
}
int main(){
    int n;
    printf("enter a number :");
    scanf("%d",&n);
    int result=find_sum(n);
    printf("the sum of digits is %d",result);

}

//10
#include<stdio.h>
void check_prime(int num) {
    if (num <= 1) {
        printf("%d is not a prime number.\n", num);
        return;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            printf("%d is not a prime number.\n", num);
            return;
        }
    }
    printf("%d is a prime number.\n", num);
}
```

```
int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    check_prime(number);
    return 0;
}

#include<stdio.h>
#include<stdbool.h>
bool check_prime(int);
bool check_prime(int num) {
    if (num <= 1) {
        printf("%d is not a prime number.\n", num);
        return false;
    }
    for (int i = 2; i * i <= num; i++) {
        if (num % i == 0) {
            return false;
        }
    }
    return true;
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
```

```
    if(check_prime(number)){
        printf("%d is a prime number",number);
    } else{
        printf("%d is not a prime number",number);
    }
    return 0;
}

//11
Fibonacci Sequence Check
Create a function that checks whether a given number belongs to the Fibonacci sequence. Pass the number by value

#include<stdio.h>
#include<math.h>
int is_perfect_square(int n) {
    int sqrt_n = (int) sqrt(n);
    return (sqrt_n * sqrt_n == n);
}
void check_fibonacci(int num) {
    if (is_perfect_square(5 * num * num + 4) || is_perfect_square(5 * num * num - 4)) {
        printf("%d is a Fibonacci number.\n", num);
    } else {
        printf("%d is not a Fibonacci number.\n", num);
    }
}

int main() {
    int number;
```

```
    printf("Enter a number: ");
    scanf("%d", &number);
    check_fibonacci(number);
    return 0;
}

#include<stdio.h>
#include<math.h>
#include<stdbool.h>
int is_perfect_square(int);
bool check_fibonacci(int);
int is_perfect_square(int n) {
    int sqrt_n = (int) sqrt(n);
    return (sqrt_n * sqrt_n == n);
}
bool check_fibonacci(int num) {
    if (is_perfect_square(5 * num * num + 4) || is_perfect_square(5 * num * num - 4)) {
        return true;
    } else {
        return false;
    }
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    if(check_fibonacci(number)){
        printf("%d is in fibonacci series",number);
```

```
    } else{
        printf("%d is not in the fibonacci series",number);
    }
}


//13

#include<stdio.h>
void binary(int);
void binary(int n) {
    int bin = 0;
    int place = 1;

    while(n > 0) {
        int rem = n % 2;
        bin = bin + rem * place;
        place = place * 10;
        n = n / 2;
    }

    printf("The binary representation is %d\n", bin);
}

int main() {
    int dec;
    printf("Enter the decimal number: ");
    scanf("%d", &dec);
    binary(dec);
```

```
    return 0;
}

#include<stdio.h>
int binary(int);
int binary(int n) {
    int bin = 0;
    int place = 1;

    while(n > 0) {
        int rem = n % 2;
        bin = bin + rem * place;
        place = place * 10;
        n = n / 2;
    }

    return bin;
}

int main() {
    int dec;
    printf("Enter the decimal number: ");
    scanf("%d", &dec);
    int result=binary(dec);
    printf("the binary representation of %d is %d",dec,result);

}

#include<stdio.h>
```

```
int is_palindrome(int);
int is_palindrome(int num) {
    int original_num = num;
    int reversed_num = 0;
    int remainder;
    while (num != 0) {
        remainder = num % 10;
        reversed_num = reversed_num * 10 + remainder;
        num = num / 10;
    }
    if (original_num == reversed_num) {
        return 1;
    } else {
        return 0;
    }
}

int main() {
    int number;
    printf("Enter a number: ");
    scanf("%d", &number);
    if (is_palindrome(number)) {
        printf("%d is a palindrome number.\n", number);
    } else {
        printf("%d is not a palindrome number.\n", number);
    }

    return 0;
}
```

```
//14
#include<stdio.h>
int matrix_trace(int,int,int);
int matrix_trace(int a, int b, int c, int d) {
   return a + d;
}

int main() {
   int a, b, c, d;
   printf("Enter the elements of the 2x2 matrix:\n");
   printf("a: ");
   scanf("%d", &a);
   printf("b: ");
   scanf("%d", &b);
   printf("c: ");
   scanf("%d", &c);
   printf("d: ");
   scanf("%d", &d);
   int trace = matrix_trace(a, b, c, d);
   printf("The trace of the matrix is: %d\n", trace);

   return 0;
}

#include<stdio.h>
void matrix_trace(int a, int b, int c, int d) {
   int sum= a + d;
   printf("The sum of diaognal element is %d",sum);
```

```
}

int main() {
    int a, b, c, d;
    printf("Enter the elements of the 2x2 matrix:\n");
    printf("a: ");
    scanf("%d", &a);
    printf("b: ");
    scanf("%d", &b);
    printf("c: ");
    scanf("%d", &c);
    printf("d: ");
    scanf("%d", &d);
    matrix_trace(a, b, c, d);
    return 0;
}

