---Daily work---

```
/* #1
#include<stdio.h>
float efficiency(float *,float *);
float efficiency(float *a,float *b){
    return (*a / *b)*100;

}
int main(){
    float i_p,o_p;
    printf("enter the input poweer");
    scanf("%f",&i_p);
    printf("enter the output power");
    scanf("%f",&o_p);
    float *p1=&i_p;
    float *p2=&o_p;
    float result=efficiency(p1,p2);
    printf("the efficiency is %.2f",result);

}
#2
#include<stdio.h>
float adjust_speed(int *a,float *b){
    int user_input;
    printf("enter '1' to add the speed and '2' to decrease the speed :");
    scanf("%d",&user_input);
    if(user_input==1){
        return *a+*b;
    }else if(user_input==2){
        return *a-*b;
    }else{
        return -1;
    }
}

int main(){
    int c_s;
    float adj_s;
    printf("enter the current speed :");
    scanf("%d",&c_s);
    printf("enter the adjustment speed :");
    scanf("%f",&adj_s);
    int *p1=&c_s;
    float *p2=&adj_s;
    float result=adjust_speed(p1,p2);
    printf("the current speed is %d\n",c_s);
    printf("the speed after adjusting is %.2f",result);
}
```

#3

```
#include<stdio.h>
int update_changes(int*a,int*b,int s){
    for(int i=0;i<s;i++){
        a[i]+=b[i];
        if(a[i]<0){
            a[i]=0;
        }
    }
    return 0;
}
int main(){
    int n;
    printf("Enter the size of array :");
    scanf("%d",&n);
    int arr[n];
    int changes[n];
    for(int i=0;i<n;i++){
        printf("Enter the element %d of array :",i+1);
        scanf("%d",&arr[i]);
    }
    for(int i=0;i<n;i++){
        printf("\nenter the change for element %d",i+1);
        scanf("%d",&changes[i]);
    }
    update_changes(arr,changes,n);
    printf("after updating :");
    for(int i=0;i<n;i++){
        printf("%d",arr[i]);
    }
}
```

#4

```
#include<stdio.h>
void update_coordinates(int *x, int *y, int *z, int dx, int dy, int dz, int xmin, int xmax, int ymin, int ymax, int zmin, int zmax) {
    *x += dx;
    *y += dy;
    *z += dz;
    if (*x < xmin) *x = xmin;
    if (*x > xmax) *x = xmax;

    if (*y < ymin) *y = ymin;
    if (*y > ymax) *y = ymax;

    if (*z < zmin) *z = zmin;
    if (*z > zmax) *z = zmax;
}

int main() {
    int x, y, z;
    int dx, dy, dz;
    int xmin = 0, xmax = 100;
    int ymin = 0, ymax = 100;
    int zmin = 0, zmax = 100;
    printf("Enter the current x-coordinate: ");
    scanf("%d", &x);

    printf("Enter the current y-coordinate: ");
    scanf("%d", &y);

    printf("Enter the current z-coordinate: ");
    scanf("%d", &z);
    printf("Enter the delta for x: ");
    scanf("%d", &dx);

    printf("Enter the delta for y: ");
    scanf("%d", &dy);

    printf("Enter the delta for z: ");
    scanf("%d", &dz);
    update_coordinates(&x, &y, &z, dx, dy, dz, xmin, xmax, ymin, ymax, zmin, zmax);

    printf("Updated coordinates: x = %d, y = %d, z = %d\n", x, y, z);

    return 0;
}
```

#5

```
#include<stdio.h>

void adjust_temperature(float *current_temp, float min_temp, float max_temp) {
    if (*current_temp < min_temp) {
        *current_temp = min_temp;
    }
    else if (*current_temp > max_temp) {
        *current_temp = max_temp;
    }
}

int main() {
    float current_temperature;
    float min_temperature, max_temperature;
    printf("Enter the current temperature: ");
    scanf("%f", &current_temperature);
    printf("Enter the minimum allowed temperature: ");
    scanf("%f", &min_temperature);

    printf("Enter the maximum allowed temperature: ");
    scanf("%f", &max_temperature);
    adjust_temperature(&current_temperature, min_temperature, max_temperature);
    printf("Adjusted temperature: %.2f\n", current_temperature);

    return 0;
}
```

