Q-1 - - WAP to increase every students mark by 5 $ then print the updated array?

Sol -

#include <stdio.h>

int main() { int n, i;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) {

}

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

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

}

marks[i] += 5;

printf("Updated marks of the students: \n");

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

}

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

printf("\n");

return 0;

}

Q-2 - - 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)?

Sol -

#include <stdio.h> int main() {

int n, i;

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

scanf("%d", &n);

int marks[n];

char grades [n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) {

}

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

for (i = 0; i < n; i++) { if (marks[i] >= 75) { grades[i] = 'A';

}

} else if (marks[i] >= 60) { grades[i] = 'B';

} else if (marks[i] >= 40) { } else if (marks[i] >= 40) {

grades[i]= 'C':

} else {

}

grades[i] = 'D' ;

printf("Marks and Grades of the students: \n"); for (i = 0; i < n; i++) {

printf("Marks: %d - Grade: %c\n", marks[i], grades[i]);}

return 0;

Q-3 - - WAP to find who scored first “99’” in an array marks?

Sol -

#include <stdio.h> int main() {

int n, i, index = -1;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) {

}

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

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

if (marks[i] == 99) { index = i;

break;

}

}

if (index = -1) {

printf("The first student to score 99 is at index: %d\n", index);

} else {

}

return 0;

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

Q-4 - - WAP to find who $ how many students have scored 99 in an array marks?

Sol -

#include <stdio.h> int main() { int n, i;

int count = 0;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n");

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

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

printf("Students who scored 99 are at indices: ");

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

if (marks[i]

}

==

99) {

printf("%d ", i);

count++;

}

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

if (count == 0) {

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

}

return 0;

Q-5 - - WAP to find sum of all scores in marks array?

Sol -

#include <stdio.h> int main() {

int n, i, sum = 0;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) {

}

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

for (i = 0; i < n; i++) { sum += marks[i];

}

printf("The sum of all scores is: %d\n", sum);

return 0;

}

Q-6 - - WAP to find average score of the marks array?

Sol -

#include <stdio.h> int main() {

int n, i;

float sum 0.0, average:

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i=0; i < n; i++) {

}

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

sum += marks[i];

average = sum / n;

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

return 0;

}

Q-7 - - WAP to check whether score is even or odd in an array?

Sol -

#include <stdio.h> int main() {

int n, i;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) {

}

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

printf("Scores and their even/odd status: \n"); for (i = 0; i < n; i++) {

}

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

printf("Score %d is even.\n", marks[i]);

} else {

}

return 0;

printf("Score %d is odd. \n", marks[i]);

}

Q-8 - - WAP to find minimum & maximum score in the marks array?

Sol -

#include <stdio.h> int main() { int n, i;

int min, max;

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

scanf("%d", &n);

int marks[n];

printf("Enter the marks of the students: \n"); for (i = 0; i < n; i++) { scanf("%d", &marks[i]);

}

min = max = marks[0];

for (i = 1; i < n; i++) { if (marks[i] < min) { min = marks[i];

}

if (marks[i] > max) { max = marks[i];

}

}

printf("Minimum score: %d\n", min); printf("Maximum score: %d\n", max); return 0;}

Q-9 - - WAP to find a peak element which is not smaller than its neighbour?

Sol -

#include <stdio.h>

int main() { int n;

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

scanf("%d", &n);

if (n <= 0) {

}

printf("Array size must be greater than 0.\n"); return 1;

int arr[n];

printf("Enter the elements of the array:\n"); for (int i = 0; i < n; i++) {

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

}

for (int i

=

if ((i

==

0; i < n; i++) {

0 || arr[i] >= arr[i - 1]) && (i == n - 1 || arr[i] >= arr[i + 1])) {

printf("Peak element found: %d at index %d\n", arr[i], i); return 0;

}

}

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

return 0;

}

Q-10 - - WAP to count prime numbers in an array?

Sol -

int isPrime(int num) {

if (num <= 1) return 0; // 0 and 1 are not prime numbers for (int i = 2; i\* i <= num; i++) {

if (num % i == 0) {

return 0; // Not a prime number

}

}

}

return 1; // Is a prime number

int main() {

int n, count = 0;

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

scanf("%d", &n);

int arr[n];

printf("Enter the elements of the array: \n"); for (int i = 0; i < n; i++) {

}

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

for (int i = 0; i < n; i++) { if (isPrime(arr[i])) {

}

}

count++;

printf("Number of prime numbers in the array: %d\n", count); return 0;

Q-11 - - WAP to implement Insert -Front, any position in between & end in an array. Print thearray before insert & after insert?

Sol -

#include <stdio.h>

void printArray(int arr[], int size) { for (int i = 0; i < size; i++) {

}

}

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

printf("\n");

int main() {

int n, arr[100], value, position, choice; // Input the number of elements printf("Enter the number of elements: "); scanf("%d", &n);

// Input elements of the array printf("Enter the elements: \n"); for (int i = 0; i < n; i++) { scanf("%d", &arr[i]);

}

// Print original array

printf("Original array: ");

printArray(arr, n);

// Choose where to insert

printf("Choose insertion option: \n1. Front \n2. Position\n3.

