ASSIGNMENT 2

WAP to increase every student mark by 5 & then print the updated array.

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

}

printf("New marks are: ");

for (int i = 0; i < num\_students; i++) {

marks[i] += 5;

printf("%.2f ", marks[i]);

}

printf("\n");

return 0;

}

Q2. WAP to print grade of students as per their marks given in an array. (>=75-- A

grade, 74 to 60--B Grade, 59 to 40--C grade below 40--D grade).

#include <stdio.h>

void print\_grade(float mark) {

if (mark >= 75) {

printf("A Grade\n");

} else if (mark >= 60) {

printf("B Grade\n");

} else if (mark >= 40) {

printf("C Grade\n");

} else {

printf("D Grade\n");

}

}

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

}

printf("Grades of the students:\n");

for (int i = 0; i < num\_students; i++) {

printf("Student %d: ", i + 1);

print\_grade(marks[i]);

}

return 0;

}

Q3. WAP to find who scored first "99" in an array marks.

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

}

int found = 0;

for (int i = 0; i < num\_students; i++) {

if (marks[i] == 99) {

found = i + 1;

break;

}

}

if (found != 0) {

printf("The first student to score 99 is student %d.\n", found);

} else {

printf("No student scored 99.\n");

}

return 0;

}

Q4 WAP to find Who & how many students have scored 99 in an array Marks

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students];

int count = 0;

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

}

printf("Students who scored 99: ");

for (int i = 0; i < num\_students; i++) {

if (marks[i] == 99) {

printf("Student %d ", i + 1);

count++;

}

}

if (count == 0) {

printf("None\n");

} else {

printf("\nTotal number of students who scored 99: %d\n", count);

}

return 0;

}

Q5   
WAP to find sum of all scores in Marks array.  
#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

float sum = 0;

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

sum += marks[i];

}

printf("Sum of all scores: %.2f\n", sum);

return 0;

}

Q6   
WAP to find average score of the Marks array.  
#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

float sum = 0;

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

sum += marks[i];

}

float average = sum / num\_students;

printf("Average score: %.2f\n", average);

return 0;

}

Q7 WAP to check whether score is even or odd in an array.  
#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

}

for (int i = 0; i < num\_students; i++) {

if (marks[i] % 2 == 0) {

printf("Student %d: Even\n", i + 1);

} else {

printf("Student %d: Odd\n", i + 1);

}

}

return 0;

}

Q8   
WAP to find maximum & minimum score in the Marks array

#include <stdio.h>

// Function to find minimum and maximum elements in an array

void findMinMax(float arr[], int n) {

float min = arr[0];

float max = arr[0];

for (int i = 1; i < n; i++) {

if (arr[i] < min)

min = arr[i];

else if (arr[i] > max)

max = arr[i];

}

printf("Minimum score: %.2f\n", min);

printf("Maximum score: %.2f\n", max);

}

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

float marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

scanf("%f", &marks[i]);

}

// Find minimum and maximum elements

findMinMax(marks, num\_students);

return 0;

}

Q9   
WAP to find a peak element which is not smaller than its neighbors.

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

}

for (int i = 0; i < num\_students; i++) {

if ((i == 0 || marks[i] >= marks[i - 1]) && (i == num\_students - 1 || marks[i] >= marks[i + 1])) {

printf("Peak element found: %d at index %d\n", marks[i], i);

return 0;

}

}

printf("No peak element found.\n");

return 0;

}

Q10 WAP to count prime numbers in an array.

#include <stdio.h>

int is\_prime(int n) {

if (n <= 1) return 0;

for (int i = 2; i \* i <= n; i++) {

if (n % i == 0) return 0;

}

return 1;

}

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students];

int prime\_count = 0;

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

if (is\_prime(marks[i])) {

prime\_count++;

}

}

printf("Count of prime numbers: %d\n", prime\_count);

return 0;

}

Q11 WAP to implement Insert -Front, any position in between & end in an array. Print  
the array before insert & after insert.

