

Q1

```
#include<stdio.h>
#include<stdbool.h>
void update_inventory(int *arr,int
    size){ int user_input;
    bool
    is_on=true;
    while(is_on){
        printf("Press '1' if you want to update and element or '2' to exit :");
        scanf("%d",&user_input);
        if(user_input==1){
            int value,new_value;
            printf("Enter the element to update :");
            scanf("%d",&value);
            printf("Enter the new value :");
            scanf("%d",&new_value);
            arr[value-1]=new_value;
        }else
            if(user_input==2){
                is_on=false;
            }else{
                printf("Enter a valid input :");
            }
        }
    }
```

```
void print_array(int arr[],int
    size){ for(int i=0;i<size;i++){
        printf("The %d element of array is %d\n",i+1,arr[i]);
    }
```

```
}
```

```
int
```

```
main(){
```

```
int n;
```

```
printf("Enter the size of array :");
```

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

```
int inventory_levels[n];
```

```
for(int i=0;i<n;i++){
```

```
    printf("Enter the element %d of array :",i+1);
```

```
    scanf("%d",&inventory_levels[i]);
```

```
}
```

```
int
```

```
size=sizeof(inventory_levels)/sizeof(inventory_levels[0
```

```
]); update_inventory(inventory_levels,size);
```

```
print_array(inventory_levels,size);
```

```
}
```

```
2
```

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

```
void adjustPrices(const int demand[], const float prices[], float adjustedPrices[], int
```

```
size) { for (int i = 0; i < size; i++) {
```

```
    if (demand[i] > 50) {
```

```
        adjustedPrices[i] = prices[i] * 1.10;
```

```
    } else {
```

```
        adjustedPrices[i] = prices[i] * 0.90;
```

```
    }
```

```
}
```

```
}
```

```

int main() {
    int demandLevels[] = {40, 60, 80, 30, 55};
    float productPrices[] = {100.0, 150.0, 200.0, 120.0,
    180.0}; int numProducts = 5;
    float adjustedPrices[numProducts];
    printf("Original Prices:\n");
    for (int i = 0; i < numProducts; i++) {
        printf("Product %d: %.2f\n", i+1, productPrices[i]);
    }
    adjustPrices(demandLevels, productPrices, adjustedPrices, numProducts);
    printf("\nAdjusted Prices Based on Demand:\n");
    for (int i = 0; i < numProducts; i++) {
        printf("Product %d: %.2f\n", i+1, adjustedPrices[i]);
    }

    return 0;
}

```

3

```

#include<stdio.h>

int total_sales(int *arr,int
s){ int sum=0;
for(int
i=0;i<s;i++){
sum+=*(arr+i);
}
return sum;
}

int
main(){
int n;

```

```

printf("Enter the size of array :");
scanf("%d",&n);
int
daily_sales_amount[n];
int i=0;
do{
    printf("enter the element %d of array :",i+1);
    scanf("%d",&daily_sales_amount[i]);
    i+=1;
}while(i<n);
int
size=sizeof(daily_sales_amount)/sizeof(daily_sales_amount[0])
; int result=total_sales(daily_sales_amount,size);
printf("the total sales for the day is :%d",result);

}

```

4

```

#include<stdio.h>
void print_array(int arr[],int s){
    printf("[");
    for(int i=0;i<s;i++){
        printf("%d",arr[i])
        ; if(i<s-1){
            printf(",");
        }
    }
    printf("]");
}

```

```
void discount_rates(int *arr1,int *arr2,int s){
```

```
    for(int i=0;i<s;i++){
```

```
        switch(arr1[i]){
```

```
            case 1 ...
```

```
                10:
```

```
                arr2[i]=5;
```

```
                break;
```

```
            case 11 ...
```

```
                20:
```

```
                arr2[i]=10;
```

```
                break;
```

```
            case 21 ...
```

```
                30:
```

```
                arr2[i]=15;
```

```
                break;
```

```
            case 31 ...
```

```
                40:
```

```
                arr2[i]=20;
```

```
                break;
```

```
            case 41 ...
```

```
                50:
```

```
                arr2[i]=25;
```

```
                break;
```

```
        }
```

```
    }
```

```
}
```

```
int
```

```
    main(){
```

```
        int n;
```

```
        printf("Enter the size of array :");
```