End\n");

scanf("%d", &choice);

switch (choice) {

case 1: // Insert at front

End\n");

scanf("%d", &choice);

switch (choice) {

case 1: // Insert at front

printf("Enter value to insert at front: "); scanf("%d", &value);

for (int i = n; i > 0; i--) arr[i] = arr[i arr[0]= value;

n++;

break;

case 2: // Insert at position

printf("Enter position (0 to %d): ", n); scanf("%d", &position);

if (position < 0 || position > n) { printf("Invalid position!\n"); return 1;

}

printf("Enter value to insert: ");

scanf("%d", &value);

1]:

for (int i = n; i > position; i--) arr[i] = arr[i - 1]; arr[position] = value;

n++;

break;

case 3: // Insert at end

printf("Enter value to insert at end: ");

scanf("%d", &value);

arr[n++] = value;

End\n");

scanf("%d", &choice);

switch (choice) {

case 1: // Insert at front

printf("Enter value to insert at front: "); scanf("%d", &value);

for (int i = n; i > 0; i--) arr[i] = arr[i arr[0]= value;

n++;

break;

case 2: // Insert at position

printf("Enter position (0 to %d): ", n); scanf("%d", &position);

if (position < 0 || position > n) { printf("Invalid position!\n"); return 1;

}

printf("Enter value to insert: ");

scanf("%d", &value);

1]:

for (int i = n; i > position; i--) arr[i] = arr[i - 1]; arr[position] = value;

n++;

break;

case 3: // Insert at end

printf("Enter value to insert at end: ");

scanf("%d", &value);

arr[n++] = value;

Q-12 - - WAP to implement delete-Front, any position in between & end in an array. Print thearray before delete & after delete?

Sol -

#include <stdio.h>

void printArray(int arr[], int size) { for (int i = 0; i < size; i++) {

}

}

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

printf("\n");

int main() {

int n, arr[100], position, choice;

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

scanf("%d", &n);

printf("Enter the elements: \n"); for (int i = 0; i < n; i++) {

}

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

printf("Original array: ");

printArray(arr, n);

printf("Choose deletion option: \n1. Front\n2. Position\n3.

End\n");

scanf("%d", &choice);

switch (choice) {

case 1: // Delete from front

if (n > 0) {

for (int i = 0; i < n - 1; i++) {

arr[i] = arr[i + 1];

for (int i = 0; i < n − 1; i++) {

arr[i] = arr[i + 1];

}

}

n-

break;

case 2: // Delete from position

printf("Enter position (0 to %d): ", n = 1); scanf("%d", &position);

if (position < 0 || position >= n) {

}

printf("Invalid position!\n"); return 1;

for (int i = position; i < n - 1; i++) { arr[i] = arr[i + 1];

}

n--;

break;

case 3: // Delete from end

if (n > 0) {

}

n--;

break;

default:

printf("Invalid choice!\n");

return 1;

}

case 3: // Delete from end

if (n > 0) {

}

n--;

break;

default:

printf("Invalid choice!\n");

return 1;

printf("Updated array: ");

printArray(arr, n);

return 0;

Q-13 - - Given an array, the task is to cyclically rotate the array clockwise by one time.

Sol -

#include <stdio.h>

void rotateArray(int arr[], int n) {

}

if (n = 1) return;

int last = arr[n-1]; // Store the last element for (int i = n - 1; i > 0; i--) {

}

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

arr[0] = last; // Place the last element at the front

void printArray(int arr[], int n) {

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

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

printf("\n");

}

}

int main() {

int arr[] = {1, 2, 3, 4, 5};

int n = sizeof(arr) / sizeof(arr[0]);

printf("Original array: ");

printArray(arr, n);

rotateArray(arr, n); // Rotate the array

printf("Rotated array: ");

printArray(arr, n);

return 0;

}

Q-14 - - Given an array of n integers. The task is to print the duplicates in the given array. If thereare no duplicates then print -1.

Sol -

#include <stdio.h>

void printDuplicates (int arr[], int n) {

int foundDuplicate = 0;

int duplicate[n];

int duplicateCount = 0;

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 < duplicateCount; k++) { if (duplicate[k] == arr[i]) {

alreadyExists = 1;

break;

}

}

if (!alreadyExists) {

duplicate[duplicateCount++] = arr[i];

}

break;

}

}

}

if (duplicateCount > 0) {

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

printf("%d"

dunlicateril):

}

if (!alreadyExists) {

duplicate[duplicateCount++] = arr[i];

}

break;

}

}

}

if (duplicateCount > 0) {

for (int i = 0; i < duplicateCount; i++) { printf("%d ", duplicate[i]);

}

printf("\n"); } else {

printf("-1\n");

}

int main() {

int arr[] = {2, 10, 10, 100, 2, 10, 11, 2, 11, 2); int n = sizeof(arr) / sizeof(arr[0]);

printDuplicates (arr, n);

return 0;