#6

```
#include<stdio.h>
void remaining_life(int*a,int*b){
    int r_m=*b-*a;
    if(r_m<0){
        r_m=0;
    }
    *a=r_m;
}
int main(){
    int c_t_u,m_l_s;
    printf("Enter the current tool usage in hours :");
    scanf("%d",&c_t_u);
    printf("enter the maximum life span :");
    scanf("%d",&m_l_s);
    int *p1=&c_t_u;
    int *p2=&m_l_s;
    remaining_life(p1,p2);
    printf("the remaining life = %d",c_t_u);
}
```

#7

```
#include <stdio.h>
float calculate_total_weight(float *weights, int n) {
   float total_weight = 0.0;
   for (int i = 0; i < n; i++) {

       if (weights[i] < 0) {
          printf("Error: Negative weight input at index %d is not allowed.\n", i);
          return -1.0;
       }
       total_weight += weights[i];
   }
   return total_weight;

int main() {
   int n;
   printf("Enter the number of materials: ");
   scanf("%d", &n);
   float weights[n];
   for (int i = 0; i < n; i++) {
       printf("Enter the weight of material %d: ", i + 1);
       scanf("%f", &weights[i]);
       if (weights[i] < 0) {
          printf("Negative weights are not allowed. Please enter a valid weight.\n");
          i--;
       }
   }
```

```
    float total = calculate_total_weight(weights, n);
    if (total != -1.0) {
        printf("The total weight of the materials is: %.2f\n", total);
    }

    return 0;
}
```

#8

```
#include <stdio.h>

// Function to adjust welding machine configuration
void configure_machine(float *voltage, float *current, float min_voltage, float max_voltage, float min_current, float max_current) {
    // Check if voltage is within the allowed range
    if (*voltage < min_voltage) {
        *voltage = min_voltage;  // Set to minimum voltage if below range
    }
    if (*voltage > max_voltage) {
        *voltage = max_voltage;  // Set to maximum voltage if above range
    }

    // Check if current is within the allowed range
    if (*current < min_current) {
        *current = min_current;  // Set to minimum current if below range
    }
    if (*current > max_current) {
        *current = max_current;  // Set to maximum current if above range
    }
}

int main() {
    float voltage, current;
    float min_voltage = 10.0, max_voltage = 100.0; // Example voltage range
    float min_current = 5.0, max_current = 50.0;   // Example current range

    // Input voltage and current
    printf("Enter the voltage: ");
    scanf("%f", &voltage);

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

    // Pointers to voltage and current
    float *p_voltage = &voltage;
    float *p_current = &current;

    // Call the function to configure the machine
    configure_machine(p_voltage, p_current, min_voltage, max_voltage, min_current, max_current);

    // Output the updated machine configuration
    printf("Updated configuration:\n");
    printf("Voltage: %.2f V\n", voltage);
    printf("Current: %.2f A\n", current);

    return 0;
}
```