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

int

sales\_volumes[n];

int discount[n];

```

int
i=0;
do{
    printf("enter the element %d of array :",i+1);
    scanf("%d",&sales_volumes[i]);
    i+=1;
}while(i<n);
int
size=sizeof(sales_volumes)/sizeof(sales_volumes[0]);
discount_rates(sales_volumes,discount,size);
printf("the sales volumes are
:");
print_array(sales_volumes,siz
e); printf("\n");
printf("The discount prices are
:"); print_array(discount,size);

}

5
#include <stdio.h>

void classify_transactions(float *transactions, int size, float
threshold) { for (int i = 0; i < size; i++) {
    if (*(transactions + i) > threshold) {
        printf("Transaction %d: Suspicious (Amount: %.2f)\n", i + 1, *(transactions + i));
    } else {
        printf("Transaction %d: Normal (Amount: %.2f)\n", i + 1, *(transactions + i));
    }
}
}
}

```

```

int main() {
    int n;
    float threshold = 10000.0;
    printf("Enter the number of transactions: ");
    scanf("%d", &n);

    float transactions[n];
    for (int i = 0; i < n;
        i++) {
        printf("Enter transaction amount for transaction %d: ", i + 1);
        scanf("%f", &transactions[i]);
    }
    classify_transactions(transactions, n, threshold);

    return 0;
}

```

6

```

#include<stdio.h
> #define year 2
#define rate 10
void print_array(int arr[],int s){
    printf("[");
    for(int i=0;i<s;i++){
        printf("%d",arr[i])
        ; if(i<s-1){
            printf(",");
        }
    }
    printf("]");
}

```



```

}

void calculate_intrest(int *arr1,int *arr2,int s){
    int intrest;
    for(int i=0;i<s;i++){
        intrest=*(arr1+i)*(1+(rate/100))^year;
        arr2[i]=intrest;
    }
}

```

```

int
main(){
    int n;
    printf("enter the size of array :");
    scanf("%d",&n);
    int
    balances[n];
    int intrest[n];
    for(int i=0;i<n;i++){
        printf("enter the amount %d
        :",i+1); scanf("%d",&balances[i]);
    }
    int
    size=sizeof(balances)/sizeof(balances[0]);
    calculate_intrest(balances,intrest,size);
    printf("The balances remaining is :");
    print_array(intrest,size);

}

```

7

```

#include<stdio.h>

int make_transactions(int *d_t,int *w_t,int size){

```

```

while(size>0){
    int
    user_input;
    printf("Enter '1' to deposit and '2' for withdrawal :");
    scanf("%d",&user_input);
    if(user_input==1
        ){ int deposit;
            printf("enter the amount to deposit :");
            scanf("%d",&deposit);
            *d_t+=deposit;

        }else
            if(user_input==2){
                int withdraw;
                printf("enter the amount to withdraw :");
                scanf("%d",&withdraw);
                *w_t+=withdraw;
            }else{
                printf("Enter a valid input!");
            }
        size--;
    }
}

int
main(){
    int n;
    int deposit_total=0;
    int
    withdraw_total=0;
    printf("Enter the number of transactions :");
    scanf("%d",&n);
    make_transactions(&deposit_total,&withdraw_total,
n); printf("Total deposits : %d\n",deposit_total);

```

```

printf("Total withdrawal : %d\n",withdrawal_total);
printf("Net balance :%d",deposit_total-
withdrawal_total); return 0;
}

```

8

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

```

void check_eligibility(int *scores, int *eligibility, int size, int
threshold) { for (int i = 0; i < size; i++) {
    if (*(scores + i) >= threshold) {
        *(eligibility + i) = 1;
    } else {
        *(eligibility + i) = 0;
    }
}
}

```

```

void print_eligibility(int *eligibility, int size) {
    for (int i = 0; i < size; i++) {
        if (*(eligibility + i) == 1) {
            printf("Customer %d: Eligible for loan\n", i + 1);
        } else {
            printf("Customer %d: Not eligible for loan\n", i + 1);
        }
    }
}

```

```

int main() {
    int n, threshold = 650;
    printf("Enter the number of customers: ");
}

```

```

scanf("%d", &n);
int
credit_scores[n];
int eligibility[n];
for (int i = 0; i < n; i++) {
    printf("Enter credit score for customer %d: ", i + 1);
    scanf("%d", &credit_scores[i]);
}
check_eligibility(credit_scores, eligibility, n,
threshold); print_eligibility(eligibility, n);

return 0;
}

```

9

```

#include <stdio.h>

void calculate_total_cost(float *prices, int size, float *total_cost) {
    *total_cost = 0;
    for (int i = 0; i < size; i++) {
        *total_cost += *(prices + i);
    }
}