//15
#include<stdio.h>
#include<math.h>

void quadratic_roots(float a, float b, float c) {
    float discriminant = b * b - 4 * a * c;
    if (discriminant > 0) {
        float root1 = (-b + sqrt(discriminant)) / (2 * a);
        float root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("The roots are real and distinct.\n");
        printf("Root 1 = %.2f\n", root1);
        printf("Root 2 = %.2f\n", root2);
```

```
    }
    else if (discriminant == 0) {
        float root = -b / (2 * a);
        printf("The root is real and repeated.\n");
        printf("Root = %.2f\n", root);
    }
    else {
        float real_part = -b / (2 * a);
        float imaginary_part = sqrt(-discriminant) / (2 * a);
        printf("The roots are complex.\n");
        printf("Root 1 = %.2f + %.2fi\n", real_part, imaginary_part);
        printf("Root 2 = %.2f - %.2fi\n", real_part, imaginary_part);
    }
}

int main() {
    float a, b, c;
    printf("Enter the coefficients a, b, and c of the quadratic equation (ax^2 + bx + c = 0):\n");
    printf("a: ");
    scanf("%f", &a);
    printf("b: ");
    scanf("%f", &b);
    printf("c: ");
    scanf("%f", &c);
    quadratic_roots(a, b, c);
    return 0;
}
```

----------2nd set----------

```
//1
#include<stdio.h>
#include<stdbool.h>
float convert_units(float,char);
float convert_units(float value,char type){
   float converted_value;
   if(type=='c'){
      converted_value=(value/2.54);
      printf("the value %f in inches is %lf",converted_value);
   }else if(type=='i'){
      converted_value=(value*2.4);
      printf("the value %f in centimeters is %lf",converted_value);
   }

}
int main(){
   float value;
   bool l=true;
   while(l){
      int user_input;
      printf("\nPress 1 to convert from cm-to-inches,Press 2 to convert inches-to-cm,Press 3 to exit");
      scanf("%d",&user_input);
      if(user_input==1){
         printf("enter the value in centimeters");
         scanf("%f",&value);
         convert_units(value,'c');
      }
      else if(user_input==2){
```

```
            printf("enter the value in inches");
            scanf("%f",&value);
            convert_units(value,'i');
        }else{
            printf("Exiting.....");
            l=false;
        }

    }
    return 0;
}

//2
#include<stdio.h>
struct cut_result{
    int org;
    int cut;
};
struct cut_result calculate_cuts(int m,int p){
    struct cut_result result;
    int org,cut;
    result.org=m/p;
    result.cut=(m-(p*result.org));
    return result;
}
int main(){
    int m,p;
    printf("Enter the material length");
    scanf("%d",&m);
```

```
    printf("enter the piece_length");
    scanf("%d",&p);
    struct cut_result ans=calculate_cuts(m,p);
    printf("the total piece that can be cut = %d\nthe total left over =%d",ans.org,ans.cut);
}

//3
#include<stdio.h>
float calculate_rpm(float,float);
float calculate_rpm(float belt_speed, float pulley_diameter){
    float rpm;
    rpm=belt_speed/(3.14*pulley_diameter);
    return rpm;
}
int main(){
    float p,b_s;
    printf("enter the pulley diamter :");
    scanf("%f",&p);
    printf("enter the speed of belt :");
    scanf("%f",&b_s);
    float result=calculate_rpm(b_s,p);
    printf("the rpm of the machine is %.2f",result);
}

//4
#include<stdio.h>
int calculate_production_rate(int,int);
int calculate_production_rate(int speed, int efficiency) {
    return (speed * efficiency) / 100;
```

```
}

int main() {
    int speed, efficiency;
    printf("Enter the machine speed (units per hour): ");
    scanf("%d", &speed);

    printf("Enter the efficiency (percentage): ");
    scanf("%d", &efficiency);
    int production_rate = calculate_production_rate(speed, efficiency);
    printf("The effective production rate is: %d units per hour\n", production_rate);

    return 0;
}

//5
#include<stdio.h>
int calculate_wastage(int,int);
int calculate_wastage(int total_length, int leftover_length) {
    return total_length - leftover_length;
}

int main() {
    int total_length, leftover_length;
    printf("Enter the total material length: ");
    scanf("%d", &total_length);
    printf("Enter the leftover material length: ");
    scanf("%d", &leftover_length);
    int wastage = calculate_wastage(total_length, leftover_length);
```

```
    printf("The material wastage is: %d units\n", wastage);
    return 0;
}

//6
#include<stdio.h>
float calculate_energy_cost(float,float,float);
float calculate_energy_cost(float power_rating, float hours, float cost_per_kwh) {
    return power_rating * hours * cost_per_kwh;
}

int main() {
    float power_rating, hours, cost_per_kwh;
    printf("Enter the power rating (in kW): ");
    scanf("%f", &power_rating);

    printf("Enter the operating hours: ");
    scanf("%f", &hours);

    printf("Enter the cost per kWh: ");
    scanf("%f", &cost_per_kwh);
    float energy_cost = calculate_energy_cost(power_rating, hours, cost_per_kwh);
    printf("The total energy cost is: %.2f\n", energy_cost);
    return 0;
}