#9

```
#include <stdio.h>

void calculate_defect_rate(int *total, int *defective, float *defect_rate) {
    if (*total <= *defective) {
        printf("Error: Total products must be greater than defective products.\n");
        *defect_rate = -1;
    } else {
        *defect_rate = ((float)*defective / *total) * 100;
    }
}

int main() {
    int total_products, defective_products;
    float defect_rate;
    printf("Enter total products: ");
    scanf("%d", &total_products);

    printf("Enter defective products: ");
    scanf("%d", &defective_products);
    int *p_total = &total_products;
    int *p_defective = &defective_products;
    float *p_defect_rate = &defect_rate;
    calculate_defect_rate(p_total, p_defective, p_defect_rate);
    if (defect_rate != -1) {
        printf("Defect rate: %.2f%%\n", defect_rate);
    }

    return 0;
```

```
}

#10
#include <stdio.h>
void adjust_timing_intervals(float *intervals, int size) {
    for (int i = 0; i < size; i++) {
        if (intervals[i] <= 0) {
            printf("Warning: Interval at index %d is non-positive, adjusting to 1.0.\n", i);
            intervals[i] = 1.0;
        }
    }
}

int main() {
    int n;
    printf("Enter the number of timing intervals: ");
    scanf("%d", &n);
    float intervals[n];
    for (int i = 0; i < n; i++) {
        printf("Enter the timing interval for station %d: ", i + 1);
        scanf("%f", &intervals[i]);
    }
    float *p_intervals = intervals;
    adjust_timing_intervals(p_intervals, n);
    printf("Adjusted timing intervals:\n");
    for (int i = 0; i < n; i++) {
        printf("Station %d: %.2f\n", i + 1, intervals[i]);
    }
    return 0;
}
```

#11

```
#include <stdio.h>
void update_coordinates(float *x, float *y, float *z, float delta_x, float delta_y, float delta_z, float max_x, float max_y, float max_z, float min_x, float min_y, float min_z) {
    *x += delta_x;
    *y += delta_y;
    *z += delta_z;
    if (*x < min_x) {
        *x = min_x;
    }
    if (*x > max_x) {
        *x = max_x;
    }

    if (*y < min_y) {
        *y = min_y;
    }
    if (*y > max_y) {
        *y = max_y;
    }

    if (*z < min_z) {
        *z = min_z;
    }
    if (*z > max_z) {
        *z = max_z;
    }
}

int main() {
    float x, y, z;
    float delta_x, delta_y, delta_z;
    float max_x = 100.0, max_y = 50.0, max_z = 200.0;
    float min_x = 0.0, min_y = 0.0, min_z = 0.0;
    printf("Enter current x coordinate: ");
    scanf("%f", &x);

    printf("Enter current y coordinate: ");
    scanf("%f", &y);

    printf("Enter current z coordinate: ");
    scanf("%f", &z);
    printf("Enter delta for x coordinate: ");
    scanf("%f", &delta_x);

    printf("Enter delta for y coordinate: ");
    scanf("%f", &delta_y);

    printf("Enter delta for z coordinate: ");
    scanf("%f", &delta_z);
    float *p_x = &x;
    float *p_y = &y;
    float *p_z = &z;
    update_coordinates(p_x, p_y, p_z, delta_x, delta_y, delta_z, max_x, max_y, max_z, min_x, min_y,
min_z);
```

```
    printf("Updated coordinates:\n");
    printf("x: %.2f\n", x);
    printf("y: %.2f\n", y);
    printf("z: %.2f\n", z);

    return 0;
}
```