```

```

int main() {
    int n;
    printf("Enter the number of items:
"); scanf("%d", &n);
    float prices[n];
    for (int i = 0; i < n; i++) {
        printf("Enter the price for item %d: ", i + 1);

```

```

        scanf("%f", &prices[i]);
    }
    float total_cost;
    calculate_total_cost(prices, n, &total_cost);
    printf("The total order value is: %.2f\n", total_cost);

    return 0;
}

10
#include
<stdio.h>
#include
<stdlib.h>

int* flag_below_threshold(int *inventory, int size, int threshold, int
    *flagged_count) { int *flagged_indices = (int*) malloc(size * sizeof(int));
    *flagged_count = 0;

    for (int i = 0; i < size; i++) {
        if (*(inventory + i) < threshold) {
            flagged_indices[*flagged_count]
                = i; (*flagged_count)++;
        }
    }

    return flagged_indices;
}

int main() {
    int n, threshold;

```

```
printf("Enter the number of products: ");
scanf("%d", &n);
printf("Enter the stock threshold for replenishment: ");
scanf("%d", &threshold);

int inventory[n];

for (int i = 0; i < n; i++) {
    printf("Enter the inventory level for product %d: ", i +
1); scanf("%d", &inventory[i]);
}

int flagged_count;
int *flagged_indices = flag_below_threshold(inventory, n, threshold, &flagged_count);

if (flagged_count > 0) {
    printf("Products below the threshold (to be
replenished):\n"); for (int i = 0; i < flagged_count; i++) {
        printf("Product %d\n", flagged_indices[i] + 1);
    }
} else {
    printf("No products need replenishment.\n");
}

free(flagged_indices);

return 0;
}
```

11

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

```
void calculate_reward_points(float *purchases, int *reward_points, int
size) { for (int i = 0; i < size; i++) {
    if (*(purchases + i) <= 100) {
        *(reward_points + i) = 0;
    } else if (*(purchases + i) > 100 && *(purchases + i) <= 500) {
        *(reward_points + i) = *(purchases + i) / 10;
    } else {
        *(reward_points + i) = *(purchases + i) / 5;
    }
}
}
```

```
int main() {
    int n;

    printf("Enter the number of customers: ");
    scanf("%d", &n);

    float purchases[n];
    int
    reward_points[n];

    for (int i = 0; i < n; i++) {
        printf("Enter the purchase amount for customer %d: ", i + 1);
        scanf("%f", &purchases[i]);
    }
}
```

```
calculate_reward_points(purchases, reward_points, n);
```

```
printf("Customer Reward
```

```
Points:\n"); for (int i = 0; i < n; i++)
```

```
{
```

```
    printf("Customer %d: %.0f points\n", i + 1, (float)reward_points[i]);
```

```
}
```

```
return 0;
```

```
}
```

12

```
#include<stdio.h
```

```
>
```

```
#include<math.h
```

```
> #define size 5
```

```
void find_min_max(float *arr,float *ma,float
```

```
*mi){ for(int i=0;i<size;i++){
```

```
    if(*(arr+i)>*ma){
```

```
        *ma=*(arr+i);
```

```
    }
```

```
}
```

```
for(int i=0;i<size;i++){
```

```
    if(*(arr+i)<*mi){
```

```
        *mi=*(arr+i);
```

```
    }
```

```
}
```

```
}
```



```

int main(){
    float trajectories[5];
    float max=-
    INFINITY,min=INFINITY; for(int
    i=0;i<size;i++){
        printf("Enter the %d trajectory point :",i+1);
        scanf("%f",&trajectories[i]);
    }
    find_min_max(trajectories,&max,&min);
    printf("the maximum altitude is %.2f and minimum altitude is %.2f",max,min);
    return 0;
}

```

13

```

#include<stdio.h
> #define size 5
void clac_signals(int
    *arr){ for(int
    i=0;i<size;i++){
    switch(arr[i]){
        case 100 ... 200:
            printf("%d is a weak signal.\n",arr[i]);
            break;
        case 201 ... 300:
            printf("%d is a moderate
            signal.\n",arr[i]); break;
        case 301 ... 400:
            printf("%d is a good signal.\n",arr[i]);
            break;
        case 401 ... 500:
            printf("%d is very strong signal.\n",arr[i]);

```

```

        break;
    }
}
}
int main()
{
    int signals[size];
    for(int
i=0;i<size;i++){
        printf("enter the signal %d :",i+1);
        scanf("%d",&signals[i]);
    }
    clac_signals(signals);
    return 0;
}