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students + 1];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

}

printf("Array before insert: ");

for (int i = 0; i < num\_students; i++) {

printf("%d ", marks[i]);

}

printf("\n");

int position, value;

printf("Enter the position to insert (1 to %d for front, %d to %d for between, %d for end): ", 1, num\_students, num\_students + 1);

scanf("%d", &position);

printf("Enter the value to insert: ");

scanf("%d", &value);

if (position < 1 || position > num\_students + 1) {

printf("Invalid position! Please enter a position between 1 and %d.\n", num\_students + 1);

return 1;

}

for (int i = num\_students; i >= position - 1; i--) {

marks[i] = marks[i - 1];

}

marks[position - 1] = value;

num\_students++;

printf("Array after insert: ");

for (int i = 0; i < num\_students; i++) {

printf("%d ", marks[i]);

}

printf("\n");

return 0;

}

Q12 WAP to implement delete-Front, any position in between & end in an array. Print  
the array before delete & after delete.

#include <stdio.h>

int main() {

int num\_students;

printf("Enter the number of students: ");

scanf("%d", &num\_students);

int marks[num\_students];

for (int i = 0; i < num\_students; i++) {

printf("Enter marks for student %d: ", i + 1);

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

}

printf("Array before delete: ");

for (int i = 0; i < num\_students; i++) {

printf("%d ", marks[i]);

}

printf("\n");

int position;

printf("Enter the position to delete (1 to %d for front, 2 to %d for between, %d for end): ", num\_students, num\_students, num\_students);

scanf("%d", &position);

if (position < 1 || position > num\_students) {

printf("Invalid position!\n");

return 1;

}

for (int i = position - 1; i < num\_students - 1; i++) {

marks[i] = marks[i + 1];

}

num\_students--;

printf("Array after delete: ");

for (int i = 0; i < num\_students; i++) {

printf("%d ", marks[i]);

}

printf("\n");

return 0;

}

Q13 Given an array, the task is to cyclically rotate the array clockwise by one time.  
Examples:  
Input: arr[] = {1, 2, 3, 4, 5}  
Output: arr[] = {5, 1, 2, 3, 4}  
Input: arr[] = {2, 3, 4, 5, 1}  
Output: {1, 2, 3, 4, 5}

#include <stdio.h>

int main() {

int num\_elements;

printf("Enter the number of elements in the array: ");

scanf("%d", &num\_elements);

int arr[num\_elements];

for (int i = 0; i < num\_elements; i++) {

printf("Enter element %d: ", i + 1);

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

}

printf("Array before rotation: ");

for (int i = 0; i < num\_elements; i++) {

printf("%d ", arr[i]);

}

printf("\n");

if (num\_elements > 1) {

int last = arr[num\_elements - 1];

for (int i = num\_elements - 1; i > 0; i--) {

arr[i] = arr[i - 1];

}

arr[0] = last;

}

printf("Array after rotation: ");

for (int i = 0; i < num\_elements; i++) {

printf("%d ", arr[i]);

}

printf("\n");

return 0;

}

Q14   
Given an array of n integers. The task is to print the duplicates in the given array.  
If there are no duplicates then print -1.  
Examples:  
Input: {2, 10,10, 100, 2, 10, 11,2,11,2}  
Output: 2 10 11  
Input: {5, 40, 1, 40, 100000, 1, 5, 1}  
Output: 5 40 1

#include <stdio.h>

int main() {

int n;

printf("Enter the number of elements in the array: ");

scanf("%d", &n);

int arr[n];

int duplicates[n];

int count = 0;

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

printf("Enter element %d: ", i + 1);

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

}

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

for (int j = i + 1; j < n; j++) {

if (arr[i] == arr[j]) {

int alreadyExists = 0;

for (int k = 0; k < count; k++) {

if (duplicates[k] == arr[i]) {

alreadyExists = 1;

break;

}

}

if (!alreadyExists) {

duplicates[count++] = arr[i];

}

break;

}

}

}

if (count == 0) {

printf("-1\n");

} else {

printf("Duplicates: ");

for (int i = 0; i < count; i++) {

printf("%d ", duplicates[i]);

}

printf("\n");

}

return 0;

}