#12

```
#include<stdio.h>
float calculate_energy(float *array,int n){
    float total_enery=0.0;
    for(int i=0;i<n;i++){
        if(array[i]<0){
            array[i]=0;
        }
        total_enery+=array[i];
    }
    return total_enery;
}

int main(){
    int n;
    printf("enter the size of array :");
    scanf("%d",&n);
    float arr[n];
    for(int i=0;i<n;i++){
        printf("Enter the element at index %d :",i+1);
        scanf("%f",&arr[i]);
    }
    float result=calculate_energy(arr,n);
    printf("the total enery consumed by systems = %f",result);
    return 0;

}

//13
#include<stdio.h>
void update_production_rate(int *rate, int adjustment, int min_rate, int max_rate) {
    *rate += adjustment;
    if (*rate < min_rate) {
        *rate = min_rate;
    }
    if (*rate > max_rate) {
        *rate = max_rate;
    }
}

int main() {
    int current_rate, adjustment_factor;
    int min_rate = 0, max_rate = 1000;
    printf("Enter the current production rate: ");
    scanf("%d", &current_rate);

    printf("Enter the adjustment factor: ");
    scanf("%d", &adjustment_factor);
    int *p_rate = &current_rate;
    update_production_rate(p_rate, adjustment_factor, min_rate, max_rate);
    printf("The updated production rate is: %d\n", current_rate);

    return 0;
}

//14
#include <stdio.h>
#include <string.h>

void update_maintenance_schedule(char *current_date, char *next_date) {
    if (strcmp(next_date, current_date) <= 0) {
        printf("Error: The next maintenance date must be later than the current date.\n");
        printf("Please enter a valid next maintenance date (YYYY-MM-DD): ");
        scanf("%s", next_date);
    }
}

int main() {
    char current_date[11], next_date[11];
    printf("Enter the current maintenance date (YYYY-MM-DD): ");
    scanf("%s", current_date);

    printf("Enter the next maintenance date (YYYY-MM-DD): ");
    scanf("%s", next_date);
    update_maintenance_schedule(current_date, next_date);
    printf("Updated maintenance schedule:\n");
    printf("Current Maintenance Date: %s\n", current_date);
    printf("Next Maintenance Date: %s\n", next_date);
    return 0;
}

//15
#include <stdio.h>

void update_quality_metrics(int *scores, int n, float *average, int *max_score, int *min_score) {
    int total_score = 0;
    *max_score = 0;
    *min_score = 100;
    for (int i = 0; i < n; i++) {
        if (scores[i] < 0) {
            scores[i] = 0;
        } else if (scores[i] > 100) {
            scores[i] = 100;
        }
        total_score += scores[i];
        if (scores[i] > *max_score) {
            *max_score = scores[i];
        }
        if (scores[i] < *min_score) {
            *min_score = scores[i];
        }
    }
    *average = (float)total_score / n;
}

int main() {
    int n;
    printf("Enter the number of products in the batch: ");
    scanf("%d", &n);
    int scores[n];
    for (int i = 0; i < n; i++) {
        printf("Enter the quality score for product %d (0-100): ", i + 1);
        scanf("%d", &scores[i]);
    }

    float average;
    int max_score, min_score;
    update_quality_metrics(scores, n, &average, &max_score, &min_score);
    printf("\nUpdated Quality Metrics:\n");
    printf("Average Quality Score: %.2f\n", average);
    printf("Maximum Quality Score: %d\n", max_score);
    printf("Minimum Quality Score: %d\n", min_score);

    return 0;
}

//16
#include <stdio.h>

void adjust_space_allocation(int *space_used, int n, int warehouse_capacity) {
    int total_space_used = 0;
    for (int i = 0; i < n; i++) {
        total_space_used += space_used[i];
    }
    if (total_space_used > warehouse_capacity) {
        int excess_space = total_space_used - warehouse_capacity;
        printf("Total space exceeds capacity by %d. Adjusting allocations...\n", excess_space);
        for (int i = 0; i < n; i++) {

            float percentage_used = (float)space_used[i] / total_space_used;
            space_used[i] -= (int)(excess_space * percentage_used);
        }
    }
}

int main() {
    int n, warehouse_capacity;
    printf("Enter the number of sections in the warehouse: ");
    scanf("%d", &n);

    int space_used[n];
    for (int i = 0; i < n; i++) {
        printf("Enter the space used for section %d: ", i + 1);
        scanf("%d", &space_used[i]);
    }
    printf("Enter the warehouse capacity: ");
    scanf("%d", &warehouse_capacity);