```

14

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

```
#define SIZE 5
```

```

void assess_threat_level(int *sensor_readings, char *threat_levels, int
size) { for (int i = 0; i < size; i++) {
    if (sensor_readings[i] <= 50) {
        threat_levels[i] = 'L';
    } else if (sensor_readings[i] <= 100) {
        threat_levels[i] = 'M';
    } else {
        threat_levels[i] = 'H';
    }
}
}

```

```
}  
}
```

```
int main() {  
    int  
    sensor_readings[SIZE];  
    char  
    threat_levels[SIZE];  
  
    printf("Enter the sensor  
readings:\n"); for (int i = 0; i <  
SIZE; i++) {  
        printf("Sensor %d: ", i + 1);  
        scanf("%d", &sensor_readings[i]);  
    }  
  
    assess_threat_level(sensor_readings, threat_levels, SIZE);  
  
    printf("\nThreat Level  
Assessment:\n"); for (int i = 0; i <  
SIZE; i++) {  
        printf("Sensor %d: Reading = %d, Threat Level = ", i + 1,  
sensor_readings[i]); if (threat_levels[i] == 'L') {  
            printf("Low\n");  
        } else if (threat_levels[i] == 'M') {  
            printf("Moderate\n");  
        } else {  
            printf("High\n");  
        }  
    }  
  
    return 0;  
}
```

15

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

```
#define SIZE 5
```

```
void calibrate_signals(int *signal_data, int size)
```

```
{ for (int i = 0; i < size; i++) {  
    *(signal_data + i) += 10;  
}  
}
```

```
int main() {
```

```
    int signal_data[SIZE];
```

```
    printf("Enter the raw signal  
data:\n"); for (int i = 0; i < SIZE;  
i++) {  
        printf("Signal %d: ", i + 1);  
        scanf("%d",  
        &signal_data[i]);  
    }
```

```
    calibrate_signals(signal_data, SIZE);
```

```
    printf("\nCalibrated Signal  
Values:\n"); for (int i = 0; i < SIZE;  
i++) {  
        printf("Signal %d: %d\n", i + 1, signal_data[i]);  
    }
```

```
    return 0;
```

```
}
```

16

```
#include<stdio.h
```

```
> #define m 2
```

```
#define n 2
```

```
void calc_rowsum(int *arr,int
```

```
    *r_s){ for(int i=0;i<m;i++){
```

```
    int sum=0;
```

```
    for(int j=0;j<n;j++){
```

```
        sum+=*(arr+i*n+j);
```

```
    }
```

```
    r_s[i]=sum;
```

```
}
```

```
}
```

```
int main(){
```

```
    int matrix[m][n];
```

```
    int
```

```
    row_sums[m];
```

```
    printf("Enter the matrix elements \n");
```

```
    for(int i=0;i<m;i++){
```

```
        for(int j=0;j<n;j++){
```

```
            printf("Enter the element matrix[%d][%d] :",i,j);
```

```
            scanf("%d",&matrix[i][j]);
```

```
        }
```

```
    }
```

```
    calc_rowsum(&matrix[0][0],row_sums);
```

```
    printf("The sum of rows are :");
```

```
    printf("[");
```

```
    for(int i=0;i<n;i++){
```

```

        printf("%d",row_sums[i]
    ); if(i<n-1){
        printf(",");
    }
}
printf("]");
return 0;

}

17
#include <stdio.h>

#define SIZE 5

float calculate_mean(const int *data, int size)
{
    int sum = 0;

    for (int i = 0; i < size;
        i++) { sum += *(data
        + i);
    }

    return (float)sum / size;
}

int main() {
    int data[SIZE];

    printf("Enter %d data points:\n", SIZE);

```

```

for (int i = 0; i < SIZE; i++) {
    printf("Data point %d: ", i + 1);
    scanf("%d", &data[i]);
}

float mean = calculate_mean(data, SIZE);

printf("The mean of the data points is: %.2f\n", mean);

return 0;
}

```

18

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

```
#define SIZE 5
```

```

float* calculate_gradient(int *temp, int size) {
    static float gradients[SIZE - 1];

    for (int i = 0; i < size - 1; i++) {
        gradients[i] = (float)(temp[i + 1] - temp[i]);
    }

    return gradients;
}

```

```

int main() {
    int temperatures[SIZE];

```

```

printf("Enter %d temperature readings:\n",
SIZE); for (int i = 0; i < SIZE; i++) {
    printf("Temperature %d: ", i + 1);
    scanf("%d", &temperatures[i]);
}

float* gradients = calculate_gradient(temperatures, SIZE);

printf("Temperature Gradients:\n");
for (int i = 0; i < SIZE - 1; i++) {
    printf("Gradient between %d and %d: %.2f\n", i + 1, i + 2, gradients[i]);
}

return 0;
}

