```c
/*
   Name: Ayush Gupta
   Roll No: MNS2025027
   Lab: APP Lab-2
   Date: 22-Aug-2025
*/



// Write a C program to reverse an array of integers without using any additional array.
#include<stdio.h>

void reverse_array(int a[],int n){
    int first = 0 , last = n-1;
    while(first<last){
        int temp = a[first];
        a[first]=a[last];
        a[last]=temp;
        first++,last--;
    }
}

int main(){
    int a[]={1,2,3,4,5};
    int n = sizeof(a)/sizeof(a[0]);

    printf("Original Array: ");
```

```c
    for(int i = 0 ;i< n;i++){
        printf("%d ",a[i]);
    }

    reverse_array(a,n);

    printf("\nReversed Array: ");

    for(int i = 0 ;i< n;i++){
        printf("%d ",a[i]);
    }

    return 0;
}

// 2. Given an array of n integers, that may contain
repeating elements. Write a C program to
// print the unique elements along with the frequency of
that element in the input array.

#include<stdio.h>


int main(){
    int a[]={1,2,3,4,5,3,2,2,3,4,5,1};
    int n = sizeof(a)/sizeof(a[0]);
```

```
    int count = 0;

    int frequency[n];

    for (int i = 0; i < n; i++) frequency[i] = 0;

    for(int i =0;i<n;i++){
        frequency[a[i]]++;
    }

    for(int i =0;i<n;i++){
        if(frequency[i])
       printf("\nThe frequency of %d in the array is: %d
",i,frequency[i]);
    }

    return 0;
}



// 3. Take a 3-dimensional array, say a[p][r][c], where p
denotes planes, each with r rows and
// c columns. Initialize the array with numbers sequentially
from 1 to p*r*c, and take a
// pointer, x, pointing to the base address of the array.
Write a C program to print the
// maximum element among the corner elements of each plane
accessed through the
```

```c
// pointer. (refer program3, Lab-1)

#include<stdio.h>

int max(int a, int b){
    if(a>b)return a;
    return b;
}

int main(){
    int a[3][4][4] = {
    {
        {  1,  2,  3,  4 },
        {  5,  6,  7,  8 },
        {  9, 10, 11, 12 },
        { 13, 14, 15, 16 }
    },
    {
        { 17, 18, 19, 20 },
        { 21, 22, 23, 24 },
        { 25, 26, 27, 28 },
        { 29, 30, 31, 32 }
    },
    {
        { 33, 34, 35, 36 },
        { 37, 38, 39, 40 },
        { 41, 42, 43, 44 },
        { 45, 46, 47, 48 }
```

```
    }
};
    int n = sizeof(a)/sizeof(a[0]);


    for (int plane = 0; plane < n; plane++) {
        int current_max= a[plane][0][0];
        for(int j = 0 ;j<4;j+=3){
            for(int k=0;k<4;k+=3)
            current_max = max(current_max,a[plane][j][k]);
        }
        printf("\nThe max element of %d plane is :
%d",plane,current_max);
    }

    return 0;
}



// 4. Given an array of integers, write a C program to find
the index where the sum of
// elements at the left-hand side of it is the same as the
sum of elements at the right-hand
// side of it. ( Example: input array : -7 1 5 2 -4 3 0 ,
output: index - 3)
```

```c
#include<stdio.h>

int midpoint(int a[],int n){
    int total = 0 , left_sum = 0;

    for(int i =0;i<n;i++){
        total+=a[i];
    }

    for(int i =0;i<n;i++){
        int right_sum = total-left_sum-a[i];
        if(right_sum==left_sum){
            return i;
        }
        left_sum+=a[i];
    }
    return -1;
}

int main (){
    int a[]={-7, 1, 5, 2, -4, 3, 0};
    int n = sizeof(a)/sizeof(a[0]);

    int index = midpoint(a,n);

    if(index!=-1) printf("%d",index);
    else printf("Sorry! no index found...");
```

```c
    return 0;
}


// 5. Using program 5, Lab-1, write a program to find the student having the highest mark.


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

#define MAX 5

typedef struct {
    char name[30];
    int roll;
    float marks;
    char grade;
} Student;

Student s[MAX];
int n = 5;

void show() {
    printf("\n--- Grade Sheet ---\n");
    for (int i = 0; i < n; i++)
```

```
        printf("%d. %s | Roll: %d | Marks: %.1f | Grade:
%c\n", i+1, s[i].name, s[i].roll, s[i].marks, s[i].grade);
}

void add() {
    if (n >= MAX) {
        printf("Limit reached.\n");
        return;
    }
    printf("Enter Name Roll Marks Grade: ");
    scanf("%s %d %f %c", s[n].name, &s[n].roll, &s[n].marks,
&s[n].grade);
    n++;
}

void remove_by_roll() {
    int r;
    printf("Enter Roll to Delete: ");
    scanf("%d", &r);
    for (int i = 0; i < n; i++) {
        if (s[i].roll == r) {
            s[i] = s[n-1];
            n--;
            printf("Deleted.\n");
            return;
        }
    }
    printf("Roll not found.\n");
```

```
}

void update() {
    int r;
    printf("Enter Roll to Modify: ");
    scanf("%d", &r);
    for (int i = 0; i < n; i++) {
        if (s[i].roll == r) {
            printf("Enter New Name Marks Grade: ");
            scanf("%s %f %c", s[i].name, &s[i].marks,
&s[i].grade);
            printf("Updated.\n");
            return;
        }
    }
    printf("Roll not found.\n");
}

void topper() {
    if (n == 0) {
        printf("No records.\n");
        return;
    }
    int top = 0;
    for (int i = 1; i < n; i++)
        if (s[i].marks > s[top].marks) top = i;
    printf("\n Topper: %s (Roll %d) with %.1f marks\n",
s[top].name, s[top].roll, s[top].marks);
```

```c
}

int main() {
    // Preloaded data
    strcpy(s[0].name, "Ayush"); s[0].roll = 101; s[0].marks
= 88; s[0].grade = 'A';
    strcpy(s[1].name, "Riya");  s[1].roll = 102; s[1].marks
= 91; s[1].grade = 'A';
    strcpy(s[2].name, "Kabir"); s[2].roll = 103; s[2].marks
= 76; s[2].grade = 'B';
    strcpy(s[3].name, "Neha");  s[3].roll = 104; s[3].marks
= 84; s[3].grade = 'A';
    strcpy(s[4].name, "Zara");  s[4].roll = 105; s[4].marks
= 95; s[4].grade = 'A';

    int ch;
    do {
        printf("\n1.Show 2.Add 3.Delete 4.Modify 5.Topper
6.Exit\nChoice: ");
        scanf("%d", &ch);
        switch(ch) {
            case 1: show(); break;
            case 2: add(); break;
            case 3: remove_by_roll(); break;
            case 4: update(); break;
            case 5: topper(); break;
        }
    } while (ch != 6);
```

```
    return 0;
}
```