    adjust_space_allocation(space_used, n, warehouse_capacity);
    printf("\nUpdated Space Allocation:\n");
    for (int i = 0; i < n; i++) {
        printf("Section %d: %d units\n", i + 1, space_used[i]);
    }

    return 0;
}
```

```
//17
#include <stdio.h>

void update_machine_settings(float *speed, float *tension, float speed_min, float speed_max, float tension_min, float tension_max) {
    if (*speed < speed_min) {
        *speed = speed_min;
        printf("Speed too low. Adjusting to minimum: %.2f\n", *speed);
    } else if (*speed > speed_max) {
        *speed = speed_max;
        printf("Speed too high. Adjusting to maximum: %.2f\n", *speed);
    }
    if (*tension < tension_min) {
        *tension = tension_min;
        printf("Wrap tension too low. Adjusting to minimum: %.2f\n", *tension);
    } else if (*tension > tension_max) {
        *tension = tension_max;
        printf("Wrap tension too high. Adjusting to maximum: %.2f\n", *tension);
    }
}

int main() {
    float speed, wrap_tension;

    // Define the safe operating limits
    float speed_min = 1.0, speed_max = 100.0;     // Speed range
    float tension_min = 0.5, tension_max = 10.0;   // Wrap tension range
    // Input the current settings
    printf("Enter the current speed of the machine: ");
    scanf("%f", &speed);

    printf("Enter the current wrap tension of the machine: ");
    scanf("%f", &wrap_tension);

    // Call the function to adjust settings based on safe limits
    update_machine_settings(&speed, &wrap_tension, speed_min, speed_max, tension_min,
tension_max);

    // Output the updated settings
    printf("\nUpdated Machine Settings:\n");
    printf("Speed: %.2f\n", speed);
    printf("Wrap Tension: %.2f\n", wrap_tension);

    return 0;
}

//18
#include <stdio.h>

void adjust_temperature(float *temperature, float min_temp, float max_temp) {
    if (*temperature < min_temp) {
        *temperature = min_temp;
        printf("Temperature too low. Adjusting to minimum: %.2f\n", *temperature);
    } else if (*temperature > max_temp) {
        *temperature = max_temp;
        printf("Temperature too high. Adjusting to maximum: %.2f\n", *temperature);
    }
}

int main() {
    float current_temp;
    float min_temp = 50.0, max_temp = 150.0;
    printf("Enter the current temperature: ");
    scanf("%f", &current_temp);
    adjust_temperature(&current_temp, min_temp, max_temp);
    printf("\nAdjusted Temperature: %.2f\n", current_temp);
    return 0;
}

//19
#include <stdio.h>

void update_scrap_count(int *scrap_count, int n) {
    for (int i = 0; i < n; i++) {
        if (*(scrap_count + i) < 0) {
            *(scrap_count + i) = 0;
            printf("Scrap count for material %d was negative. Adjusting to 0.\n", i + 1);
        }
    }
}

int main() {
    int n;
    printf("Enter the number of materials: ");
    scanf("%d", &n);
    int scrap_counts[n];
    for (int i = 0; i < n; i++) {
        printf("Enter scrap count for material %d: ", i + 1);
        scanf("%d", &scrap_counts[i]);
    }
    update_scrap_count(scrap_counts, n);
    printf("\nUpdated Scrap Counts:\n");
    for (int i = 0; i < n; i++) {
        printf("Material %d: %d\n", i + 1, scrap_counts[i]);
    }

    return 0;
}

//20
#include <stdio.h>

void analyze_shift_performance(int *production_data, int n, int *total_production, float
*average_production) {
    *total_production = 0;
    for (int i = 0; i < n; i++) {
        if (*(production_data + i) < 0) {
            printf("Invalid input: Production data for shift %d is negative. Adjusting to 0.\n", i + 1);
            *(production_data + i) = 0;

        *total_production += *(production_data + i);
    }
    if (n > 0) {
        *average_production = (float)(*total_production) / n;
    } else {
        *average_production = 0.0;
    }
}

int main() {
    int n;
    printf("Enter the number of shifts: ");
    scanf("%d", &n);

    int production_data[n];
    int total_production;
    float average_production;
    for (int i = 0; i < n; i++) {
        printf("Enter production for shift %d: ", i + 1);
        scanf("%d", &production_data[i]);
    }
    analyze_shift_performance(production_data, n, &total_production, &average_production);
    printf("\nPerformance Metrics:\n");
    printf("Total production: %d\n", total_production);
    printf("Average production per shift: %.2f\n", average_production);

    return 0;
}
```