//7
#include<stdio.h>
float calculate_heat(float power_usage, float efficiency) {
```

```
    return power_usage * (1 - (efficiency / 100));
}

int main() {
    float power_usage, efficiency;
    printf("Enter the power usage (in Watts): ");
    scanf("%f", &power_usage);
    printf("Enter the efficiency (in percentage): ");
    scanf("%f", &efficiency);
    float heat_generated = calculate_heat(power_usage, efficiency);
    printf("The heat generated is: %.2f Joules\n", heat_generated);
    return 0;
}

//8
#include<stdio.h>
float calculate_wear_rate(float,int);
float calculate_wear_rate(float timee, int material_type){
     float wear_rate;
    if (material_type == 1) {
        wear_rate = 1.0;
    } else if (material_type == 2) {
        wear_rate = 5.0;
    } else {
        printf("Invalid material type!\n");
        return -1.0;
    }
    float total_wear = wear_rate * timee;
    return total_wear;
```

```
}
int main(){
   int type;
   float timee;
   printf("enter the material_type :");
   scanf("%d",&type);
   printf("enter the total running time :");
   scanf("%f",&timee);
   float result=calculate_wear_rate(timee,type);
   printf("the total wear time = %f",result);
}

//9
#include<stdio.h>
int calculate_reorder_quantity(int consumption_rate, int lead_time) {
   return consumption_rate * lead_time;
}

int main() {
   int consumption_rate, lead_time;
   printf("Enter the consumption rate (units/day): ");
   scanf("%d", &consumption_rate);

   printf("Enter the lead time (days): ");
   scanf("%d", &lead_time);
   int reorder_quantity = calculate_reorder_quantity(consumption_rate, lead_time);
   printf("The reorder quantity is: %d units\n", reorder_quantity);

   return 0;
```

```
}

//10
#include<stdio.h>
float calculate_defective_rate(int defective_items, int batch_size) {
    if (batch_size == 0) {
        printf("Error: Batch size cannot be zero.\n");
        return -1;
    }
    float defective_rate = ((float)defective_items / batch_size) * 100;
    return defective_rate;
}

int main() {
    int defective_items, batch_size;
    printf("Enter the number of defective items: ");
    scanf("%d", &defective_items);
    printf("Enter the total batch size: ");
    scanf("%d", &batch_size);
    float rate = calculate_defective_rate(defective_items, batch_size);
    if (rate != -1) {
        printf("The defective rate is: %.2f%%\n", rate);
    }
    return 0;
}

//11
#include<stdio.h>
```

```
float calculate_efficiency(int output_rate, int downtime) {
    int total_time = 60;
    float efficiency = ((float)(total_time - downtime) / total_time) * 100;
    return efficiency;
}

int main() {
    int output_rate, downtime;
    printf("Enter the output rate (units per hour): ");
    scanf("%d", &output_rate);
    printf("Enter the downtime (minutes): ");
    scanf("%d", &downtime);
    float efficiency = calculate_efficiency(output_rate, downtime);
    printf("The assembly line efficiency is: %.2f%%\n", efficiency);
    return 0;
}

//12
#include<stdio.h>

float calculate_paint(float area, float coverage) {
    if (coverage == 0) {
        printf("Error: Coverage per liter cannot be zero.\n");
        return -1;
    }
    float required_paint = area / coverage;
    return required_paint;
}
```

```
int main() {
    float area, coverage;
    printf("Enter the surface area to be painted (in square meters): ");
    scanf("%f", &area);

    printf("Enter the paint coverage per liter (in square meters per liter): ");
    scanf("%f", &coverage);
    float required_paint = calculate_paint(area, coverage);
    if (required_paint != -1) {
        printf("The required paint is: %.2f liters\n", required_paint);
    }

    return 0;
}

//13
#include<stdio.h>
int calculate_maintenance_schedule(int current_usage, int interval) {
    if (interval == 0) {
        printf("Error: Maintenance interval cannot be zero.\n");
        return -1;
    }
    int remaining_hours = interval - (current_usage % interval);
    return remaining_hours;
}

int main() {
    int current_usage, interval;
    printf("Enter the current usage (in hours): ");
```

```
    scanf("%d", &current_usage);

    printf("Enter the maintenance interval (in hours): ");
    scanf("%d", &interval);
    int remaining_hours = calculate_maintenance_schedule(current_usage, interval);
    if (remaining_hours != -1) {
        printf("The remaining hours for maintenance are: %d hours\n", remaining_hours);
    }
    return 0;
}
//14
#include<stdio.h>
float calculate_cycle_time(int speed, int operations) {
    float speed_per_second = speed / 3600.0;
    float cycle_time = operations / speed_per_second;
    return cycle_time;
}

int main() {
    int speed, operations;
    printf("Enter the machine speed (units per hour): ");
    scanf("%d", &speed);

    printf("Enter the number of operations per cycle: ");
    scanf("%d", &operations);

    float cycle_time = calculate_cycle_time(speed, operations);
    printf("The optimal cycle time is: %.2f seconds\n", cycle_time);
```

```
    return 0;
}
```