```

19

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

```
#define SIZE 5
```

```
void normalize_data(float *data, int
size) { float min = data[0], max =
data[0];
```

```

for (int i = 1; i < size;
i++) { if (data[i] < min)
{
    min = data[i];
}
}

```



```

        if (data[i] > max)
        { max =
          data[i];
        }
    }

    for (int i = 0; i < size; i++) {
        data[i] = (data[i] - min) / (max - min);
    }
}

int main() {
    float data[SIZE];

    printf("Enter %d data points:\n",
    SIZE); for (int i = 0; i < SIZE; i++) {
        printf("Data point %d: ", i + 1);
        scanf("%f", &data[i]);
    }

    normalize_data(data, SIZE);

    printf("Normalized Data:\n");
    for (int i = 0; i < SIZE; i++) {
        printf("%.2f ", data[i]);
    }

    return 0;
}

```

20

```
#include<stdio.h>

void find_calc(int *arr,int *h_s,float *avg,int
n){ int sum=0;
for(int i=0;i<n;i++){
    sum+=*(arr+i);
    if(*(arr+i)>*h_s){
        *h_s=*(arr+i);
    }
}
*avg=(float)sum/n;
}

int
main(){
int n;
printf("Enter the number of marks to input :");
scanf("%d",&n);
int marks[n];
for(int
i=0;i<n;i++){
    printf("Enter the mark %d :",i+1);
    scanf("%d",&marks[i]);
}
int
highest_score=0;
float avg_score;
find_calc(marks,&highest_score,&avg_score,n);
printf("The highest score obtained is %d.\n",highest_score);
printf("The average score is %.2f.",avg_score);
return 0;
}
```

21

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

```
#define SIZE 5
```

```
void assign_grades(int *marks, char *grades, int
size) { for (int i = 0; i < size; i++) {
    switch (*(marks + i) / 10) {
        case 10:
        case 9:
            *(grades + i) =
            'A'; break;
        case 8:
            *(grades + i) =
            'B'; break;
        case 7:
            *(grades + i) =
            'C'; break;
        case 6:
            *(grades + i) =
            'D'; break;
        default:
            *(grades + i) =
            'F'; break;
    }
}
}
```

```
int main() {
```

```

int marks[SIZE];
char
grades[SIZE];

for (int i = 0; i < SIZE; i++) {
    printf("Enter mark for student %d: ", i +
        1); scanf("%d", &marks[i]);
}

assign_grades(marks, grades, SIZE);

printf("Grades of the students:\n");
for (int i = 0; i < SIZE; i++) {
    printf("Student %d: %c\n", i + 1, grades[i]);
}

return 0;
}

22
#include <stdio.h>

#define SIZE 5

int* track_defaulters(float *attendance, int *defaulters, int size)
{
    int count = 0;
    for (int i = 0; i < size; i++) {
        if (*(attendance + i) < 75.0) {
            defaulters[count++] = i;
        }
    }
}

```

```

    }
    return defaulters;
}

int main() {
    float
    attendance[SIZE]; int
    defaulters[SIZE];

    for (int i = 0; i < SIZE; i++) {
        printf("Enter attendance percentage for student %d: ", i + 1);
        scanf("%f", &attendance[i]);
    }

    int* defaulter_indices = track_defaulters(attendance, defaulters, SIZE);

    printf("Defaulters'
    indices:\n"); for (int i = 0; i <
    SIZE; i++) {
        if (defaulter_indices[i] != 0) {
            printf("Student %d\n", defaulter_indices[i] + 1);
        }
    }

    return 0;
}

```

23

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

```
#define SIZE 5
```

```

void categorize_performance(const int *scores, int
size) { for (int i = 0; i < size; i++) {
    if (*(scores + i) >= 90) {
        printf("Student %d: Excellent\n", i + 1);
    } else if (*(scores + i) >= 75) {
        printf("Student %d: Good\n", i +
1);
    } else if (*(scores + i) >= 50) {
        printf("Student %d: Average\n", i + 1);
    } else {
        printf("Student %d: Poor\n", i + 1);
    }
}
}

```

```

int main() {
    int scores[SIZE];

    for (int i = 0; i < SIZE; i++) {
        printf("Enter score for student %d: ", i +
1); scanf("%d", &scores[i]);
    }

    categorize_performance(scores, SIZE);

    return 0;
}

```